Nanopore adaptive sequencing of gigabase length genomes for mixed samples, whole exome capture, and targeted panels

Alexander Payne
School of Life Sciences
University of Nottingham

Doctor of Philosophy

March 2022

Abstract

Single molecule sequencing technologies, such as nanopore sequencing, provide new ways to investigate genomes and genetics. They permit the detailed analysis of stretches of DNA orders of magnitude larger than previously possible. Studying genomes at this detail allows for a better understanding of genome organisation and structural variants that are typically difficult to resolve using short read sequencing.

Oxford Nanopore Technologies sequencers drive single molecules of DNA through membrane bound protein nanopores by applying a voltage across the membrane. This applied voltage draws ions and DNA through the nanopore, which is measured as a real-time data stream of ionic current. Inspecting the current data in real-time allows for specific molecules to be rejected by reversing the voltage across an individual nanopore. This process is called "Read Until".

Previously, Read Until has been carried out by inspecting and comparing the current data produced during sequencing. This dissertation proposes a method for implementing Read Until using graphics cards to accelerate basecalling and optimised real-time alignment.

To build up to a full system for selective sequencing, the raw signal data that nanopore sequencers output must be assessed (Chapter 3). Specifically to better understand the characteristics of the continuous data stream. This is accomplished by inspecting bulk FAST5 files, first a visualisation application is built. This visualisation application is then used to assess both DNA and RNA samples, specifically looking at how unblocking behaviour is actioned and the impact it has on sequencing.

With a grasp of raw signal data an application, readfish, is developed aiming to enable real-time basecalling of read chunks for currently sequencing molecules (Chapter 4). This approach uses GPU accelerated basecalling and fast alignment to make decisions on selecting and rejecting individual molecules. In addition, a schema is designed to allow for arbitrary experiments to be devised allowing multiple experiments to take place simultaneously. Then, an optimised CPU basecaller and barcode demultiplexing are incorporated extending the platforms and types of samples that can be considered.

As a proof-of-concept readfish is used to selectively sequence target panels encompassing thousands of loci in the form of whole exome sequencing of the human cell line NA12878. This single experiment demonstrates great flexibility in the chosen target panel and the ability to use reference genomes at a gigabase scale. In further experiments using the ZymoBIOMICS mock community adaptive techniques

are introduced as the experimental parameters are updated — dynamically — in response to the data generated by the same experiment.

Finally, exemplar problems and applications of selective sequencing are considered as well as other practical mechanisms for real-time feedback making the whole process adaptive (Chapter 5). These exemplar problems show how the methods developed in this thesis enable the time-efficient screening using panels of gene targets, decrease the time to identifying fusions in a leukaemic cell line, and reduce sequencing costs through standard library preparation methods.

Acknowledgements

I want to thank the following people for their help over the last four years.

My supervisor, Matt Loose, for his constant support and making me think. Most of all for being a fantastic mentor, his encouragement, and pushing me to finish this thesis. Stu Reid, George Pimm, Alex Merry, Forrest Brennan, Steve Pool, and Chris Seymour at Oxford Nanopore for constantly pointing me in the right direction. The team at DeepSeq for their endless patience and help with my experiments. Everyone else around the QMC over the years: Teri, Darren, Ninin, Fiona, Lewis, Chris, Sam, Thomas, Johnny, Emma. I would especially like to thank Andrew Renault, Stephen Gray, and Bill Wickstead for their cheerful and encouraging scientific chats. Rory, Luke, Louisa, Alex J, and Morgan for all the pints, listening to me rant, and happy distractions. I would also like to thank Jack, Josh, and Finn for the climbing, drinks, and laughs.

I would to thank my parents and family for their support and motivation in finishing and putting up with me forgetting to phone them. I expect you to read it all. Finally, I would like to thank Bethany for just being Bethany.

Thank you all for the patience, support and laughter.

Contents

List of Figures

List of Tables

List of Abbreviations, Acronyms, and Symbols

α-HL	Alpha haemolysin
A	Adenine
API	Application Programming Interface
APL	Acute Promyelocytic Leukaemia
ASIC	Application Specific Integreated Circuit
ATP	Adenosine triphosphate
bp	Base pair
BRCA1	Breast Cancer 1 Gene
C	Cytosine
CCS	Circular Consensus Sequence
cDNA	Complementary deoxyribonucleic
ChIP	Chromatin immunoprecipitation
CHR	Chromosome
CLR	Continuous Long Read
CNN	Convolutional Neural Network
CNV	Copy Number Variation
COSMIC	Catalogue Of Somatic Mutations In Cancer
CPU	Central Processing Unit
CRISPR	Clustered Regularly Interspaced Short Palindromic Repeats
CsgF	Curli production assembly/transport component CsgF
CsgG	Curli production assembly/transport component CsgG
CT47	Cancer/Testis Antigen Family 47

CTC	Connectionist Temporal Classification
CUDA	Compute Unified Device Architecture
Dda	ATP-dependent DNA helicase Dda
ddNTP	Dideoxyribonucleotide triphosphate
DNA	Deoxyribonucleic acid
dNTP	Deoxyribonucleotide triphosphate
DTW	Dynamic Time Warping
FM-index	Ferragina-Manzini index
G	Guanine
GPU	Graphics Processing Unit
gRPC	Google Remote Procedural Calls
GRU	Gated Recurrent Unit
HAC	High Accuracy
HDF5	Hierarchical Data Format 5
HMM	Hidden Markov Model
HMW	High Molecular Weight
HOXC4–13	Homeobox proteins 4–13
LSTM	Long Short-Term Memory
mRNA	Messenger Ribonucleic acid
mt-co1	Cytochrome c oxidase subunit 1
NB4	An acute promyelocytic leukaemia cell line
NBR1	Next to BRCA1 gene 1
NCBI	National Center for Biotechnology Information
ND3	NADH-ubiquinone oxidoreductase chain 3
ND5	NADH-ubiquinone oxidoreductase chain 5
NGS	Next Generation Sequencing
nt	Nucleotide
OLC	Overlap Layout Consensus

ONT	Oxford Nanopore Technologies
PacBio	Pacific Biosciences
PAF	Pairwise mApping Format
PCR	Polymerase Chain Reaction
pH	Power of hydrogen
PML	Protein PML
QC	Quality Control
RARA	Retinoic acid receptor alpha
RNA	Ribonucleic acid
SMRT	Single-Molecule Real-Time
SNP	Single Nucleotide Polymorphism
SNV	Single Nucleotide Variant
ssDNA	Single-stranded deoxyribonucleic acid
STR	Short Tandem Repeat
SV	Structural Variant
T	Thymine
TALENS	Transcription activator-like effector nucleases
TLB	Tris Lysis Buffer
TOML	Toms Obvious Markup Language
WES	Whole Exome Sequencing
WGA	Whole Genome Amplification
WGS	Whole Genome Sequencing
WSL	Windows Subsystem for Linux
×g	Times gravity
ZFN	Zinc Finger Nucleases
ZMW	Zero-Mode Waveguide

Chapter 1

Introduction

Living systems can vary in scale including single cells, multicellular organisms, and entire communities. A single cell is often considered the smallest, autonomous, unit of life as it contains all the necessary components for self-propagation (Alberts et al., 2017). Therefore, one approach to learn how a living system functions is to study and understand its underlying molecular organisation. All cells use carbon-based (organic) molecules that can be categorised into sugars, fatty acids, amino acids, and nucleic acids (Alberts et al., 2017). Sugars are an immediate source of energy for cells and can be incorporated into polysaccharides for energy storage. Fatty acids are also used for energy storage but are essential for the formation of cell membranes. Amino acids organise into long chains that fold into proteins. And, finally, nucleotides are used for energy transfer, while also serving as the subunits for the informational macromolecules, ribonucleic acid (RNA) (Zalokar, 1960) and deoxyribonucleic acid (DNA) (Avery et al., 1944; McCarty, 2003). This genetic information is carried between cells during cell division, and from one generation to the next through reproduction; it determines the characteristics of individual cells and whole organisms. Therefore, understanding the structure of DNA can inform how different cells gain or lose their functions, how different organisms develop, and how different species evolve.

1.1 Nucleic acids

Structure of DNA

DNA is formed from monomeric subunits, nucleotides. A nucleotide is assembled from three distinct components: a phosphate ion, a sugar molecule, and a nucleobase (either a purine or pyrimidine; Figures 1.1a and 1.1b). The sugar, deoxyribose, is in a cyclic form and covalently linked with one of four cyclic bases (Watson and Crick, 1953; Saenger, 1984). This arrangement produces the four normal nucleosides: adenine (A), cytosine (C), guanine (G), and thymine (T). To form a nucleotide the sugar molecule must be phosphorylated (Saenger, 1984).

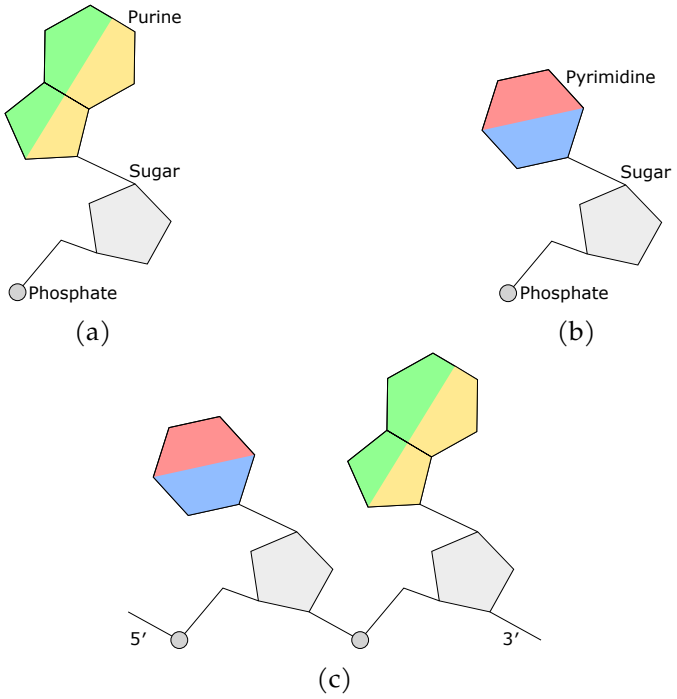


Figure 1.1: General structure of (a) purines: adenine and guanine, and (b) pyrimidines: cytosine and thymine, DNA nucleotides. (c) A polynucleotide chain containing two nucleotides. The sugar-phosphate backbone can be seen with bases protruding from it. The colours here representing the specific bases adenine, yellow; guanine, green; cytosine, blue; and thymine, red. These correspond to complementary pairing diagrams in Figure 1.2.

A polynucleotide strand is formed by covalently linking nucleotides using the sugar and phosphate molecules. This forms a "backbone" of alternating sugar-phosphate-sugar-phosphate (Figure 1.1c). Base-base hydrogen bonding occurs, between two polynucleotide strands, according to a strict rule defined by the complementary structures of the nucleobases: A binds to T (Figure 1.2a), and C binds to G (Figure 1.2b) (Alberts et al., 2017; Saenger, 1984). These two strands run antiparallel to each other and twist, forming a DNA double helix (Figure 1.2c), a double-stranded structure where each stand is complementary to the other.

The way in which the nucleotides are linked together gives a DNA strand a chemical polarity (Alberts et al., 2017). Each sugar molecule has a phosphate attached to the 5′ carbon and a space on the 3′ carbon that allows phosphate to bond there. Consequently, a polynucleotide chain will be formed of subunits all in the same orientation and each end will be distinguishable as either the 5′ or 3′ depending on the sugar molecule (Figure 1.1c).

The hydrogen bonds between the nucleobases are weaker than the covalent bonds in the sugar-phosphate backbone, this allows the DNA strands to be pulled apart

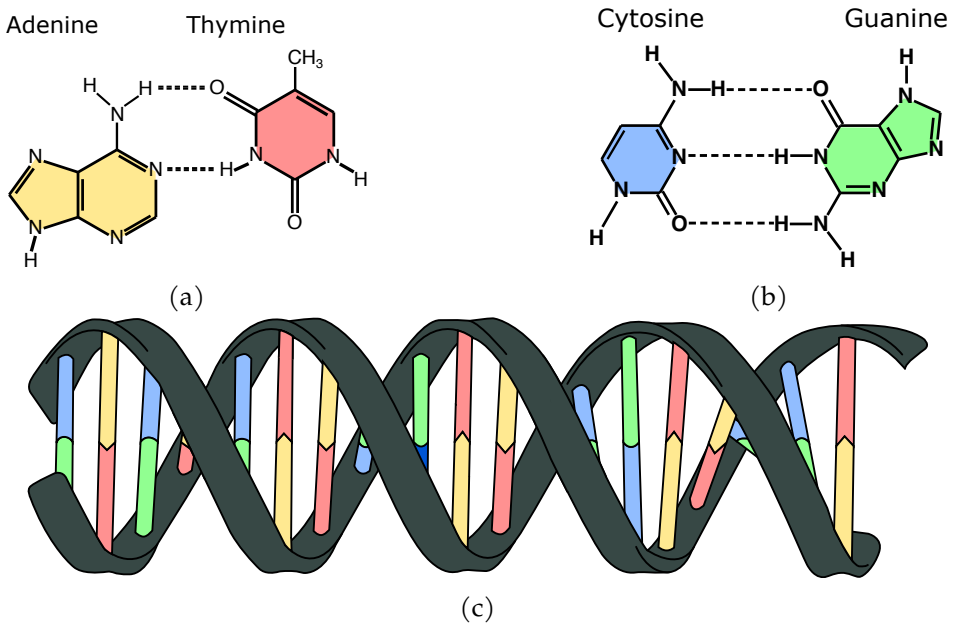


Figure 1.2: Complementary base pairing between the nucleobases, dashed lines represent hydrogen bonds. (a) base pairing between adenine and thymine; (b) base pairing between cytosine and guanine. (c) Double stranded representation of DNA; Adapted from: Difference between DNA and RNA; used under CC BY 3.0.

without disrupting the order of the bases (Alberts et al., 2017). Each strand then can serve as a template for the synthesis of a fresh DNA strand. Through this process genetic instructions, in the form of nucleic acids, can be stored, retrieved, and translated within an organism. Moreover, this hereditary information is passed from one generation to the next.

Structure of RNA

Like DNA, RNA is a linear polymer comprised of different nucleotide monomers covalently linked by phosphodiester bonds. Unlike DNA, the nucleotides are ribonucleotides (using ribose, not deoxyribose) and the base thymine is replaced by uracil (Saenger, 1984). Uracil, like thymine, can pair by hydrogen-bonding with adenine so the complementary base-pairing properties of DNA also apply to RNA. Moreover, strands of RNA exhibit the same chemical polarity as DNA, having a 5′ and a 3′ end.

Despite these slight differences, DNA and RNA differ in overall structure. While DNA occurs in a double-stranded helix, RNA is single-stranded. Though RNA strands can fold into complex three-dimensional shapes, which allows some RNA molecules to have precise structural and catalytic functions (Walter and Engelke, 2002; Raina and Ibba, 2014).

1.2 Sequencing nucleic acids

The central dogma of molecular biology is "DNA makes RNA, and RNA makes protein"[1]. That is, the sequence of the DNA monomers determines the order of the subsequent RNA molecule, which in turn determines the sequence of amino acids in the final protein. As DNA and RNA are capable of storing and transferring genetic information, understanding how this information is encoded is critical.

There are currently three generations of sequencing technologies (Figure 1.3). The "first-generation" methods, Sanger sequencing (Sanger and Coulson, 1975; Sanger et al., 1977) and Maxam-Gilbert sequencing (Maxam and Gilbert, 1977). "Next-generation" methods, mostly rely on the same concepts as first-generation sequencing but increased sequencing volume by introducing massively parallel sequencing. Finally, "single-molecule" (also known as third-generation) methods, incorporate the scale of next-generation technologies with single-molecule, long-read, and real-time sequencing.

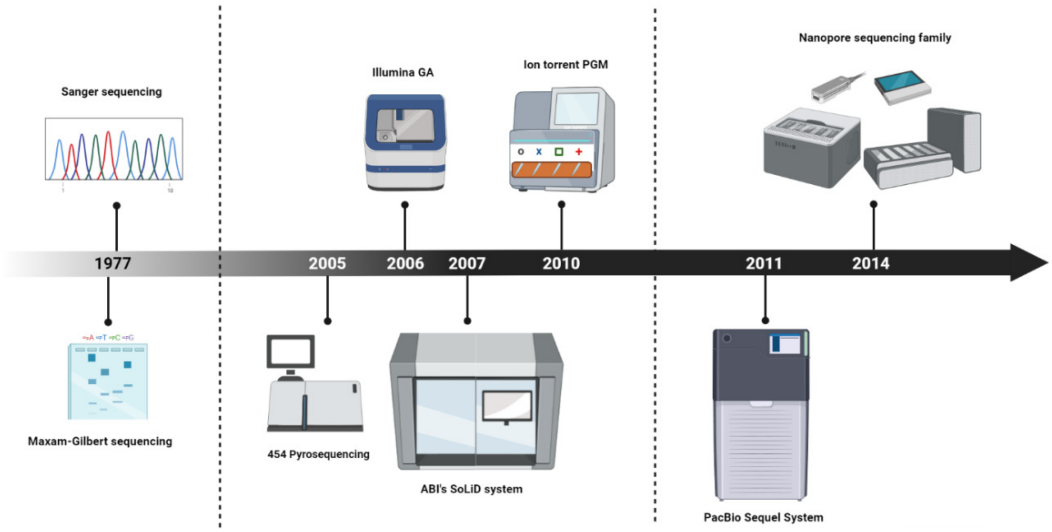


Figure 1.3: Sequencing technologies milestones. These are split into three eras, "first-generation", "next-generation", and "single-molecule". Adapted from Athanasopoulou et al. (2021)

1.2.1 First-generation sequencing

Sanger Sequencing

In 1977 Sanger et al. developed "Sanger sequencing" (also known as dideoxy or chain termination sequencing) (Sanger et al., 1977). This method uses the same

[1]More precisely: "The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information." Crick (1970)

sequence as polymerase chain reaction (PCR): where DNA molecules are denatured, a complementary DNA primer is annealed and extended using DNA polymerase. The DNA sample is divided into four separate reactions, each contains the four standard dNTPs (dATP, dCTP, dGTP, dTTP) and the DNA polymerase. In each round of primer extension small amounts of dideoxynucleotides (ddNTPs) are included. These ddNTPs randomly terminate the extension as they lack the 3′ hydroxyl group required for creating the phosphodiester bond in the DNA backbone. By using labelled ddNTPs (either radiolabelled or a fluorescent dye) and the exponential amplification of PCR an extremely large number of fragments, of varying size, can quickly be generated. The fragment length and the label distinguish which base corresponds to this fragment size.

In the original method, sequence is determined by polyacrylamide gel electrophoresis, using a lane per ddNTP used. Modern Sanger sequencing platforms, for example the Applied Biosystems 310 Genetic Analyzer, uses high-resolution capillary electrophoresis to separate fragment sizes. A laser is used to excite the fluorescent labels as fragments exit the capillary and the terminating colour is detected. This generates a readout, or a Sanger sequencing "trace", that can be basecalled and assigned error probabilities (Ewing et al., 1998).

Sanger sequencing is typically used for fragments of 500–700 bp, though sequences of up to 1 kbp can be sequenced (Shendure and Ji, 2008). In addition, modern Sanger sequencing has high basecall accuracies, as high as 99.999% (Shendure and Ji, 2008).

1.2.2 Next-generation sequencing

Next-generation sequencing technologies are typically characterised by their use of massively-parallel sequencing arrays. These arrays carry out a cyclic sequencing procedure, conventionally an enzymatic manipulation followed by sensing. The enzymatic manipulation is typically a cyclic reversible termination (Illumina) or a single-nucleotide addition (454, Ion Torrent) (Goodwin et al., 2016). Sensing most frequently uses imaging to detect fluorescence, though one example (Ion Torrent) uses pH.

Library preparation is accomplished by random fragmentation of DNA, followed by ligation of common adaptor sequences. Colonies of amplicons are then generated by amplification using techniques such as emulsion PCR (Dressman et al., 2003) or bridge PCR (Adessi, 2000; Fedurco, 2006). These amplification methods result in spatially clustered libraries, either to a single location on a glass slide (bridge PCR) or to the surface of a bead (emulsion PCR). Like Sanger sequencing next-generation

platforms rely on sequencing by synthesis, either using a polymerase (Mitra et al., 2003) or a ligase (Shendure et al., 2005) for the elongation step.

454 Pyrosequencing

Pyrosequencing, developed in the late 1990s (Ronaghi et al., 1996, 1998), is a technique that detects fluorescent bursts generated as a by-product of DNA strand synthesis. A library is prepared by ligating fragments to genomic DNA then denaturing to create single-stranded DNA. Mixing these fragments with micron-scale beads and carrying out emulsion PCR results in millions of copies of each DNA template on each bead (Figure 1.4a; Siqueira et al. (2012); Goodwin et al. (2016)). The beads are then transferred to a picotiter plate such that each well on the plate is occupied by a single bead.

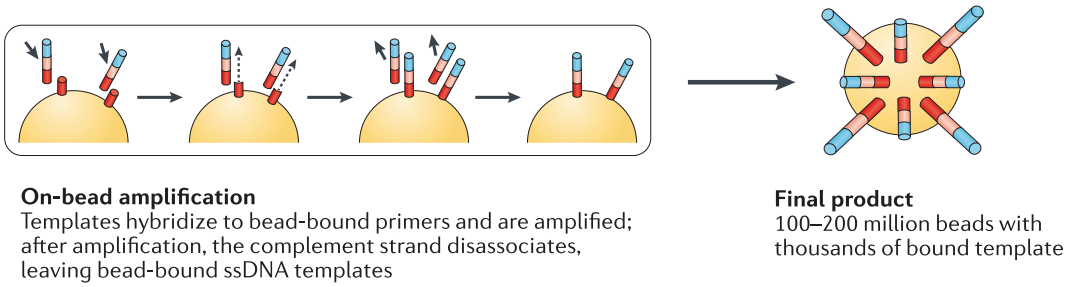
Sequencing begins with the addition of enzymes and luciferin. Then, dNTPs are added one at a time; the dNTPs are incorporated into the template strands releasing pyrophosphate. Pyrophosphate is subsequently converted into adenosine triphosphate (ATP) by the enzyme ATP sulfuryase. ATP is used by the enzyme luciferase to oxidise luciferin releasing light (McElroy and Green, 1956). The burst of light is captured using a charge-coupled device that records the wells that fluoresce on each cycle. Finally, the enzyme apyrase degrades remaining ATP and unincorporated dNTPs for the next dNTP to be added. As nucleotides are added in a fixed and known order 454 data can be basecalled by recording the order and intensity that each well fluoresces (Beuf et al., 2012).

Illumina

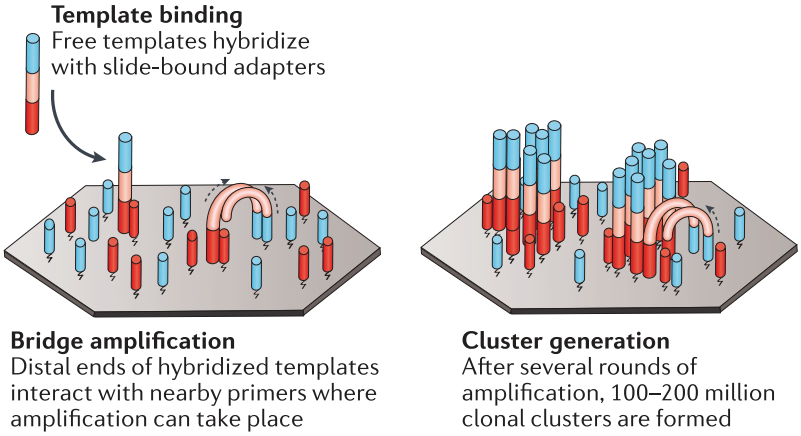
Libraries for Illumina sequencers are prepared by fragmenting the DNA sample into short (<300 bp) sections. Adapters are then ligated on to the ends of the fragments. The library can then be loaded on to a flow cell, a glass slide with eight lanes, for clustering. Each lane is a channel coated with a "lawn", composed of oligonucleotides that are complementary to one of the ligated adapters (Figure 1.4b). Bridge amplification then occurs where the DNA fragments bind to the lawn and bend to create a single-stranded bridge that is amplified by PCR. After repeated amplification clusters containing millions of copies of the original input are tightly packed together (Goodwin et al., 2016).

Sequencing is conducted by washing fluorescently labelled dNTPs in successive rounds (one for each base). These dNTPs, like in Sanger sequencing, are chain-terminating so that only one nucleotide is bound per cycle. Imaging then takes place

where each cluster of strands fluoresces. After imaging the flow cell is washed removing only the fluorophore and terminator leaving the incorporated nucleotide in place (Goodwin et al., 2016). The data from the fluorescence is basecalled, producing a read from each cluster as well as error probabilities.



(a)



(b)

Figure 1.4: (a) Fragmented DNA templates are ligated to adapter sequences alongside a bead covered with complementary adapters. PCR is carried out, covering each bead with thousands of copies of the same DNA sequence (b) Solid-phase bridge amplification, DNA fragments are ligated to adapters and bound to a primer on a solid support. The free (unbound) end interacts with nearby primers, forming a bridge. PCR is then used to create a second strand. Adapted from Goodwin et al. (2016).

Ion Torrent

Sequencing of DNA is done using a semiconductor chip that has millions of wells (Rothberg et al., 2011). These wells capture the change in pH that DNA polymerisation generates and translates it into basecalls.

The sequencing process starts when a sample of DNA is fragmented and then each fragment is attached to its own bead. The beads then undergo emulsion PCR to amplify each fragment to millions of copies (Goodwin et al., 2016). These beads then flow across the chip each depositing into a well. Then dNTPs are added one at

a time, whenever a nucleotide is incorporated into a single strand of DNA a hydrogen ion is released. These hydrogen ions are sensed by the semi-conductor chip and recorded. As the dNTPs are non-blocking runs of homopolymers can be incorporated in a single step. After the introduction of a single dNTP the unincorporated bases are washed away allowing the next base to be added.

Ion Torrent devices have an accuracy of ~98–99% based on reads of ~200 b (Quail et al., 2012). As the pH change is imperfectly proportional to the number of nucleotides incorporate Ion Torrent has limited accuracy measuring homopolymer lengths (Goodwin et al., 2016).

Limitations of next-generation sequencing platforms

Next-generation sequencing platforms offer a trade-off between yield and read length. Most sequencing platforms offer shorter average read lengths (30–400 b) than Sanger sequencing (500–1000 b; Hert et al., 2008). These shorter read lengths limit the available experiments that these methods are applicable to. For example, it is still difficult and time consuming to assemble a genome *de novo* into high-quality genomes from short read fragments (Salzberg et al., 2011; Chin et al., 2014). These challenges are a critical issue for large genome assembly as short reads result in highly fragmented assemblies due to regions with unsolvable repetitive or high GC content (Alkan et al., 2010; Mardis, 2013; Petersen et al., 2017). As a result many whole genome sequencing projects, which use next-generation technologies, focus on comparisons with existing reference genomes — re-sequencing (Alkan et al., 2010; Lischer and Shimizu, 2017); for example studies using UK Biobank samples (Backman et al., 2021).

1.2.3 Single-moleccule sequencing

Compared to previous generation sequencing technologies, single-molecule sequencing technologies can sequence kilobase length sequences (>10 kb) directly from native (unamplified) DNA. These long reads are achieved using direct detection of nucleotides in the target DNA molecules without any [clonal] amplification step required. Long-read sequencing is particularly useful for genotyping as it can allow for phasing alleles and address issues with *de novo* assembly (Stancu et al., 2017; Loose, 2017).

Pacific Biosciences

Pacific Biosciences (PacBio) published their method for real-time sequencing of single molecules in 2009 (Eid et al., 2009). This method uses a single polymerase enzyme to perform uninterrupted synthesis of a single DNA template molecule incorporating fluorophore labelled dNTPs. PacBio "SMRT"[2] sequencing is performed in a nanoscale chamber called a Zero-Mode Waveguide (ZMW; Figure 1.5a). A ZMW, like the wells in 454 Pyrosequencing and Ion Torrent can be observed from the underside. The base of the ZMW is glass and constructed to act as a microscope, capable of focusing on a 20 zeptolitre (10^{-21} litre) volume (Eid et al., 2009). Unlike short sequencing-by-synthesis platforms PacBio fix their polymerase to the bottom of the ZMW, this keeps the site of dNTP incorporation stationary improving focusing on single molecules (Eid et al., 2009; Goodwin et al., 2016).

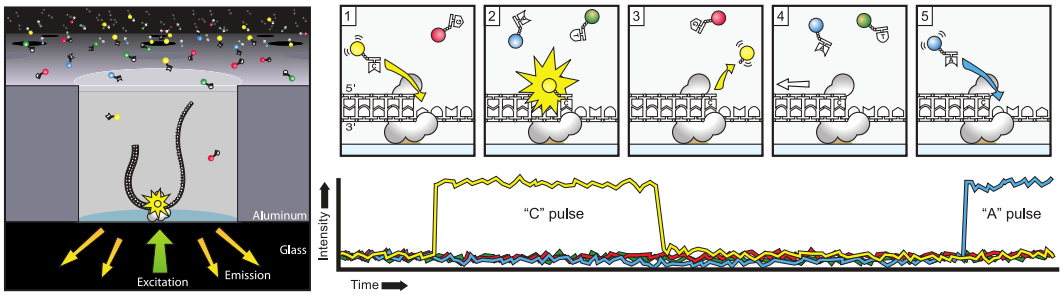
Target molecules are prepared by fragmenting and ligating a pair of hairpin adapters to each end, creating a topologically circular molecule (Logsdon et al., 2020). Then a polymerase is bound and the molecule is loaded into on to a "SMRT cell" and into the ZMW via diffusion or magnetic beads. The PacBio platforms (Sequel I and Sequel II) have two sequencing modes: Continuous Long Reads (CLR) and Circular Consensus Sequencing (CCS, also called HiFi reads). CLR mode generates data from a single pass of large (>25 kb) molecules. CCS mode exploits the circular molecule created by the hairpins and sequences multiple copies of each molecule up to 25 kb in length (Pacific Biosciences, 2021). During sequencing, the polymerase removes the fluorophores from dNTPs so that they can be incorporated into the strand being synthesised. A laser and camera beneath the SMRT cell capture the colour and duration of fluorescence and use this data for basecalling (Goodwin et al., 2016) (Figure 1.5b).

A single SMRT cell in a Sequel II has an average throughput of ~50–100 Gb for CLR reads and ~15–30 Gb for HiFi (CCS) reads (Logsdon et al., 2020). The read accuracy of reads from the Sequel II is 8–13 % for CLR reads and >99 % for HiFI reads (Logsdon et al., 2020). In addition, it is possible to indirectly sequence RNA molecules for full-length characterisation (Sharon et al., 2013).

Oxford Nanopore Technologies

All of the sequencing techniques described so far require an enzyme to synthesise a complementary strand of DNA such that individual bases can be detected for sequencing. However, Oxford Nanopore Technologies (ONT) sequencing methods

[2]Single Molecule Real-Time, Pronounced "smart"



(a) (b)

Figure 1.5: (a) DNA polymerase is immobilized at the bottom of a ZMW, which is illuminated from below by laser light. (b) dNTP incorporation cycle. (1 and 2) A nucleotide is incorporated with the template molecule causing an elevation of the fluorescence output on the corresponding color channel. (3) The dye-linker is cleaved along with the phosphate chain and diffuses out of the ZMW. (4 and 5) The polymerase translocates to the next position, and another nucleotide can bind creating a new pulse. Adapted from Eid et al. (2009).

do not; instead they directly sense the modulations in electronic current a polynucleotide strand exhibits when passing through a nanopore (Branton et al., 2008). A more detailed description of nanopore sequencing is in Section 1.3.

1.3 Nanopore sequencing

1.3.1 A brief history

In 1989, Professor David Deamer proposed that a protein channel could be incorporated into the membrane of a liposome, and that ATP could then pass across the membrane. In addition, if ATP can pass through this channel so could other dNTPs and so could DNA. And further, if each nucleotide produces a specific blockade of ionic current as it passed through the channel, they can be discriminated.

Later, Deamer, Branton, and Kasiannowicz used alpha haemolysin (α-HL), a pore-forming protein secreted by *Staphylococcus aureus*, to detect DNA translocation through an α-HL nanopore (Kasianowicz et al., 1996). However, these translocations were too fast typically taking <1.3 µs for a ~210 b long strand of single-stranded poly(U) RNA. This was followed by using an engineered "DNA-nanopore" complex, developed by the Bayley lab, for the detection of single-stranded DNA molecules for the detection of specific sequences such as antimicrobial resistance (AMR) genes (Howorka et al., 2001). Though, these "DNA-nanopore" complexes required specific engineering that is complementary to each potential analyte. The next major breakthrough came from the Akeson lab. This was the use of a molecular motor,

in the form of phi29 DNA polymerase, that controlled the speed of translocation (Lieberman et al., 2010; Cherf et al., 2012).

In parallel, a company "Oxford Nanolabs" was formed in 2005 which would later become "Oxford Nanopore Technologies". The chemistry and scale of nanopore sequencing was refined and adapted into a product that was first shown at the AGBT [Advances in Genome Biology and Technology] conference in 2012 (Brown, 2015). Finally, the MinION sequencer was released via the "MinION Access Programme" in 2014.

1.3.2 Nanopore sensing

Nanopore sensing is a method that is able to detect DNA and RNA molecules based on the decrease in ionic current that the molecule produces when interacting with the lumen of a nanopore. This is similar to Coulter counting used for detecting analytes in electrolytes (Coulter, 1953; Bezrukov, 2000). The principle of Coulter counting alongside developments in electrophysiology techniques reduced the target analyte size from millimeters to nanometres, going from cells to individual biomolecules (Wanunu, 2012).

In nanopore devices a salt solution is divided into two wells, *cis* and *trans*, divided by a thin insulating membrane. Protein nanopores that span the membrane connect the *cis* and *trans* wells, and are the only path between the compartments. Electrodes placed in each compartment create a potential difference across the membrane (Clarke, 2019). This difference in voltage causes ions to flow through the pore by electrophoresis, which can be measured by an amplifier (Figure 1.6a). As DNA is negatively charged it is also drawn through the pore, while in the lumen of the pore the DNA reduces the flow of ions (Figure 1.6b), creating blockades. These blockades, called resistive pulses, can be measured and characterised by their amplitude and duration (Kasianowicz et al., 1996). Once the molecule has translocated the pore the current returns to it's open value current until another molecule occupies the channel again (Figure 1.6b).

The number of translocations is directly related to the concentration of DNA in the *cis* well, therefore sequencing libraries with fewer molecules will see less frequent translocations. Moreover, these translocations are too fast to resolve individual nucleotides on a strand of DNA with the speed of sequencing being ~1–7 μs (Kasianowicz et al., 1996). Both of these issues are resolved by the addition of molecular motors, such as DNA polymerase from phi29, which has been successfully used to control the rate of translocation (Lieberman et al., 2010; Cherf et al.,

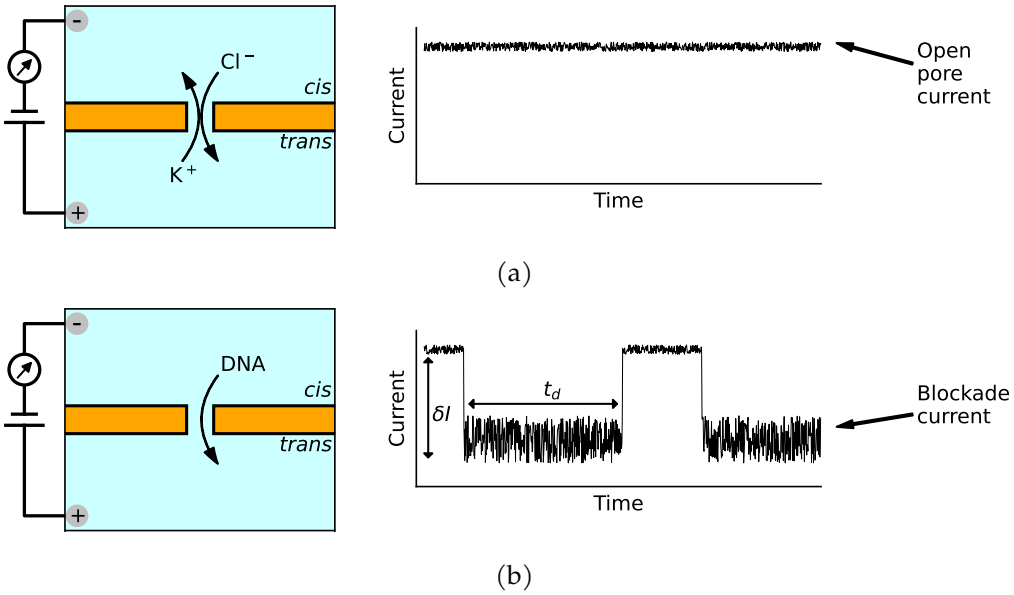


Figure 1.6: (a) Applying a voltage across a nanopore causes ions to migrate towards the membrane, as these ions pass through the nanopore an electric current is measured using an ammeter. (b) When analytes, such as DNA, are added to the *cis* chamber they diffuse towards the nanopore and enter it. This results in measurable "resistive pulses". These samples are characterised by their dwell time (t_d) and their event amplitude (δI).

2012) and the use of hydrophobic anchors to concentrate molecules at the membrane.

With single base resolution possible (Cherf et al., 2012; Manrao et al., 2012) ionic currents for known strands of polynucleotides could be characterised to allow the development of basecalling algorithms. The identity of the nucleotides within the constriction site of the nanopore specifically determine the current level at that point along the strand. The raw — signal level — data is recorded by a picoammeter. These raw signal data are typically called "squiggles" they are a time-series of 16-bit integers that are sampled at 4 kHz. As DNA translocates at ~400–450 b/s there are ~9–10 data points associated with each individual nucleotide.

The ASIC (Application Specific Integrated Circuit) is a high density array of low-noise amplifier circuits. It is used to measure the current flow between each *trans* compartment electrode and the common *cis* chamber electrode. The ASIC can also receive commands from the controlling computer to control the sensor array (Clarke, 2019). Finally, the ASIC is able to use the applied potential, that draws DNA through the nanopore, to eject any DNA or contaminants by momentarily reversing the applied potential across an individual pore. Furthermore, the ASIC is able to use the applied potential, that draws DNA through the nanopore, to eject any DNA or

contaminants by momentarily reversing the applied potential across an individual pore. As data can be analysed whilst a molecule is still translocating a pore these mechanisms allow for reads to be selectively ejected based on the first few bases in the strand.

Nanopores and motors

There are currently two revisions of nanopores available from ONT R9 (released in 2016) and the newer R10 (released in 2019). The R9 nanopore (Figure 1.7a) is currently on version R9.4.1 and is a mutant form of the CsgG lipoprotein (Clarke, 2019). The newer R10 nanopore (Figure 1.7b), currently on version R10.4, consists of two proteins, CsgF and CsgG, covalently attached together creating a pore with two sensing regions (der Verren et al., 2020). This dual sensing region allows for the length of homopolymers to be more accurately characterised and improves the signal-to-noise ratio of the nanopore sensor (der Verren et al., 2020).

The amount of current that can pass through a nanopore depends upon the nucleobase that is currently occupying the lumen. In reality it is not a single nucleotide that creates the blockade but a group, known as a "*k*mer". The R9 nanopore has a "sharp" reader head (Figure 1.7a; Branton, 2019) that resulted in a sensing zone of ~4–5 nt (Branton, 2019). R10 adds a second reader head that maintains the ~0.75 nm radius of the first reader head (der Verren et al., 2020).

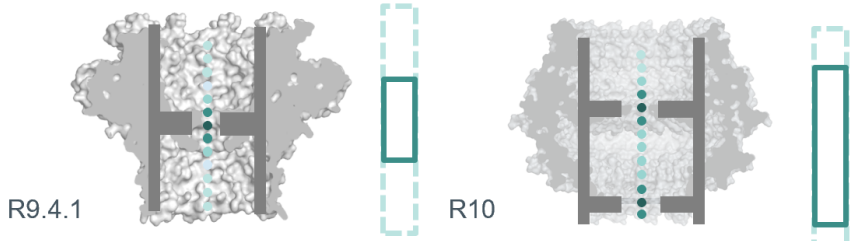


Figure 1.7: Cross section of (a) R9.4.1 nanopore and (b) R10 nanopore. The sensing zone of each nanopore version is represented by the darker coloured dots at the constriction point of each nanopore. Adapted from Oxford Nanopore Technologies, (2019)

As previously mentioned processive enzymes, motor proteins, such as polymerases and helicases were essential in slowing the rate of DNA translocation (Kasianowicz et al., 1996). Moreover, the rate of translocation depends on the motor protein selected with rates ranging from 10–1000 b/s (Byrd and Raney, 2019). The successful detection of ionic current from a polynucleotide used DNA polymerase and pulled the strand out of the pore (Manrao et al., 2012).

An alternative to a polymerase is a helicase enzyme. First, helicases are capable of separating double-stranded DNA into single-stranded DNA; crucially helicases are also capable of "unzipping" duplex RNA molecules and can move both 5′-3′ and 3′-5′ (Byrd et al., 2012; Byrd and Raney, 2019). This allows a single motor protein to be used for all polynucleotide sequencing. Secondly, the motor protein is too large to pass through the nanopore and binds tightly to polynucleotide strand; as such it makes an good quality brake (Byrd and Raney, 2019). Finally, the helicase is only activated when it is in contact with a nanopore in the presence of ATP (Byrd and Raney, 2019).

ONT use proprietary motor proteins in their sequencing systems. When the first nanopore sequencer was made available the nanopore used was R7 and the motor protein was called "E5". This combination permitted sequencing DNA at ~30 b/s, this was improved to ~70 b/s by 2015 (ONT, 2021). Finally with the introduction of the R9 pore and further improvement of the motor protein (E8) the speed of sequencing DNA reached ~450 b/s with a direction of 5′–3′ (ONT, 2021). For direct RNA sequencing the motor protein "M1" is used, with a speed of ~70 b/s and a direction of 3′–5′ (Heron, 2019).

ONT Platforms

ONT launched the first commercial nanopore sequencing device, the "MinION" in 2014 (Figure 1.8; Jain et al. (2016)). The MinION is a pocket-sized, portable DNA sequencer weighing only 90 g. It operates with a consumable flow cell that contains a sensing array of 2,048 ~1 nm biological nanopores. Nanopores are controlled in groups of four, allowing 512 pores to simultaneously report current.

In addition to the MinION ONT released the GridION and the PromethION in 2017 and 2019 respectively. The GridION builds in support for five MinION flow cells while the PromethION uses a larger flow cell design with 3,000 nanopores. More recent additions include the MinION Mk1C, which incorporates a Jetson TX2 embedded computer (with GPU), that can manage a single MinION flow cell. Finally, the P2, a self-contained device with GPU and capacity for two PromethION flow cells. Compute Unified Device Architecture (CUDA) enabled GPUs accelerate basecalling by using their highly parallel architecture to process large blocks of signal data in real-time, which is essential for live basecalling.

The sequencing control software, MinKNOW, continuously processes the incoming raw signal for every sequencing nanopore on the flow cell. It is analysing these signals to identify when strands of DNA enter and exit each nanopore and to check whether a nanopore is blocked and may require unblocking. As a result, MinKNOW

writes segmented sections of raw signal data to read FAST5 files, with each section representing a single molecule. These read files contain the necessary information for a basecaller (Guppy) to convert the recorded current data into FASTQ format; an essential step for bioinformatics analysis.

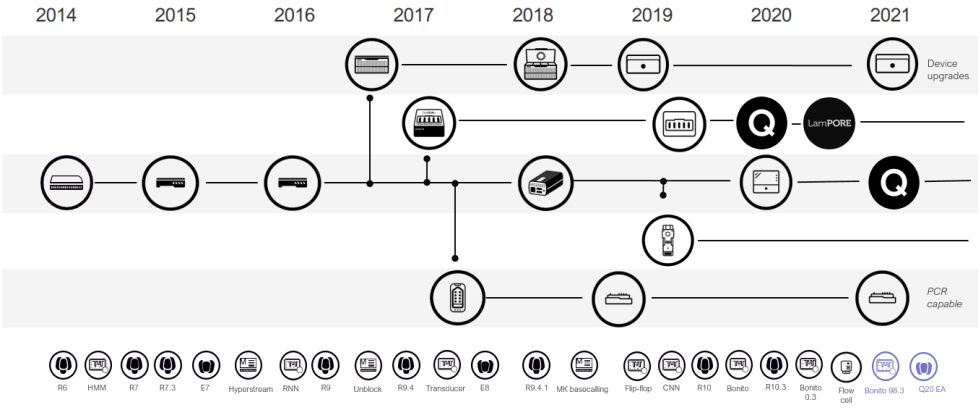


Figure 1.8: Timeline of ONT devices starting in 2014 with the MinION and branching into PromethION, GridION, and MinION Mk1C. From https://nanoporetech.com/about-us/news/blog-you-cant-put-label-innovation-or-can-you

Basecalling

As data measured from nanopore sequences is delivered from the device as an electrical current, "squiggle", it must be decoded into bases for use with downstream analysis tools. When the MinION was first released basecalling was performed using Hidden Markov Model (HMM) methods on a cloud compute platform called Metrichor; requiring an active internet connection so that raw signal could be uploaded and decoded data downloaded. Later an open source basecaller, Nanocall (David et al., 2016), which used an HMM with comparable performance to Metrichor was released allowing offline basecalling and analysis.

When basecalling with an HMM first the raw signal is segmented into *events*. Each event ideally corresponds to an individual *k*mer and so subsequent events will only differ by a single base. In Nanocall the HMM has a series of states that represent all possible *k*mers (Figure 1.9a). During basecalling, the most probable path through these states is calculated by Viterbi decoding. The path is converted to nucleotide sequence by overlapping consecutive states. Consequently, homopolymer repeats of a length greater than the size of the *k*mer for this nanopore cannot be detected.

An alternate approach to HMMs is the use of Recurrent Neural Networks (RNNs). This was first publicly implemented by DeepNano (Boža et al., 2017) which used

segmented data and a RNN for basecalling. These RNNs do not rely on *k*mers for classifications, taking into account both upstream and downstream event information through the use of a bidirectional RNN (Figure 1.9b).

Early basecallers used segmented event data as input to determine DNA sequence; however, current basecallers use raw current signal as input. For example, BasecRAWller (Stoiber and Brown, 2017) uses two separate RNNs (Figure 1.9c). The first RNN predicts the probability that a raw signal corresponds to a new *k*mer and the identity of the *k*mer. The raw signal is then segmented and the *k*mer probabilities are averaged over the segments. The second RNN then predicts the final DNA sequence. The use of long-short-term-memory (LSTM) allows information to only pass forwards which allows BasecRAWller to keep up with reads in real-time (Stoiber and Brown, 2017).

Unlike BasecRAWller, Chiron (Teng et al., 2018) does not undertake a segmentation step at all (Figure 1.9d). In Chiron, a Convolutional Neural Network (CNN) takes raw signal as input detecting local structures. The CNN outputs are passed through to a series of RNN in the form of LSTMs which pass their outputs to a Connectionist Temporal Classification (CTC) decoder for decoding to bases.

Local basecalling was integrated into MinKNOW in the form of Albacore (a transducer basecaller) while research basecallers in the form of nanonet and scrappie (ONT, 2019) were made available for testing new neural-network approaches (Wick et al., 2019).

In late 2017 ONT released Guppy, a graphical processing unit (GPU) accelerated basecaller. Which, like scrappie, is a general purpose basecaller. Guppy is RNN based basecaller that is trained using real sequencing data (Wick et al., 2019; Clarke, 2019). Guppy specifically aims for basecalling speed improvements by using the hardware features of GPUs that enable parallelisation of basecalling.

Basecaller training

Oxford Nanopore Technologies develops and trains basecaller models using data from sequencing experiments[3]. A dataset of reads is selected for using in training a model. These datasets typically contain both native and PCR-amplified reads, which are >1000 b in length, from samples including human, *Escherichia coli*, *Caenorhabditis elegans*, and the ZymoBIOMICS Microbial Community Standard[4]. By including native DNA base modifications are preserved in the training data.

[3]https://community.nanoporetech.com/technical_documents/data-analysis/v/
[4]https://www.zymoresearch.com/collections/zymobiomics-microbial-community-standards

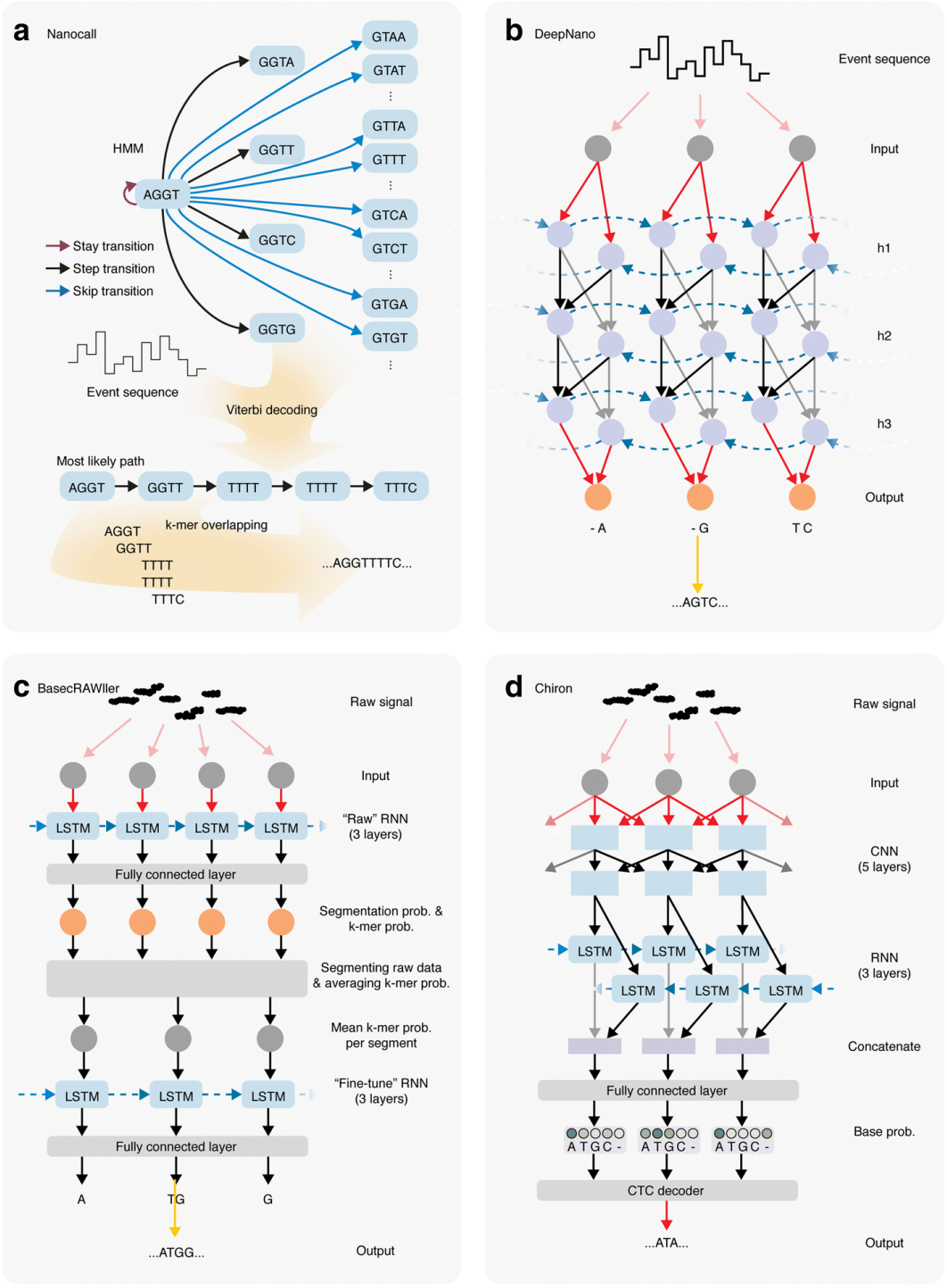


Figure 1.9: (a) Nanocall uses a Hidden Markov Model (HMM) for basecalling. (b) DeepNano was the first published basecaller to use Recurrent Neural Networks (RNN). Labels h1–h3 represent three hidden layers in the RNN. (c) BasecRAWller uses two RNNs, one to segment the raw measurements and one to infer k-mer probabilities. (d) Chiron makes use of a Convolutional Neural Network (CNN) to detect patterns in the data, followed by an RNN to predict k-mer probabilities, which are evaluated by a Connectionist Temporal Classification (CTC) decoder. Adapted from Rang et al. (2018).

Base modification

Single molecule nanopore sequencing of native DNA and RNA can detect modifications on individual nucleotides and has been show to discriminate among all five C5-Cytosine variants in synthetic DNA (Schreiber et al., 2013; Wescoe et al., 2014). Furthermore, N6-methyladenine modifications in *Escherichia coli* genomic DNA can be detected at 84–94% accuracy depending on coverage (McIntyre et al., 2017).

Milestones of Nanopore Sequencing

The portability, cost, and simple library preparation of the MinION uniquely enables rapid progression from sequence acquisition to analysis. As such the MinION and nanopore sequencing has seen widespread adoption for use in clinical settings (Votintseva et al., 2017; Leggett and Clark, 2017), in the field for pathogen surveillance and outbreak tracing (Quick et al., 2016), and environmental metagenomics on a glacier (Edwards et al., 2016). Perhaps the most extreme example is the use of the MinION sequencer on the International Space Station, which demonstrated sequencing and *de novo* assembly of lambda phage and *Escherichia coli* genomes, as well as mouse mitochondrial DNA (Castro-Wallace et al., 2017). Concluding that there was no significant difference in the quality of sequence data generated aboard the ISS and in control experiments that were performed in parallel on Earth (Castro-Wallace et al., 2017).

The scale of the projects that nanopore based sequencers have been applied to has increased in magnitude from making genome assembly more tractable for both small bacterial genomes to the human genome (Koren and Phillippy, 2015; Jain et al., 2018a). Then extending to the population level sequencing, using the PromethION sequencer, of 3,622 Icelanders (Beyter et al., 2021). In 2020, during the COVID-19 pandemic, nanopore sequencers were used throughout academic and hospital laboratories to create a large-scale network of surveillance locations for monitoring SARS-CoV-2 in the UK (Nicholls et al., 2021).

Nanopore sequencing has been demonstrated detection of cytosine methylation in genomic DNA (Simpson et al., 2017). This study developed an HMM that could distinguish cytosine and 5-methylcytosine with 82% accuracy in human genomic DNA. Similarly, Rand et al. (2016) used a HMM that distinguished cytosine and 5-methylcytosine and 5-hydroxymethylcytosine with 80% accuracy, but in synthetic DNA.

Read Length Nanopore read lengths substantially exceed those of other sequencing platforms. Reads over 300 kb have been achieved using *E. coli* genomic DNA (Ip et al., 2015) and using human genomic DNA reads greater than 1 Mb have been sequenced with putative reads exceeding 2 Mb (Payne et al., 2018). The current record read length for nanopore sequencers is >4 Mb (ONT, 2022). These longer reads are able to span gaps in reference genomes that are highly repetitive (Jain et al., 2015, 2018b). Here reads of 36 kb and greater were used to resolve a ~50 kb gap in the human reference sequence (Figure 1.10). This gap contained a series of 4.8 kb tandem repeats of the gene CT47. Ultra-long reads are also important in improving *de novo* assembly and have been shown to double NG50 from ~3 Mb to ~6.4 Mb during the nanopore sequencing of the human genome (Jain et al., 2018a). The MinION-derived genome assembly expanded the *Caenorhabditis elegans* reference genome by more than 2.5 Mb due to more accurate determination of repetitive sequence (Tyson et al., 2018).

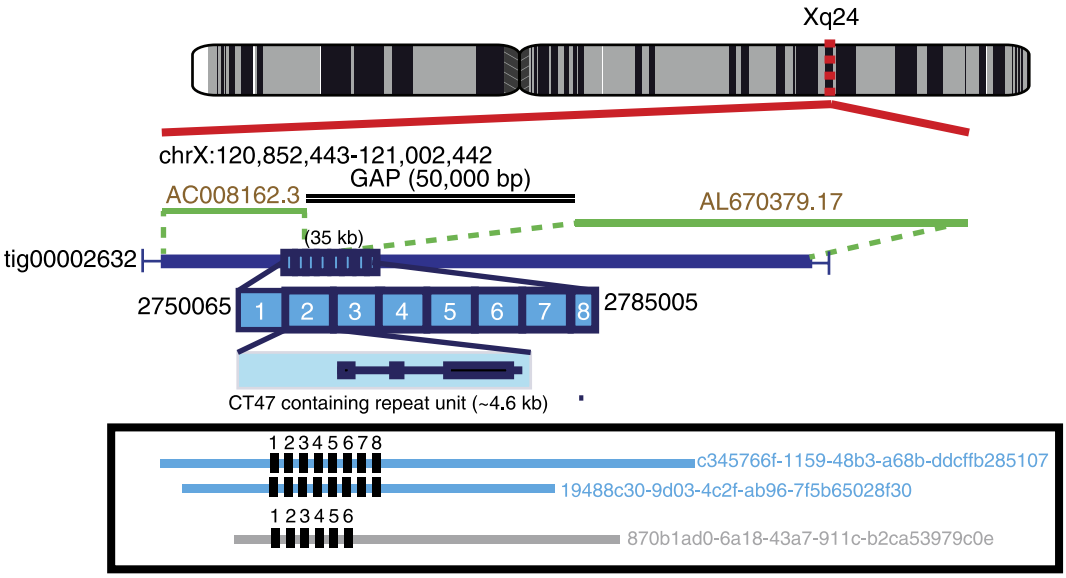


Figure 1.10: An unresolved scaffold gap on Xq24 (GRCh38; adjacent to scaffolds AC008162.3 and AL670379.17). This gap spans a ~4.6 kb tandem repeat containing CT47. This gap was closed by assembly and has eight tandem copies of the repeat. This repeat was validated by alignment of >100 kb ultra-long reads also containing eight copies of the repeat. Adapted from Jain et al. (2018a)

Read Until The combination of real-time inspection of raw signal and the ability to eject molecules, both possible while a molecule is translocating enable nanopore sequencers to be interactive. That is, the first few hundred bases of a strand of DNA can be analysed; if this region is not of interest for the particular experiment that is

being conducted it can be rejected and another molecule sampled from the library that has been loaded. This procedure continues allowing only preferred strands of DNA to be sequenced completely. This method of sequencing was first described in a London Calling talk by Clive Brown in 2015 (Figure 1.11).

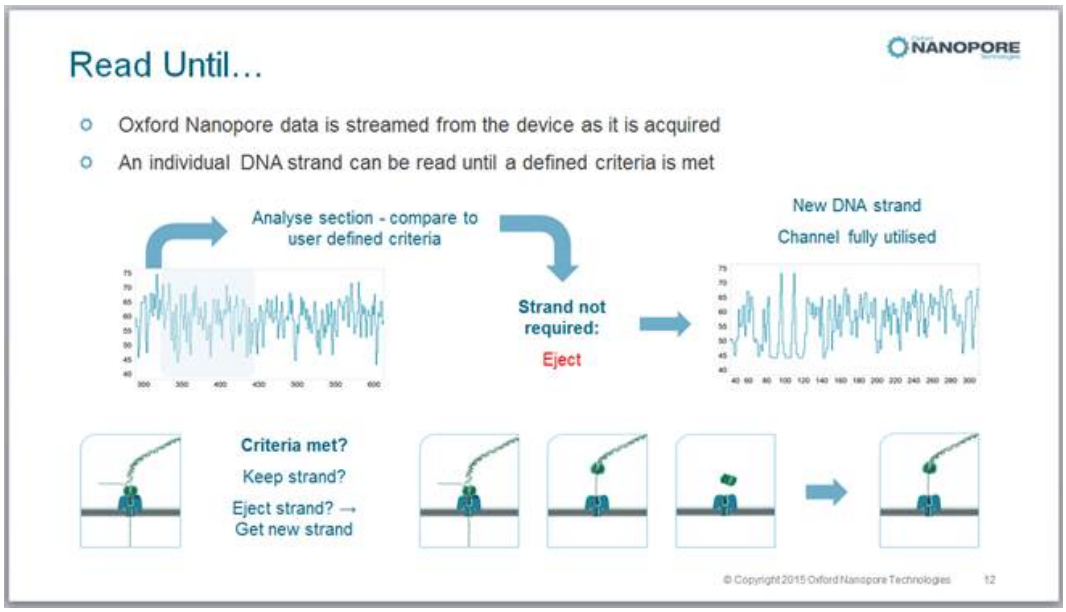


Figure 1.11: Initial presentation of Read Until from London Calling 2015. This cartoon describes how an in-progress molecule can be analysed while in the nanopore and ejected if it is not of interest, otherwise it is sequenced.

1.4 Targeted Sequencing

Targeted sequencing is commonly used in NGS workflows to remove regions of DNA that are not of interest for a particular experiment. By targeting specific regions such as exons, greater sequencing coverage can be achieved. As these approaches usually sample smaller regions of interest there is a saving in both time and cost.

1.4.1 Molecular methods of targeted sequencing

Typical enrichment methods include hybrid capture, in which DNA strands are hybridized specifically to prepared fragments that are complementary to the targets (Gnirke et al., 2009). There are commercial solutions for this from a variety of vendors [Agilent, IDT, Life Technologies]. These assays have high performance and are cost effective when used in parallel over the same genomic region in multiple samples. However, they are costly for small target regions or use with a single

sample. Moreover, the read fragment size of current technologies remain a limiting factor, producing fragments generally shorter than 1 kb.

Another technique, selective circularisation, where single-stranded DNA circles containing the target region sequences are formed, using gap-filling and ligation chemistries, in a highly specific manner creating structures with common DNA elements that are then used for selective amplification of the targeted regions.

PCR can be used on the targeted regions to amplify them; either by conducting either multiple long-range PCR in parallel, a limited number of standard multiplex PCR, or using highly multiplexed PCR. However, heavily relying on PCR amplification may result in bias for sequences that amplify well and completely eliminates any native features of the sample such as base modification.

Finally, engineered DNA-binding molecules allow for physically selecting molecules from within samples based on DNA motifs. These include: zinc finger proteins (ZFNs), transcription activator-like effector nucleases (TALENs) proteins, clustered regularly interspaced short palindromic repeats (CRISPR) system, and immunoprecipitation (ChIP) techniques. In these techniques, the CRISPR/Cas9 system is the most convenient, economical and time-efficient. CRISPR/Cas9 has been used for targeted sequencing microsatellite-spanning sequences (Shin et al., 2017) and to achieve coverage of 675× over genomic targets that enabled single-nucleotide variants, structural variations, and methylation to be assessed (Gilpatrick et al., 2020).

1.4.2 Nanopore real time selective sequencing

First demonstrated by Loose et al. in 2016, Read Until is a unique feature of ONT's real-time single-molecule platform (Loose et al., 2016). It allowed for targeted enrichment of specific genomic regions within a sample without any prior amplification.

This implementation directly compared the live "squiggle" of molecules as they passed through a nanopore against a simulated reference — a FASTA reference that had been converted into squiggle. The algorithm chosen to match the squiggles is called Dynamic Time Warping (DTW) (Kruskal, 1983). It is an audio processing algorithm that is able to compare two time-based sequences that vary in speed and amplitude. As the simulated reference is derived from the ideal sensing of *k*mers, at a specific speed, it may not always be a close match to squiggles that are seen during a sequencing experiment (Figure 1.12). Indeed, only 20% of squiggle data could be identified without normalization of the signal to account for changes in amplitude (difference from the average value) and frequency. However, after applying *z*-score normalization all 256 b sequences could be placed (Loose et al., 2016).

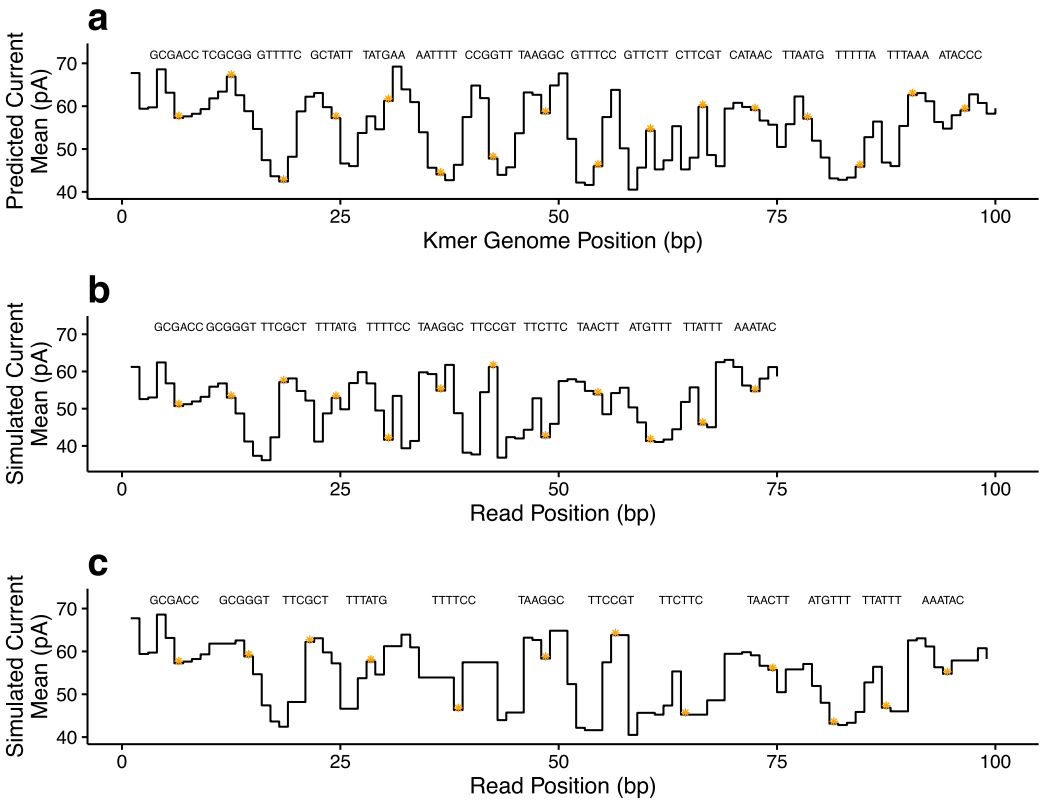


Figure 1.12: (a) shows a model squiggle inferred from the first 100 b of bacteriophage lambda. Illustrative kmers are shown above asterisked events in the squiggle. (b) an example read derived from the same 100 b region as in (a) but incorporating shift, scale and drift, along with randomly skipped kmers. (c) shows this same read, but stretched in the time axis to map directly to the original reference. Comparing (a) with (b) reveals the requirement for an algorithm such as DTW for comparing a read to reference. Adapted from Loose et al. (2016).

Using this approach two 5 kb regions of the lambda phage genome were enriched while all other regions were discarded. This experiment was run using two sequencing chemistries: SQK5, which moved DNA at 30 b/s; and SQK6, which moved DNA at 70 b/s.

While these experiments demonstrate the principle of Read Until, DTW required a lot of computational power. The experiments previously described required a 22 core server to run the analysis while another computer ran the sequencing. Moreover, as the time taken to find a match by DTW is a function of the reference length and the query length, the reference genome size that can be used was fixed at up to 5 Mb.

1.5 Aims

The aim of this research is to develop software based selective sequencing methods using ONT sequencers. This will involve addressing the following objectives:

1. An assessment of raw nanopore signal, specifically looking at how the rejection of reads impacts sequencing. In addition, evaluating the ability to use all of the raw signal data through using bulk data.
2. Following assessment of raw signal a basecalling approach for real-time read fragment classification will be developed. This real-time classification system aims to be used with in-progress reads for Read Until.
3. Implementation of a real-time selective sequencing method using the developed classification approach. Allowing for the arbitrary selection of molecules of interest from a native genomic sequencing library.
4. Evaluation of software based real-time selective sequencing approaches for some model experiments.

Chapter 2

Materials and Methods

2.1 Wet lab

2.1.1 DNA extraction

During this project we created many sequence datasets using nanopore sequencing. In Chapter 3 the human cell line GM12878 was used and DNA was extracted using the phenol chloroform protocol. In Chapters 4 and 5 the human cell lines GM12878 and NB4 were used and DNA was extracted using phenol chloroform and QIAGEN genomic tip.

Phenol chloroform

Adapted from Quick (2018) and Sambrook and Russell (2001) (chapter 6 protocol 1). For the isolation of ultra-long unfragmented high molecular weight (HMW) DNA.

1. Approximately 50 million cells are resuspended in 100 µL PBS and 10 mL Tris-Lysis Buffer (TLB) and incubated at 37 °C for 1 h.
2. Proteinase K (QIAGEN) was added and mixed by slow inversion then incubated at 50 °C for 3 h.
3. The lysate was purified using 10 mL buffer saturated phenol and phase-lock gel falcon tubes, followed by phenol:chloroform (1:1).
4. DNA was precipitated by adding 4 mL 5 M ammonium acetate and 30 mL ice-cold ethanol.
5. DNA was recovered using a glass hook and washed twice in 70 % ethanol.
6. After spinning down at 10,000 ×g, ethanol was removed followed by 10 min drying at 40 °C.
7. 150 µl Elution Buffer was added to the DNA and left at 4 °C overnight to resuspend.

QIAGEN genomic tip

For the isolation of non-ultra-long genomic DNA, that is sized up to 150 kb with an average length of 50–100 kb, QIAGEN genomic tip was used. Cells were lysed and cellular proteins are initially digested in the appropriate lysis buffer. Lysates are then loaded into a column that binds DNA allowing other cell components to pass through. Finally, pure DNA is eluted and precipitated in isopropanol before drying and resuspension in Tris buffer.

Shearing DNA

Genomic DNA was mechanically sheared to fragments of ~20 kb using a g-TUBE (Covaris) by spinning at the manufacturer's recommended speed for the mass of input DNA for one minute.

2.1.2 RNA extraction

Adapted from Workman et al. (2019).

1. ~5×10^7 cells, in a frozen pellet, were resuspended in 4 mL TRI-Reagent (Invitrogen AM9738), vortexed immediately, and incubated at room temperature for 5 min.
2. Either 400 µL 1-Bromo-3-chloro-propane or 200 µL Chloroform was added for each 1 mL in the resuspended sample, vortexed, and incubated at room temperature for 5 min.
3. Then, vortexed again and centrifuged for 10 min at 12,000 ×g at 4 °C.
4. The aqueous phase was pooled in a LoBind Eppendorf tube and combined with an equal volume of isopropanol, mixed, and incubated at room temperature for 15 min.
5. Then centrifuged for 15 min at 12,000 ×g at 4 °C.
6. The supernatant was removed and the RNA pellet was washed with 750 µL 80 % ethanol and then centrifuged for 5 min at 12,000 ×g at 4 °C.
7. The supernatant was removed and the pellet was air-dried for 10 min, resuspended in nuclease-free water with a final volume of 100 µL for quantification and either storage at −80 °C or further poly(A) purification.

Poly(A) Selection

Using RNA from the previous step, 100 µg aliquots were diluted in 100 µL nuclease-free water. Poly(A) RNA were selected using NEXTflex Poly(A) Beads (NOVA-512980) and eluted into nuclease-free water and stored at −80 °C.

2.1.3 DNA and RNA quantification

Following isolation DNA and RNA were quantified for purity and concentration. Purity of DNA and RNA were roughly quantified using the NanoDrop 2000 spectrophotometer (Thermo Fisher) using the A_{260}/A_{280} ratio, aiming for values of ~1.8–2.0 for DNA and ~2.0–2.2 for RNA (as alkaline solutions will over-represent A_{260}/A_{280} values by ~0.2–0.3 [Wilfinger et al., 1997]). Deviation from these values is indicative of protein or phenol contamination. Concentration of DNA and RNA was assessed using either the dsDNA or RNA high-sensitivity assay on a Qubit fluorometer (Thermo Fisher). All quantification steps were carried out in accordance with the manufacturer's protocols.

2.1.4 Library preparation

Ligation sequencing kit

SQK-LSK109 is the ligation sequencing kit from ONT. This sequencing kit is used to prepare double stranded DNA for sequencing, taking roughly 1 h. DNA ends are repaired and dA-tailed, and then sequencing adapters are ligated onto the prepared ends (Figure 2.1a).

Rapid sequencing kit

SQK-RAD004 is the rapid sequencing kit from ONT. This kit generates sequencing libraries from extracted gDNA in 10 min using a two-step protocol (Figure 2.1b). A transposase simultaneously cleaves template molecules and attaches tags to the cleaved ends; sequencing adapters are then added to the tagged ends ready for sequencing.

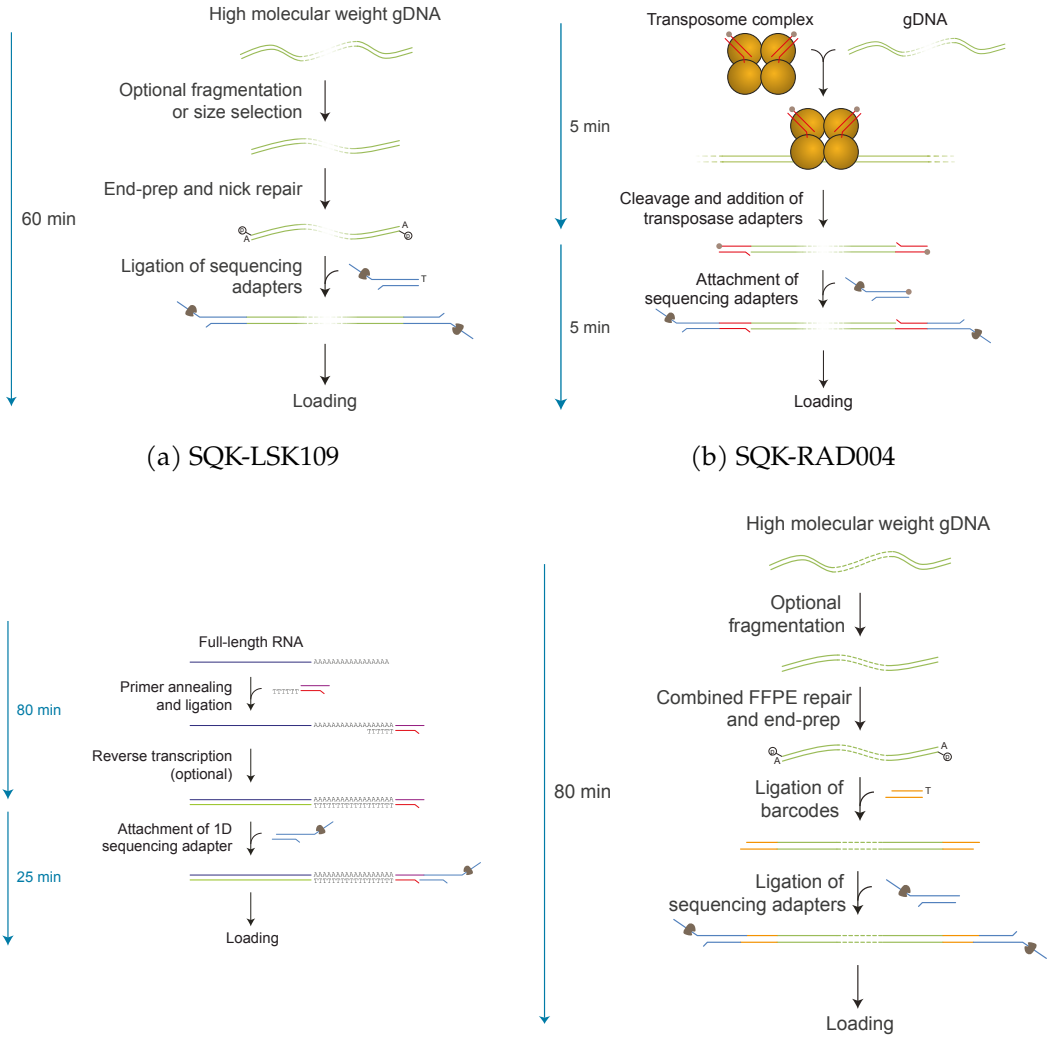
Direct RNA sequencing

SQK-RNA002 is the direct RNA sequencing kit from ONT. It is used to prepare any RNA with a 3` poly(A) tail for sequencing sequencing Figure 2.1c.

The Direct RNA sequencing protocol contains an optional reverse transcription step. The synthesised cDNA strand is not sequenced but significantly improves sequencing output.

DNA barcoding

Barcoding tags the ends of DNA with unique molecules, this allows samples to be multiplexed on a single device (Figure 2.1d). The process of attaching barcodes is relatively simple process, it is very similar to the SQK-LSK109 protocol. First, DNA ends are repaired and dA-tailed. Then, a unique complementary barcode adapter is ligated to the dA tail. Samples can now be pooled for sequencing adapter ligation, before loading and sequencing.



(a) SQK-LSK109 (b) SQK-RAD004

(c) SQK-RNA002 (d) Barcoding

Figure 2.1: Library preparation and barcoding workflows. (a, b, d) SQK-LSK109, SQK-RAD004, and barcoding uses DNA extracted as in Section 2.1.1; (c) SQK-RNA002 RNA extracted as in Section 2.1.2.
In SQK-LSK109 (a), DNA undergoes optional size-selection, blunt ends are dA-tailed (end-prep) and sequencing adapters ligated.
In SQK-RAD004 (b), DNA undergoes a simultaneous double strand cleavage and tag attachment. In SQK-RNA002 (c), RNA adapters are ligated to RNA molecules followed by reverse transcription. When barcoding (d), DNA undergoes end-prep, followed by barcode ligation. Following each of these preparation steps sequencing adapters are ligated and the library is ready for loading on to a flow cell for sequencing.

Adapted from: (a) Ligation Kit (LSK109; https://store.nanoporetech.com/uk/ligation-sequencing-kit.html), (b) Rapid Kit (RAD004; https://store.nanoporetech.com/uk/rapid-sequencing-kit.html), (c) RNA Kit (RNA002; https://store.nanoporetech.com/uk/direct-rna-sequencing-kit.html), and (d) Barcoding Kit (https://store.nanoporetech.com/uk/native-barcoding-expansion-1-12.html).

2.1.5 Running sequencing

MinKNOW

MinKNOW is the software that controls ONT sequencers and devices. It carries out several core tasks required for sequencing: data acquisition from the flow cell, real-time analysis of the data stream, base calling, controlling the flow cell and sequencer. It takes the raw data stream from every active channel and converts it into reads. This is accomplished by recognising the characteristic change in current that occurs when a DNA strand enters and leaves the pore. MinKNOW can then base call the segmented reads, and writes out the data into FAST5 and FASTQ files. The minimum specification for a computer running MinKNOW is given in Table 2.1.

Table 2.1: MinKNOW minimum IT requirements. These are lowest expected system parameters for a device running MinKNOW. For real-time basecalling a GPU is required.

Component	Specification
CPU	Modern (Intel i7/AMD Ryzen 5 or better), $\geq$ 4 cores
RAM	$\geq$ 16 GB
Storage	$\simeq$ 1 TB of fast SSD storage

MinKNOW acquires data from the sequencing device in defined chunks. This chunk size determines the frequency with which MinKNOW carries out all of it's underlying tasks and is configured prior to a run starting. These tasks include data acquisition, segmentation, real-time analysis of library statistics, and sending data for base calling. In addition, MinKNOW can grant access to the real-time data stream through a gRPC endpoint. This allows for third-party tools to be used to analyse the data stream for in-progress molecules and provide feedback on whether to keep sequencing or eject each molecule.

MinKNOW is configured using a sequencing protocol. These protocols control the hardware settings such as sequencing temperature or voltage; in addition they also control real-time detection settings such as when molecules should be unblocked or how the voltage should be adjusted. These protocols also expose a feature called "playback" which uses a bulk FAST5 file (Chapter 3), to replay a previous run.

Guppy

Guppy is a base caller provided by ONT that can be used via it's command-line interface, through MinKNOW, or as a server with arbitrary clients. It utilises custom

Recurrent Neural Network algorithms, developed by ONT, to interpret the signal data from the nanopore, and base call the molecule passing through the pore.

Guppy offers three different base calling models: a Fast model, a High accuracy (HAC) model, and Super accurate (sup) model. The Fast model is designed to process ~160 kb/s (when sequencing at ~400 bases/s) when sequencing with most nanopore devices (MinION Mk1C, GridION, PromethION). The HAC model provides a higher raw read accuracy than the Fast model and is currently 5–8 times more computationally-intensive. The Super accurate model has an even higher raw read accuracy, and is ~3 times more intensive than the HAC model. All three models are trained on the same datasets, with the primary difference being the detailed architecture of the recurrent neural networks.

Guppy is highly optimised for running on NVIDIA Graphical Processing Units (GPUs) using CUDA. It is generally several orders of magnitude faster running on a GPU compared to a CPU. Guppy implements stable features from development and demonstrator software that ONT produces.

Throughout this work Guppy versions 3.4.5–5.0.11 have been used.

2.1.6 Flow cell washing

Washing a flow cell removes the previous library allowing it to be reused immediately or later.

1. Stop or pause the sequencing experiment in MinKNOW, leaving the flow cell in it's postion.
2. Prepare 400 µL of washing solution by combining 2 µL wash mix and 398 µL wash diluent.
3. Mix well by pipetting, and place on ice. Do not vortex the tube.
4. With the SpotON port and the priming port closed, remove all fluid from the waste channel.
5. Open the priming port. Ensure that there is continuous buffer from the priming port across the sensor array.
6. Load 400 µL of the prepared washing solution into the flow cell via the priming port, avoiding the introduction of air.
7. Close the priming port and wait for 60 min.
8. Ensure that the priming port cover and SpotON sample port cover are both closed.
9. Using a P1000 pipette, remove all fluid from the waste channel through the waste port.

The flow cell is now ready to either be stored for later use or run a second sequencing library.

2.2 Bioinformatics

2.2.1 Curation of target regions

The software presented throughout this thesis expects either individual chromosome names or a `csv` formatted string to be provided as targets for enrichment or depletion. In addition to being specified inline, targets can be stored in an external text file as a list or as a `csv`. If a `csv` is given the format `chromosome,target-start,target-end,target-strand` is expected. When loaded the targets file are tested to ensure that the target contigs are present in the reference being used for the experiment and that the region specified is within it's bounds.

Files in the `csv` format can be converted to a six column BED (BED6) file that preserves the strand information like so:

```
sed "s/,/\t/g; s/\t/\t.\t.\t/3" < TARGETS.txt > TARGETS.bed
```

Target sets were curated from online resources. EMBL-EBI (BioMart) was used to ascertain exon coordinates in GRCH38.p13. Using the "Human Genes" dataset, filters were applied to limit the chromosomes to those found in hg38 canonical set. This set was further refined to include genes with transcript names (and IDs) only; and finally to limit the gene type to only "protein coding". The minimal attributes required for this dataset are the chromosome name, exon region start, and exon region end. The query should be visible here: BioMart bookmark URL.

For the COSMIC panel (Forbes et al., 2010; Tate et al., 2018) the target loci were downloaded from cancer.sanger.ac.uk (COSMIC Release v90). All genes with coordinates were converted into the `csv` format required by the software.

Target regions are routinely extended to increase the likelihood of seeing on target reads. This is done through the incorporation of flanking sequence both upstream and downstream of the original coordinates. These intergenic regions are included so that reads starting close to — but outside of — the target region are also sequenced.

2.2.2 Programmes and tools used

Throughout this work many pre-existing bioinformatics programmes and tools have been used. Custom scripts and programmes, written in Python, were used for data analysis and management.

Alignment and classification tools

Minimap2 (Li, 2018) is a long read aligner that is designed for long-read sequencing data. It uses a hash table built from a reference genome's *k*-mers to find "anchors". These are short but perfect alignments. Then, by chaining anchors together determines the approximate location of a read. Dynamic programming is then used to join the gaps between anchors providing base-level alignment. In addition to Minimap2, there is a Python interface called "Mappy". This is just a CPython layer that allows Python to utilise the underlying optimised C code. When running Minimap2, default parameter values for Oxford Nanopore data were used. This is acheived by supplying the flag `-x map-ont`.

For taxonomy assignment metagenomic classifiers **Centrifuge** (Kim et al., 2016) and **Kraken2** (Wood et al., 2019) have been used. Kraken2 was used to identify species from assembled genomes (Sections 5.4.1 and 5.4.2) and Centrifuge was used to classify unassembled DNA reads against a reference database (Section 5.4.2). Typically, metagenomic classifiers use *k*mer (Kraken2) for assigning short DNA fragments (~50 bases) but this can lead to very large index databases, so others (Centrifuge) employ the Burrows–Wheeler transform to compress the database.

Data management and analysis

Samtools (Li et al., 2009; Danecek et al., 2021) is a programme for processing and analysing high-throughput sequencing data. Primarily samtools is used for file format conversion and for querying, sorting, computing statistics and quality control on aligned datasets.

Bedtools (Quinlan and Hall, 2010) is suite of tools built for handling genomic data and doing genomic analyses. While it primarily is concerned with genomic intervals and ranges, bedtools is able to parse many formats. Within this thesis bedtools has primarily been used in the management of genomic ranges in the curation of target sets.

Mosdepth (Pedersen and Quinlan, 2017) is a tool for calculating genome-wide sequencing coverage. It measures depth from BAM files and can calculate either per-base coverage or coverage in a specified region. For sections of this thesis dealing with genomic coverage, mosdepth will have been used with default parameters.

Assembly

Miniasm (Li, 2016) is an overlap-layout-consensus (OLC) assembler that identifies overlapping sequences, using an all-versus-all alignment. It is designed for use with long reads such as those from Oxford Nanopore Technologies and PacBio. Miniasm does not actually carry out a consensus step, instead just merging unambiguous regions into unitig sequences. As such, assemblies produced by miniasm have similar base quality to the input reads.

Flye and **MetaFlye** (Kolmogorov et al., 2019, 2020) are both repeat-graph based assemblers targeting single-genome and meta-genome assembly respectively. They use approximate sequence matching instead of exact *k*mer matches as with de Brujin assemblers.

Consensus generation and base modification

Nanopolish (Loman et al., 2015; Simpson et al., 2017) uses the raw (electric current) signal from nanopore-based sequencing and a hidden Markov model to evaluate draft genome assemblies. This is accomplished by calculating the probability that an arbitrary nucleotide sequence can be derived from the raw signal that was observed. This consensus generation process is iterated with the improved assembly being fed back into nanopolish (usually 50 times). In addition to improving consensus sequence, nanopolish is also able to detect base modifications using an expanded HMM and nucleotide alphabet.

Similarly, **Medaka** (ONT, 2021a) is used for creating consensus sequences using only base called data. Using a draft assmebly generated using Flye medaka creates a pileup of reads and processes these with neural-network models.

Structural variant calling

The error rates of long reads make accurate SNP and small indel calling complex. However, structural variants (SV) can be identified where read alignments show large breaks. Long reads are beneficial to SV detection as they are more likely to cross break-points or completely span the gap.

Sniffles (Sedlazeck et al., 2018) detects indels, duplications, inversions, and translocations. Likewise, **SVIM** (Heller and Vingron, 2019) is able to detect and classify six classes of structural variation: deletions, insertions, inversions, tandem duplications, interspersed duplications, and translocations. **truvari** (https://github.com/spiralgenetics/truvari) is used for comparison of SV calls between different tools.

These tools rely on alignment accuracy, and therefore the error profile of reads and alignment tool chosen will impact results. In addition, errors in the reference genome or source of base truth will cause false positives.

Other tools and libraries

Throughout this project dependency management has primarily been carried out using **conda** conda.io. Conda allows for easy generation of isolated environments that can be replicated on separate computers.

Data analysis and visualisation has been carried out using the Python libraries NumPy (Harris et al., 2020), Pandas (The Pandas Development Team, 2021), Matplotlib(Hunter, 2007), and seaborn (Waskom, 2021).

2.2.3 Published datasets used

- Nanopore human genome, (Jain et al., 2018a)
- Nanopore human transcriptome, (Workman et al., 2019)
- BulkVis bulk FAST5 file, (Payne et al., 2018)
- Comparison Zymo data, (Nicholls et al., 2019)

Chapter 3

Raw Nanopore Data

Preface

Research presented as part of this chapter has been published as

Payne, A., Holmes, N., Rakyan, V., & Loose, M. BulkVis: a graphical viewer for Oxford nanopore bulk FAST5 files. *Bioinformatics* **35(13)**, 2193–2198 (2018). (Page 157)
and
Workman, R. E., *et al.* Nanopore native RNA sequencing of a human poly(a) transcriptome. *Nature Methods* **16(12)**, 1297–1305 (2019). (Page 163)

3.1 Introduction

As previously covered (Section 1.3) raw nanopore data, squiggles, are the direct detection of polynucleotide strands using a picoammeter. These picoammeter readings are streamed from the ASIC at frequent intervals, typically occuring at a rate of 4 kHz for every sequencing pore. This continuous data stream is the real-time data that Read Until needs to process to enable selective sequencing. The data in this stream differs from the raw output of the sequencing experiment as it contains portions of signal that are not measurements of DNA or RNA; for example: open pore current, when there is no analyte present or when there is some non-nucleotide contaminant. This whole process is managed by MinKNOW, the sequencing control software.

During a sequencing experiment, MinKNOW determines if a pore is present and available through a flow cell quality control (QC) and "mux selection" (muxing) process. When the flow cell is undergoing QC/muxing each of the four wells in each of the 512 channels are tested and ranked on signal quality. Channels that are identified as being viable are used and a voltage difference is maintained across each channel. This keeps ions flowing from *cis* to *trans* and keeps drawing DNA molecules into the nanopores to be analysed. As such, every channel on the flow cell must be constantly sampled so that data can be collected about every sequencing

molecule. The flow cell is sampled in 1 s chunks, but this value can be changed prior to starting sequencing.

MinKNOW processes the real-time data stream to monitor for signals that are characteristic of DNA and RNA, or issues with a well such as membrane breakdown or a blocked pore. If a pore is "sequencing" a molecule, MinKNOW attempts to partition the signal that has been captured into discrete reads, excluding non-read-signal, in a process called segmentation. In addition to read segmentation, MinKNOW will use this real-time data stream to generate some general library statistics as well as saving the raw data to disk. All subsequent analysis assumes that each read corresponds to the complete translocation of a single molecule through a nanopore; that is, the continuous stream of data from the sequencer has been correctly segmented into individual reads. Incorrectly segmenting reads can lead to either accidentally concatenating two (or more) reads into one, creating chimeras; or over-segmenting a read into multiple reads. When live basecalling is enabled, MinKNOW incorporates extra information in the general statistics panels, showing both estimated (from the real-time stream) and basecalled metrics (from FASTQ).

3.1.1 FAST5 files

The FAST5 format is based on the Hierarchical Data Format 5 (HDF5) format, which mimics a file system containing folders (called groups) and files (called datasets). The groups and datasets can have metadata associated with them in the form of attributes that are stored as key-value pairs. As it is a highly generic format, with a mature set of libraries that facilitate working with HDF5 files on any computing platform, it is an ideal format for storing raw data and metadata.

There are two kinds of FAST5 file, a "read" file and a "bulk" file. MinKNOW writes segmented signal data into read FAST5 files, these files only contain the raw signal data and relevant metadata for the sequencing run. This metadata includes what is needed for basecalling, the offset and scaling parameters, which are used for normalization and signal conversion. Once basecalled the read FAST5 files are not essential unless using tools that utilise raw data such as nanopolish and megalodon (Simpson et al., 2017; ONT, 2021b). MinKNOW will also, optionally, write a bulk FAST5 file. These contain the entirety of the unsegmented signal data seen throughout the duration of an experiment. This file includes both the raw current traces and metadata for every sequencing channel on the flow cell. One such piece of metadata is the real-time classifications that MinKNOW made during sequencing, these are real-time decisions that MinKNOW made about DNA molecules. These are used to label what the pore can "see", for example classifications include "strand", "pore",

or "adapter" (Table 3.1) and correspond to occupied by a molecule, unoccupied, and occupied by adapter sequence respectively. These can typically be seen in a "duty time" plot, within the MinKNOW interface, that is used for gauging sequencing efficiency. Finally, as the bulk FAST5 file is a complete record of the sequencing experiment it can be used to replay experiments to simulate sequencing and test real-time processes without the need to actually use consumables.

3.1.2 Aims

Typically the basecalled FASTQ data is the most important output from a sequencing experiment. In contrast, when considering real-time processes — like Read Until — the initial signals that are observed are the most important for classification. Due to the read segmentation that MinKNOW carries out, we noticed that there was some data loss that in the raw data outputs. Moreover, these data losses are amplified when running selective-sequencing. These losses cannot be seen nor analysed as they are not written to read FAST5 files. This is because the duration that an unblock signal is being sent to a nanopore is never seen in read FAST5 files, which we were particularly interested in observing as Read Until aims to send unblocks. Therefore, a bulk FAST5 visualiser was required to understand what is happening on the flow cell surface as this was the only way to capture discarded (segmented regions) of the raw signal stream.

3.1.3 Work contribution

The author of this thesis carried out the majority of the work presented in this chapter. Including data analysis and programming. The bulk FAST5 files used in this analysis were derived from sequencing carried out by Deep Seq at the University of Nottingham. DNA extractions and sequencing were carried out by Sunir Malla. RNA extractions and sequencing were carried out by Nadine Holmes.

3.2 Results

3.2.1 BulkVis

BulkVis is a bulk FAST5 file visualization tool and associated command line scripts. For visualization, BulkVis uses Python3 and the bokeh package and the Python HDF5 library (Bokeh Development Team, 2018; Collette, 2013). Bokeh was selected as it had a large set of features centered around interactivity that enabled quick progression from an initial concept to an application that allowed quick inspection of raw signal data from bulk FAST5 files. The command line interface was also written

Table 3.1: Classification Descriptions. There is no detailed description of the relationship between bulk FAST5 file classifications and MinKNOW labels seen in the "channel panel" and "duty time plots". This table presents our assumptions about the relationship between bulk FAST5 labels and MinKNOW classifications.

Bulk FAST5 classifications	MinKNOW Labels	Description
pore, good_single, inrange	pore	A single sequencing pore is present in the channel
strand, strand1	strand	DNA is detected in a single pore in the channel
unavailable	unavailable	A single pore which is currently blocked
multiple	multiple	More than one pore is detected in the channel
adapter	adapter	An adapter sequence is currently detected within the pore
mux_uncertain, unblocking	active feedback	The channel is being unblocked
saturated	saturated	A channel is passing too much current and has been switched off
zero	zero	No current is passing through the pore — likely no pore is present in the channel
below, user1	out of range 1	Negative current is being seen
above, user2	out of range 2	Current is flowing but it is neither pore nor strand
unclassified, unclassed	unlabelled	An unlabelled channel which has no classification assigned.
event	*Unknown*	No precise definition of event is available.
transition	*Unknown*	We believe this represents a rapid and large change in current measured.
unclassified_following_reset	*Unknown*	A state associated with mux changes.
pending_manual_reset	*Unknown*	A state associated with mux changes.
pending_mux_change	*Unknown*	A state associated with mux changes.

using Python3 and utilises the pandas and HDF5 libraries (The Pandas Development Team, 2021; Collette, 2013).

Visualisation server

BulkVis is started via a command-line interface. On startup, BulkVis scans either the current or specified input folder for bulk FAST5 files. As both read FAST5 and bulk FAST5 files utilise the same file extension bulk files are identified by their unique datasets (which are also required for visualisation). After this scan is complete the web-browser is opened and all available bulk FAST5 files are listed in a dropdown list that is presented to the user. Once a file has been selected from the list a user can begin to browse the raw signal dataset for any channel or they can choose to inspect a specific read by providing the header from a FASTQ record. If "browsing" an individual channel and time offset (in seconds), in the form `channel:start-end`, must be entered and the corresponing squiggle will be drawn. In addition to drawing the squiggle for that period overall metadata for the sequencing run will be displayed on the left (Figure 3.1).

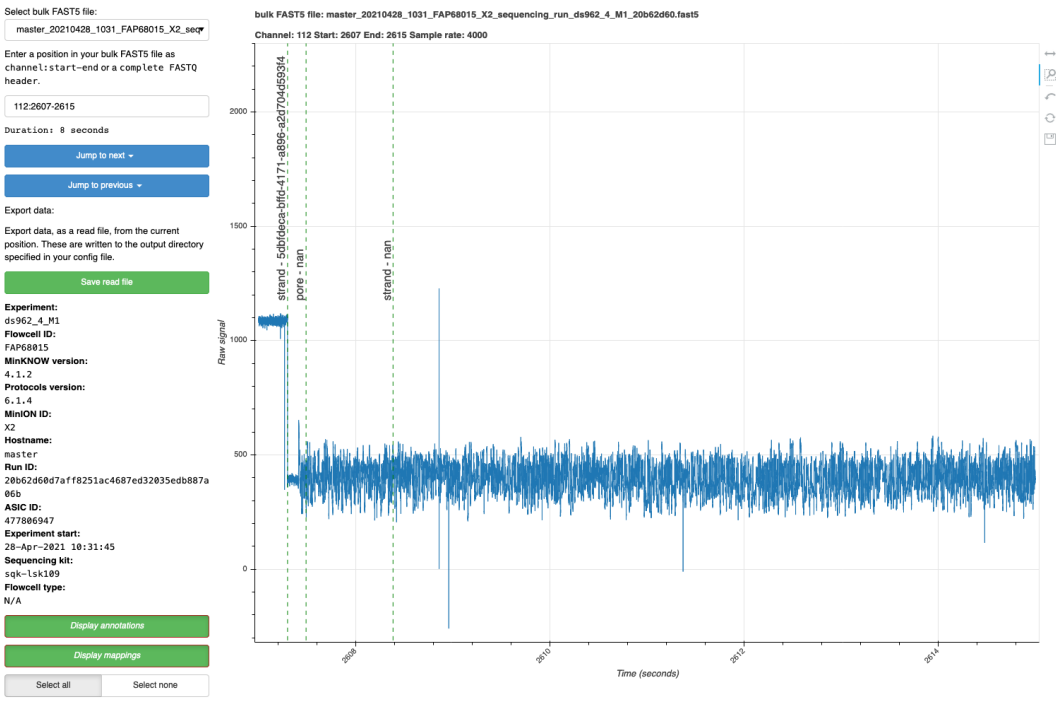


Figure 3.1: Screenshot of a typical bulkvis view. After selecting a bulk FAST5 file to view using the dropdown (top left) coordinates are required to navigate to a specific channel and then a specific time range in that channel's data stream. Once this information is provided the viewer displays the signal trace for the coordinates.

BulkVis overlays MinKNOW classification labels (Table 3.1 and Figure 3.1) at the time point that they occured over the raw signal plot. These labels correspond to more generic labels that are seen in the "duty time" plots and are used for internal classification of reads within MinKNOW. This further aids inspection as a user can quickly see when MinKNOW made these decisions and what signal the classification was based on. Optionally, a user can associate alignments, from a PAF file (Li, 2018), to annotate each read's genomic position. After an alignment has been done it is integrated with the sequencing summary file for this experiment and saved alongside the bulk FAST5 file with a file name that corresponds to the unique run identifier for this experiment[1]. When a bulk FAST5 file is opened by BulkVis the corresponding alignment-summary is also loaded if it is available. Both of these annotations can be seen in Figures 3.2a to 3.2d. MinKNOW's annotations are overlaid on the signal plot as vertical dashed lines, labelled with the type and associated ID if available (Figure 3.2). Alignments can also be overlaid horizontally above the signal, with blue and red spans indicating forwards and reverse mappings, respectively (Figure 3.2). The alignments include the chromosome, start, and stop coordinates for the read IDs that mapped. This process continues until the sever is closed, allowing any available channel to be inspected. Selecting another file from the dropdown menu will close the current file and open the new selection.

In addition to being a viewer for signal data, run metadata, and signal contextual data (classifications and alignments) BulkVis is able to export signal for basecalling, quickly navigate between classification labels and only show labels of interest. Jumping between labels is useful for quickly assessing if there was a relationship between a specific signal and classification (e.g. "transition", see Table 3.1). MinKNOW makes a lot of classifications so it is essential that they can be selectively turned on and off otherwise the signal plot could be entirely obscured. Enabling specific annotations also makes them available for navigation, allowing a user to jump to the next or previous occurence of a classification of interest, for example unblock signals. Exporting arbitrary sections of signal to a new read FAST5 file is useful for when MinKNOW has incorrectly segmented a read or truncated the read early. These new BulkVis derived FAST5 files can be basecalled by ONT basecallers, such as Albacore and Guppy. Through basecalling these incorrectly segmented reads as though it were a single molecule some extra nucleotide data is recovered from the inclusion of signal that was discarded by MinKNOW.

[1] That is: `<run id>.bmf`

BulkVis was developed in part to observe the effects of unblocking — the removal of molecules by the reversal of voltage across a specific channel. In BulkVis' development data generated in the course of sequencing the human genome on a MinION (Jain et al., 2018a), using ultra-long DNA molecules (Quick, 2018), was used. During library preparation, adapter sequences are added to DNA molecules such that every sequenced read should begin with an adapter sequence. Using BulkVis we observed reads that did not follow the expected "pore", "adapter", "strand" sequence (Figure 3.2a). We found "strand" sequences separated by either "above" or "transition" (Figure 3.2c) or even "unblock" (Figure 3.2d) signals without any evidence of "pore" or "adapter" sequences present. Every sequenced read should begin with a pore and adapter state, reads that do not can be described as having "unusual split events".

Close examination of reads before and after these unusual read split events, looking at read mappings just prior and post the events shown in Figures 3.2c and 3.2d, showed the two sequences were derived from adjacent positions on the same chromosome (Table 3.2). These reads, sequenced one after another, were most likely derived from single molecules. The alternative explanation is the chance sequencing of two independent molecules that map adjacently on the human reference, one after another, through the same pore.

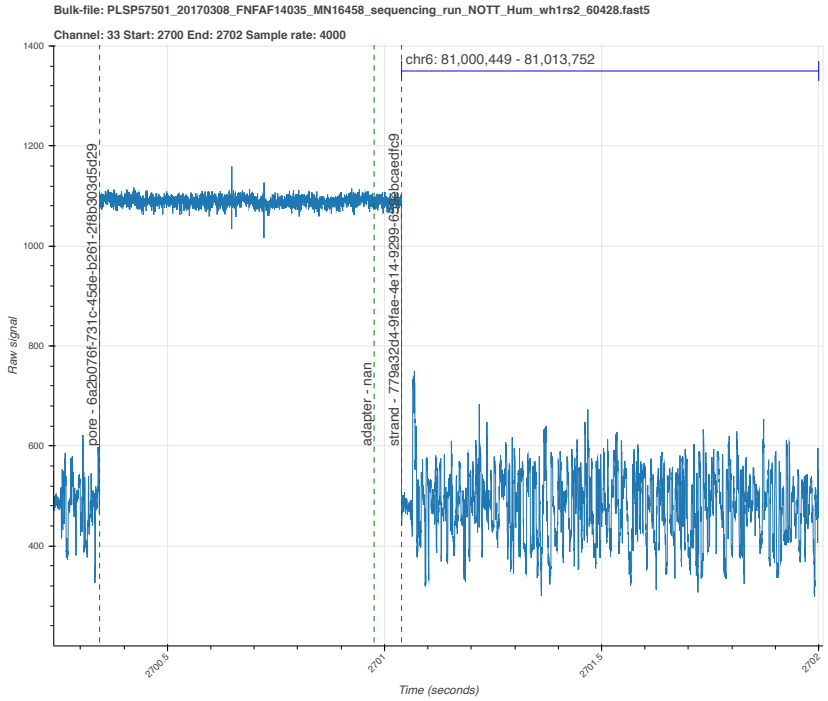
Table 3.2: Alignments for Figures 3.2c and 3.2d. The "Read ID" has been truncated for clarity. These reads are separated by either unusual current (Figure 3.2c) or by an unblock signal (Figure 3.2d). When these reads were aligned to GRCh38, using minimap2 (Li, 2018), they aligned contiguously.

	Read ID	Channel	Length	Chr	Start	End
Figure 3.2c	7ed4aafb...	176	10,275	5	122,184,560	122,199,454
	83d0cea6...		43,145		122,133,985	122,184,329
Figure 3.2d	c13c1e73...	68	5068	19	55,435,454	55,439,579
	50117d5d...		25,596		55,409,626	55,433,153

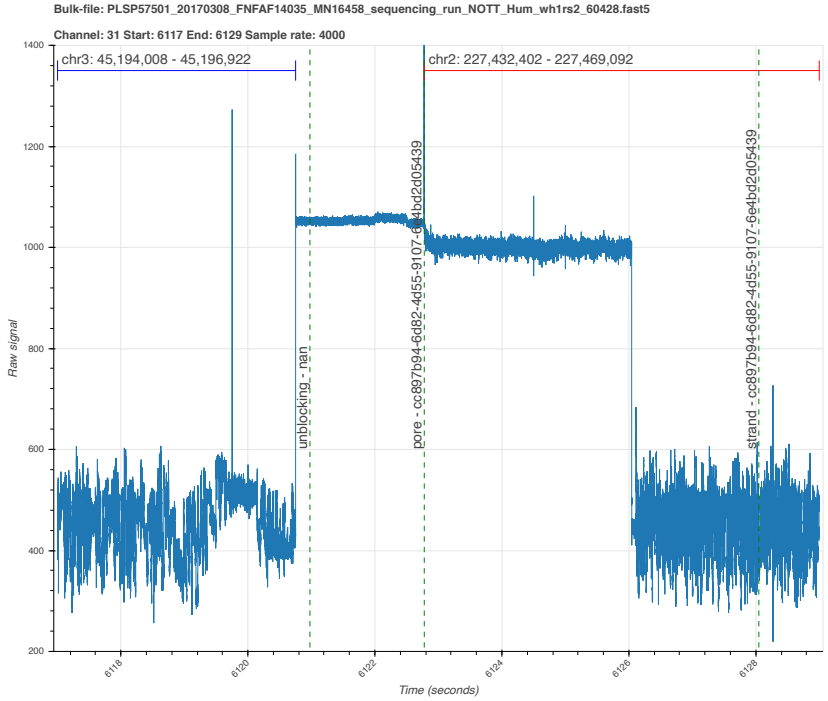
Command line scripts

The scripts described in this section are available: https://github.com/LooseLab/bulkvis.

While it is possible to determine whether a pair of consecutive reads are incorrectly segmented by eye, this process is cumbersome and time consuming. It is instead possible to use the data found in consecutive reads alignments to determine

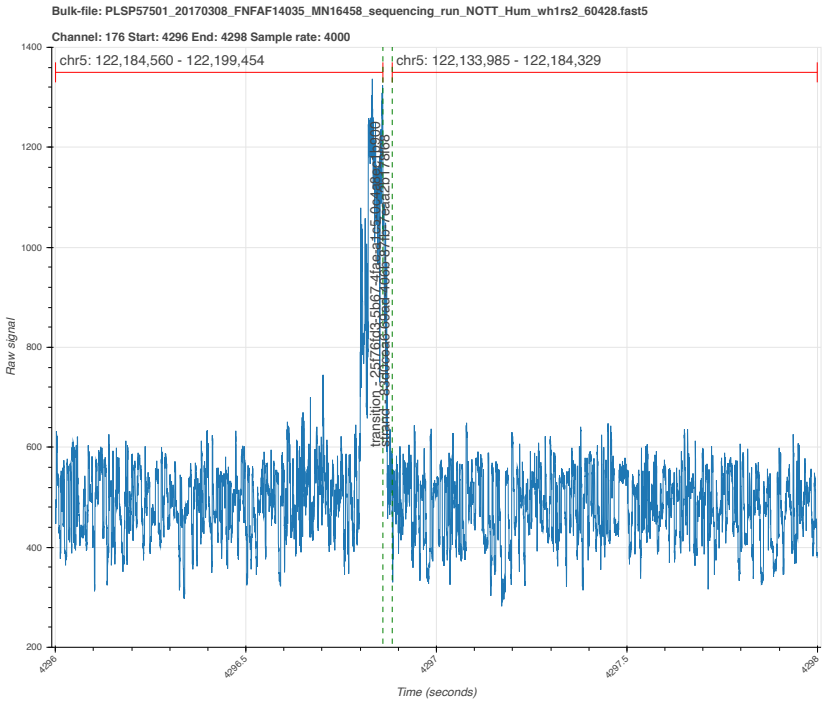


(a)

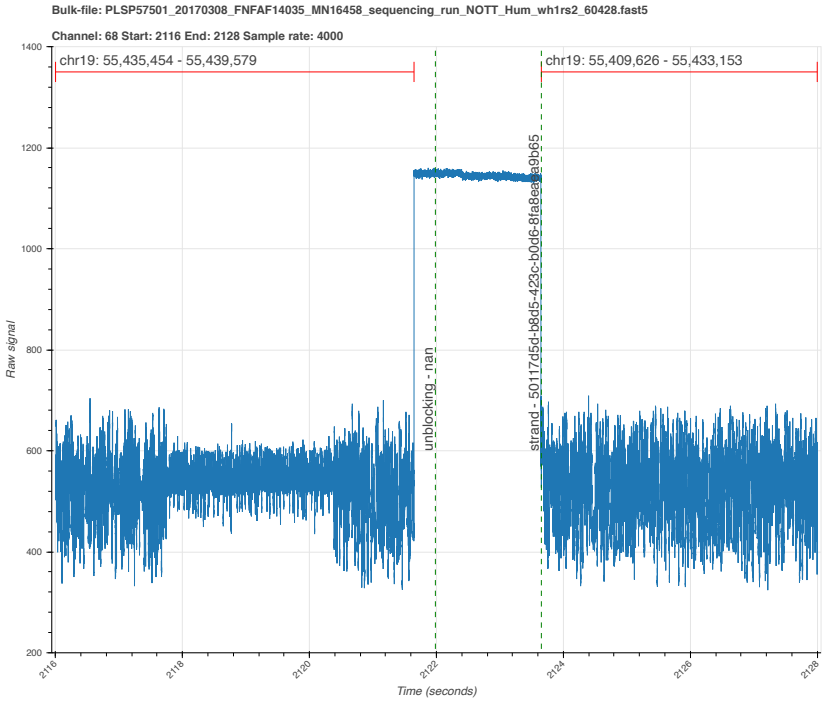


(b)

Figure 3.2: Continued of the following page.



(c)



(d)

Figure 3.2: Raw signal features. (a) The start of a read mapping to chromosome 6. Open channel "pore", followed by an "adapter", and "strand" as annotated by MinKNOW. (b) Read ending with an "unblock" followed by "pore" and then a new read. (c) Adjacent reads from a channel separated by unusual current patterns. These two reads are reported as distinct molecules by MinKNOW, they map consecutively to the reference. (d) Two adjacent reads separated by an "unblock" signal. The unblock does not successfully remove the DNA. Instead the read continues to sequence again mapping adjacently to the reference. From Payne et al. (2018).

whether they are incorrectly split or not. Using the order that each read translocated through a channel and their alignment we could establish if reads mapped to contiguous positions of their reference genome. This is carried out using the analysis script "whale_watch.py"[2] or command `bulkvis fuse`. Using a single flow cell of data from Jain et al. (2018a) we aligned the reads to the human reference genome (Schneider et al., 2017); 2983 of 75,689 total reads were incorrectly split with pairs of reads mapping adjacently to the reference. Concatenating the basecalled reads together (using "whale_merge.py" or command `bulkvis merge`) increased the read length N50 from 98,876 to 103,925 bases. In addition the mean read length of incorrectly split reads, 55,190 bases, is higher than the entire dataset, 23,717 bases. Re-examining previous ultra-long datasets revealed incorrect read splitting occurred 1–10% of the time (Table 3.3). Incorrectly split reads had consistently higher mean read lengths than those which appear to be true single molecules.

Table 3.3: Read length statistics for 14 runs from Jain et al. (2018a) with incorrectly split reads calculated using whale_watch.py after alignment to GRCh38.

Read count			Mean			N50		
Original	Split	%	Original	Split	Corrected	Original	Corrected	Increase
82,136	3953	4.81	22,532	64,810	23,134	126,793	138,627	11,834
53,720	1539	2.86	24,431	41,913	24,804	84,015	85,947	1932
41,384	932	2.25	20,299	51,910	20,534	59,500	61,168	1668
19,673	908	4.62	31,962	37,958	32,738	132,277	135,990	3713
73,752	2489	3.37	28,268	56,948	28,777	129,792	135,156	5364
75,689	2982	3.94	24,957	55,190	25,482	98,876	103,925	5049
61,223	2769	4.52	26,129	59,149	26,776	114,934	123,304	8370
65,138	4193	6.44	26,340	49,005	27,271	102,785	111,444	8659
270,189	12,045	4.46	10,680	14,967	10,936	26,744	27,759	1015
9663	882	9.13	35,380	63,434	37,242	110,455	125,144	14,689
72,931	6860	9.41	21,243	55,293	22,410	102,621	123,768	21,147
68,167	1209	1.77	26,477	71,002	26,722	132,550	136,916	4366
71,150	2687	3.78	25,611	54,145	26,152	129,656	137,644	7988
451,019	2697	0.60	8475	10,554	8501	14,957	15,016	59

Analysing the annotation states from a bulk FAST5 file showed that the some classifications occur alongside the start and end of incorrectly split reads (Figure 3.3). The most frequent classifications occuring at the start and end of split reads are "above" and "transition"; both "unblocking" and "unclassified" also occur occasionally, but not as frequently. The "transition" classification can be seen in Fig-

[2]Colloquially referring to the "whale scale" https://nanoporetech.com/about-us/news/blog-kilobases-whales-short-history-ultra-long-reads-and-high-throughput-genome

ure 3.2c. Analysis of the physically adjacent channels on the flow cell did not show any indication of these signals "above" or "transition" signals co-occuring.

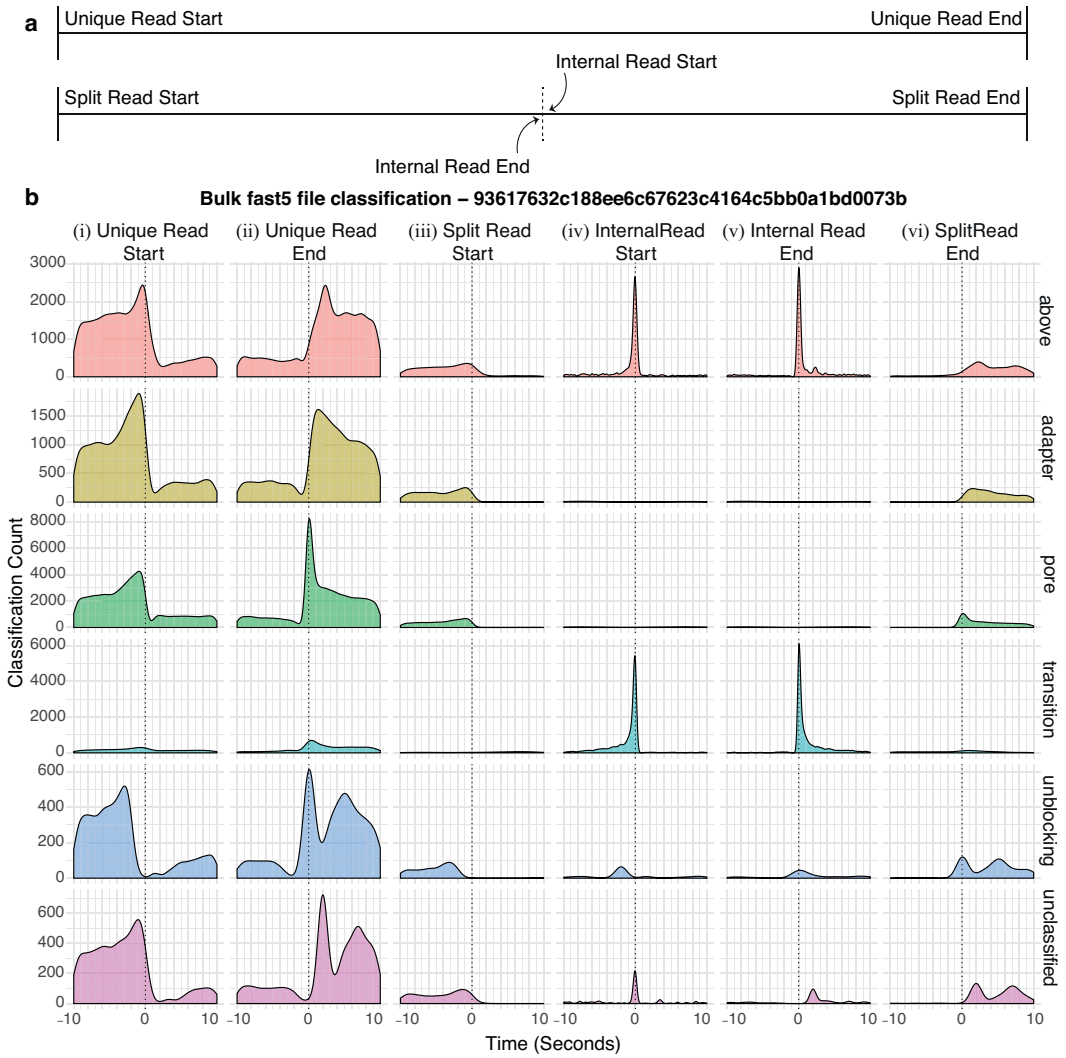


Figure 3.3: (a) Shows the labels used for reads. Unique Read Starts and Split Read Starts are genuine new molecules being sequenced. Unique Read Ends and Split Read Ends are the real end of a read. Internal Read End and Start refers to just those incorrectly split reads. (b) Shows the density of each selected MinKNOW classifications (Table 3.1) in a 10 second window before and after each of these read labels. The classifications "above" and "transition" mainly occur at split read starts and ends.

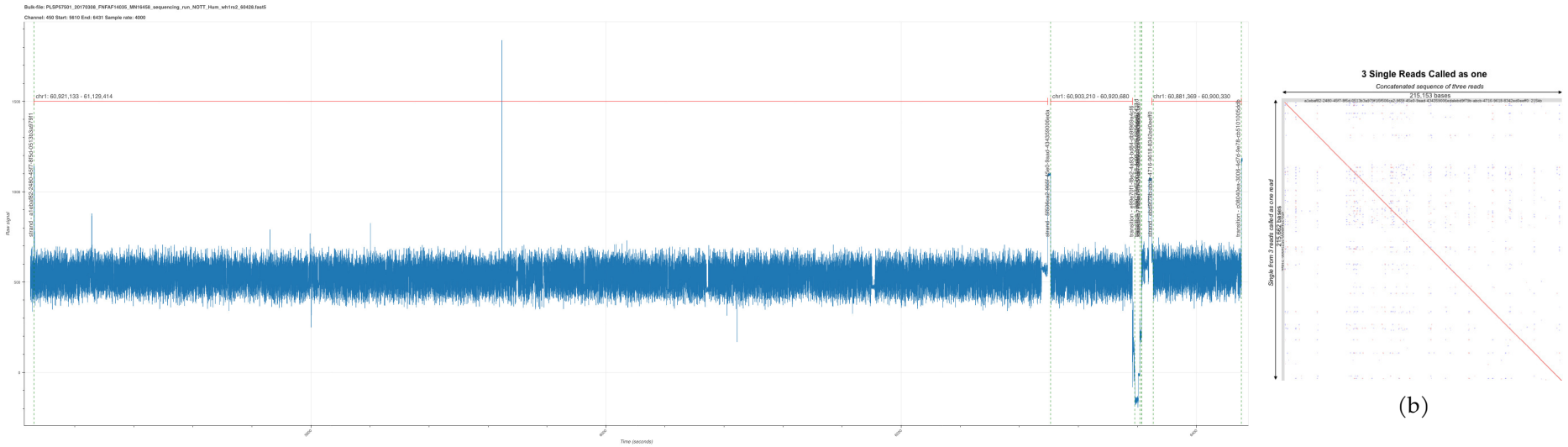
Reads that are incorrectly partitioned by MinKNOW can be rejoined, either by concatenation of the basecalled FASTQ or by generating a read that encompasses all the sub-reads using BulkVis. In the case of creating new FAST5 for basecalling, the region captured by three single reads (Figure 3.4a) has a combined length of 215,153 bases; when basecalled again as a single read has a length of 215,662 bases that aligns contiguously with the original three (Figure 3.4b).

In additional experiments when sequencing for ultra-long reads as in Jain et al. (2018a) and Quick (2018), BulkVis was able to detect incorrectly split reads in up to 30 % of reads in one run. The differences between these runs include the input library, the sequencing kit (RAD004 rather than LSK108), and other equipment such as the flow cell (both r9.4) and MinKNOW (version 1.10.23). In this experiment a single read with a length of 1,204,840 bases was sequenced, when analysing this dataset for split reads a set of eleven reads were discovered that, when merged together, spanned 2,272,580 bases and aligned to 2,290,436 bases of the human genome. Unfortunately, this part of the sequencing experiment was not captured in a bulk FAST5 file. The longest incorrectly split read that was present in a bulk FAST5 file was 1,385,925 bases in length. It was derived from nine individual reads (Figures 3.5a and 3.5b). In this instance, BulkVis could be used to generate a single read from these nine that, when basecalled, results in a single read aligning entirely to a single genomic locus. Highlighting the value of the data discarded during segmentation.

Investigating further revealed changes in normal current flow that cause real-time MinKNOW read detection to split the read. These events sometimes trigger an unblock signal to be sent to the channel, after which the read should be ejected. However, reads can occasionally continue to sequence from the same point on the molecule. In one instance a read failed to unblock for more than 46 min, as the molecule occupying that pore appeared to be stuck (Figure 3.6). In this example the pore could not sample further molecules until the blockage was cleared preventing >1.2 Mb of data being generated. Furthermore, this molecule was not rejected from the pore by the unblock, instead it continued sequencing.

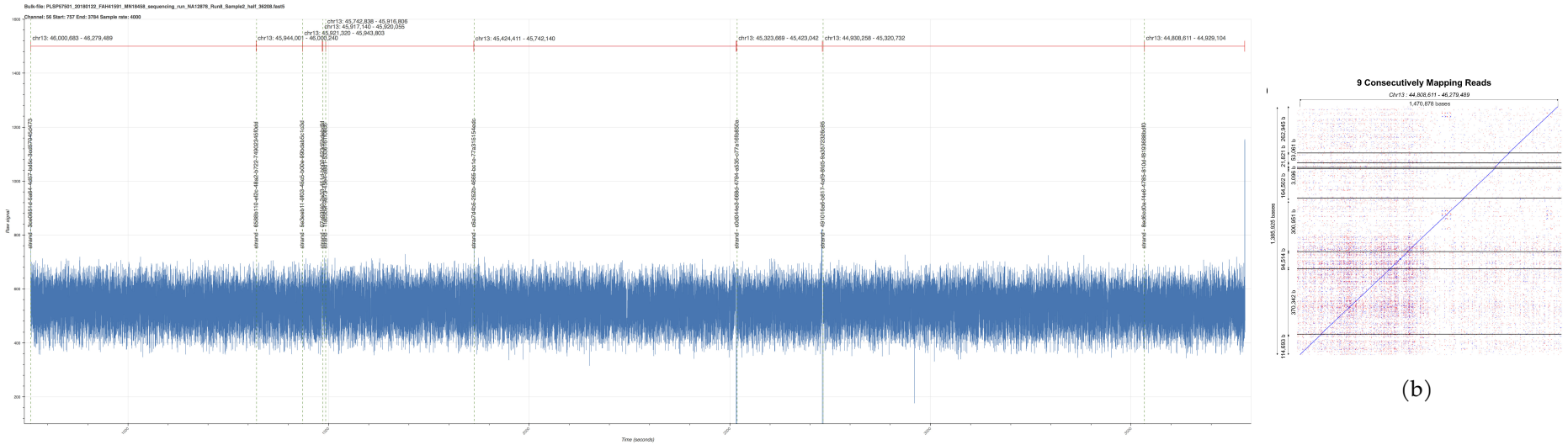
The most complex fused read observed consists of 38 individual reads mapping contiguously to the genome (Figures 3.7a and 3.7b).

Thus far, just looking at raw signal data from DNA has shown that extra, useful, contextual information can be recovered. Through visualisation, sequences where MinKNOW cannot make optimal decisions can be observed.



(a) (b)

Figure 3.4: (a) BulkVis plot of three reads as determined by MinKNOW. These reads are separated by strand classifications, but not pore or adapter. Using BulkVis a new read FAST5 file was generated, for basecalling, that encompasses all three reads. (b) Last alignment and dot plot of the three individual base-called reads aligned against the merged signal for the same three reads but basecalled as one read by BulkVis. A zoomed in view can be seen in Figure A.1 (Page 143)



(a)

(b)

Figure 3.5: (a) BulkVis plot of nine reads as determined by MinKNOW. These reads are separated by strand classifications, but not pore or adapter. Using BulkVis a new read FAST5 file was generated, for basecalling, that encompasses all nine reads. (b) Alignment of the 1,385,925 bases from the merged signal results into a single contiguous alignment to chromosome 13 on the human reference spanning 1,470,878 bases.

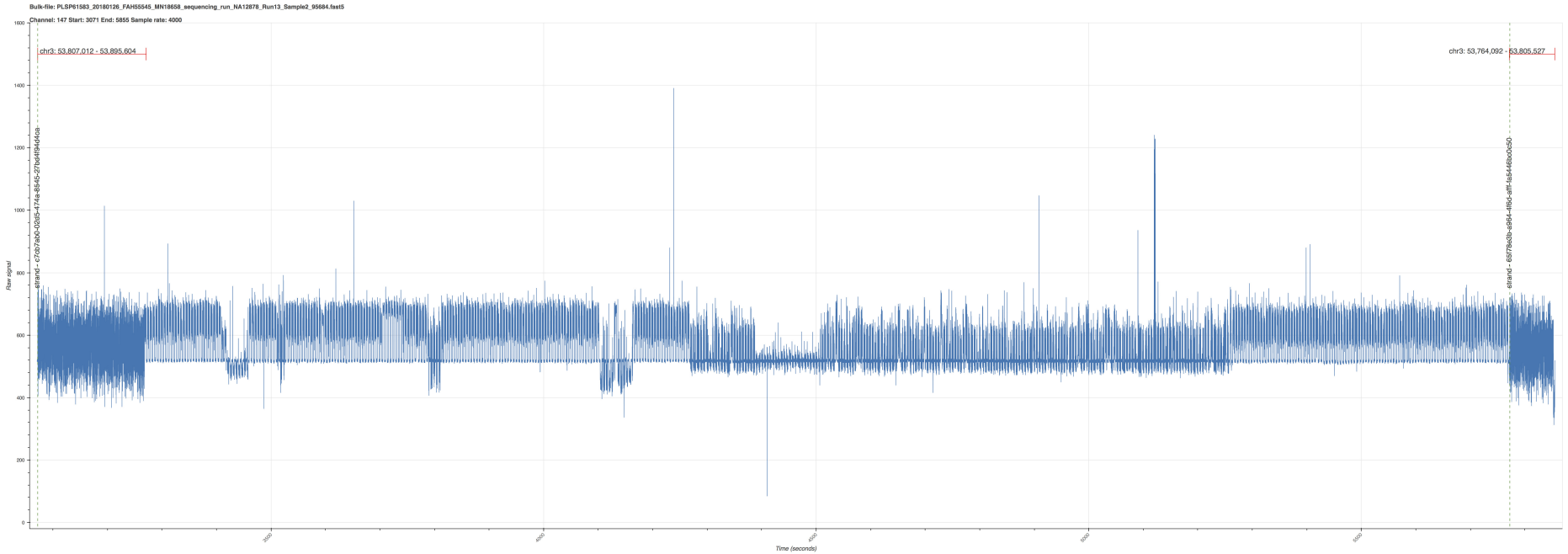
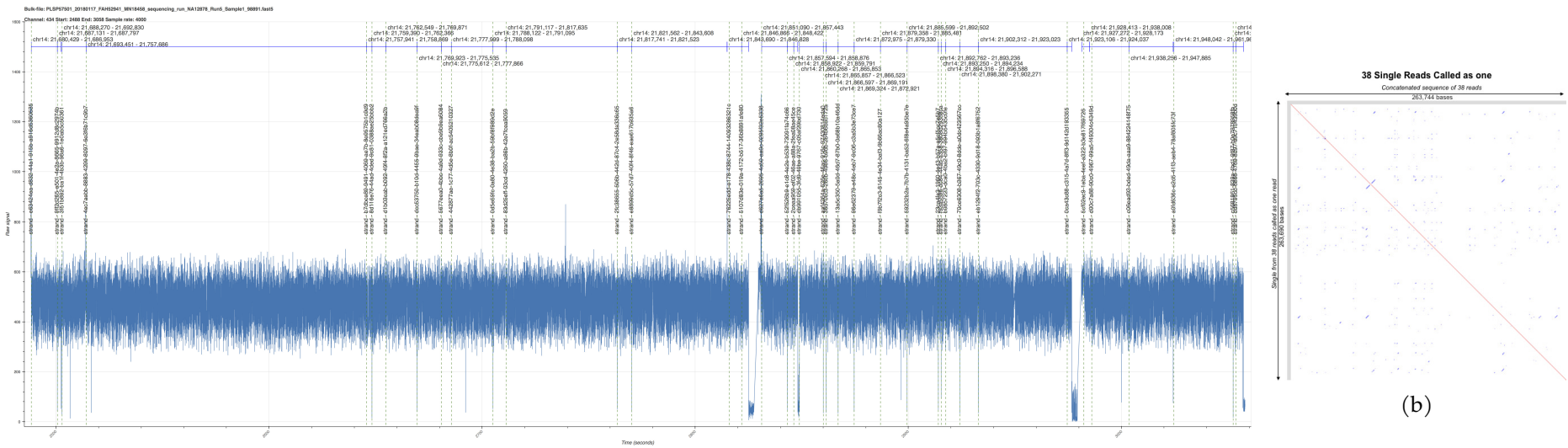


Figure 3.6: A read that could not be unblocked. Unblock signals were sent to this channel for over 46 minutes, but the next sequenced molecule aligns in the same genomic location. Note that the second read does not appear to have an adapter. Dashed lines indicate the start of new molecules as determined in real-time by MinKNOW.



(a) (b)

Figure 3.7: BulkVis full length signal plot for a region spanning 38 individual reads from a bulk FAST5 file. Dashed lines indicate new reads as identified by MinKNOW. When generating a new read from this entire sequence it base calls as a read with length 263,744 bases.

RNA

Nanopore RNA sequencing shares the same features as nanopore DNA sequencing. That is, a motor protein (M1) drives a polynucleotide strand of RNA through a protein pore at a steady rate (70 b/s). As this method of sensing is the same the effects seen in DNA molecules are likely not just limited to DNA molecules.

With poly(A) RNA sequencing full-length[3] transcripts are expected. However, some mitochondrial transcripts showed a random distribution of truncated reads below their expected full length (Figure 3.8b). Quantifying the fraction of truncated reads by their expected transcript length for ten mitochondrial mRNAs, we found a strong negative correlation (Figure 3.8c). This can also be seen in the number of full-length transcripts over each mitochondrial gene as there are more partial transcripts creating a saw-tooth coverage distribution on the heavy strand (Figure 3.8a).

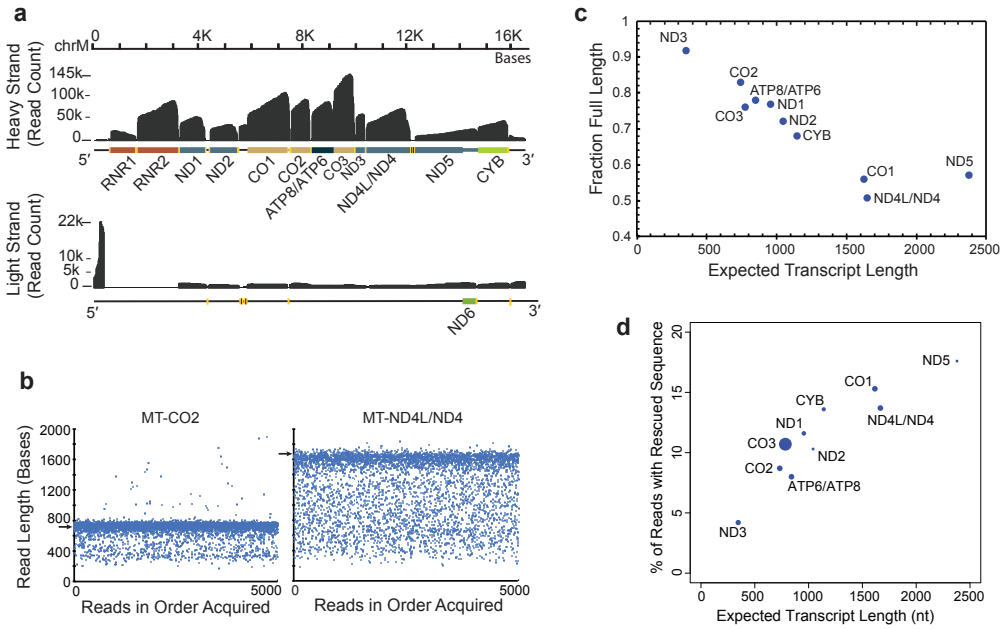


Figure 3.8: (a) Read coverage of the heavy strand (top) and the light strand (bottom). (b) Distribution of nanopore read lengths for MT-CO2 and MT-ND4L + MT-ND4 transcripts. Each point represents 1 of ~5000 reads in the order acquired from a single MinION experiment. Horizontal arrows are expected transcript read lengths. (c) Relationship between expected transcript read length and fraction of nanopore poly(A) RNA reads that were full length. (d) Percent of artificially truncated reads where sequence was recovered from the ionic current signal. Dot sizes indicate relative number of reads. Adapted from Workman et al. (2019).

Analysis of bulk FAST5 files derived from these RNA sequencing experiments,

[3]Extending to within at least 25 nt of the genes expected 5′ terminus

revealed that MinKNOW sometimes removes too much signal when segmenting reads into discrete molecules (Workman et al., 2019). Using 2729 mitochondrial RNA reads aligning to mitochondrially encoded cytochrome c oxidase I (MT-CO1), a systematic analysis identified 527 reads that started or ended abnormally. By using the methods developed in BulkVis and including ionic current segments that were identified before or after many of these truncations, ~300 reads were reconstructed with longer alignments to MT-CO1 (Figure 3.9). These truncation events are length dependent (Figure 3.8d), ranging from 4.2 % of reads with rescued segments for ND3 (full length 346 nt) to 17.6 % for ND5 (full length 2379 nt).

Visual analysis[4], through read overlapping, indicated that read truncations were more often caused by electronic signal noise such as current spikes of unknown origin (Figures 3.9a to 3.9c). However, despite these current spikes meaningful signal can be recovered from the raw signal data in the bulk FAST5 file. We showed that meaningful biological signals can be recovered from bulk FAST5 files around these truncations, suggesting that future improvements to the MinKNOW read segmentation pipeline are needed.

[4] https://github.com/nanopore-wgs-consortium/NA12878/tree/master/nanopore-human-transcriptome/scripts/bulk_signal_read_correction

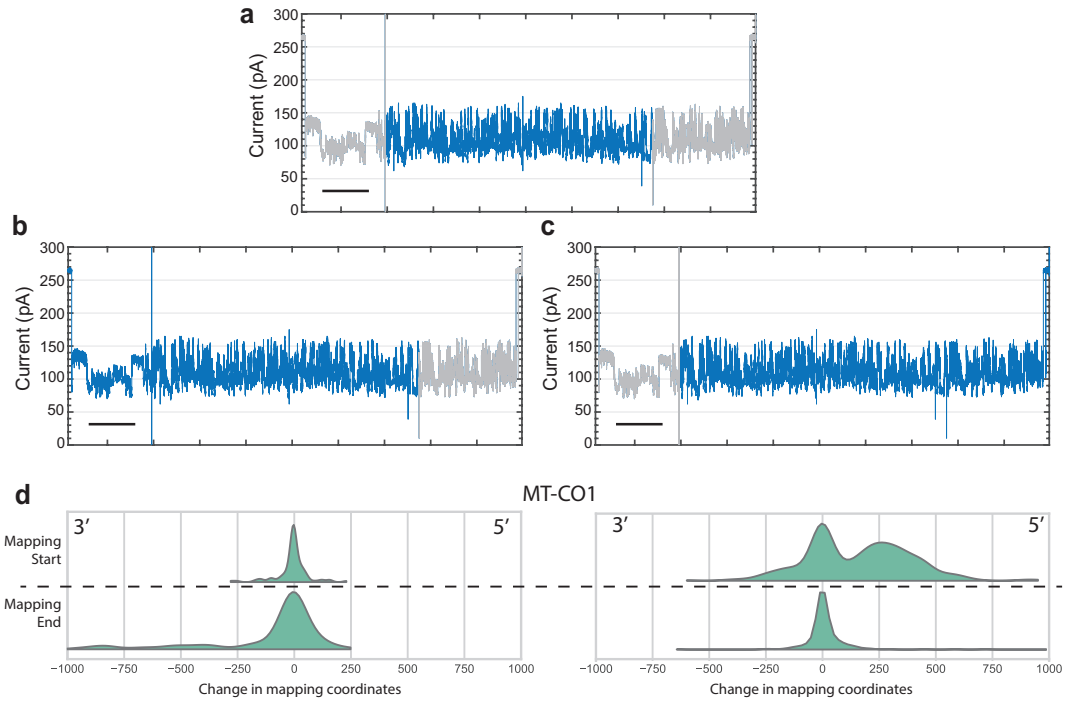


Figure 3.9: (a) Example ionic current signal for a MT-CO1 transcript. This trace is representative of reads that were artificially truncated by a signal anomaly. The highlighted blue section represents the MinKNOW segmented read (positions 474–1532 of the MT-CO1 gene), and the blue and right grey sections represent the manually segmented and rescued read (positions 27–1532 of the MT-CO1 gene). The signal in grey was not present in the MinKNOW output read FAST5 file, but could be extracted from the continuous FAST5 file using BulkVis. (b) Recovery of data at the 3′ end of a read (shaded) using BulkVis. (c) Recovery of data at the 5′ end of a read (shaded) using BulkVis. (d,e) Effect of additional ionic current data on the mapping coordinates (start and end positions for an alignment) relative to the reference transcript for all MT-CO1 reads in bulk FAST5 files. Increasing the amount of decodable nucleotide data enables for better, longer alignments that better place the reads in their genomic context.

3.3 Discussion

Nanopore sequencers allow the capture of continuous raw signal data, in a bulk FAST5 file. This includes continuous ionic current for all channels on a flow cell sampled at 4 kHz for DNA and 3 kHz for RNA; and the real-time classifications made by MinKNOW during the original sequencing run. This crucially differs from the more widely known read FAST5 file, that only contains raw data and some limited metadata.

There is no need for the routine collection of bulk FAST5 files. However, as extra data can be rescued resulting in longer, more contiguously aligning, reads; the methods of generating "fused" reads (by either concatenation or re-basecalling) are likely of interest for *de novo* genome assembly. Especially of non-model organisms, which may not sequence or basecall as well as standard models, such as human or bacterial samples.

Analysing a bulk FAST5 file and the outputs of a sequencing experiment is relatively easy using the provided command line interface; which is a useful post-sequencing check for users working with a well curated reference genome. Despite these limitations, bulk FAST5 file analysis has show evidence of of incorrect read segmentation across all Oxford Nanopore platforms (MinION, GridION, and PromethION) and both of the current analytes (DNA and RNA).

In response to some of these findings ONT have refined MinKNOW's ability to detect and avoid incorrect segmentation (introduced in MinKNOW 2.0 between May-Oct 2018). One such mechanism is the use of a "progressive unblock", which replaces the original unblock (2 s of reverse current) with a more gentle (starting at ~0.1 s) reversal duration and only intensifying if needed. Moreover, ONT have also introduced molecular methods such as the nuclease-flush[5] and reload to physically clear blocked channels (Sept 2019).

BulkVis provides a tool for the visual inspection of raw signal data with the goal of understanding what is being seen and discarded during a sequencing experiment. Crucially, it provides the opportunity to re-interpret the signals and measurements that were recorded during sequencing. Unfortunately these measurements are not entirely free from *some* interpretation as MinKNOW is still managing the sequencing. But, greater information can be acquired by inspecting signal where MinKNOW's classification deviates from the expected cycle: "pore", "adapter", strand". This could be of particular use for challenging sequences and samples to discern if there are unexpected artefacts as in (Parker et al., 2020).

[5] https://store.nanoporetech.com/uk/flow-cell-wash-kit-r9.html

There is a lot of information held in the signal data. It is primarily used for base-calling, as this is a nucleotide sequencing platform. But some tools are able to make use of the signal data to provide enhanced biological information in the form of modified bases (Simpson et al., 2017; Müller et al., 2019; Boemo, 2021), or by rescuing truncated signal to extend reads (Payne et al., 2018), or through alignment of raw signal data (Kovaka et al., 2020; Zhang et al., 2021).

Chapter 4

Readfish development

Preface

Research presented as part of this chapter has been published as

Payne, A., Holmes, N., Clarke, T., Munro, R., Debebe, B.J., & Loose, M. Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nat Biotechnol* **39**, 442–450 (2021). (Page 175)

4.1 Introduction

4.1.1 Nanopore sequencing

As covered in Section 1.3 the technology at the heart of Oxford Nanopore Technologies' platform consists of a protein nanopore embedded in a synthetic membrane. Molecules of DNA or RNA are actively driven through the channels that these nanopores create, travelling from the *cis* to *trans* side of the membrane, by a combination of electrophoretic force and from a motor protein that mechanically "walks" the nucleotide strand to control the rate of translocation (Branton and Deamer, 2019).

During the course of a normal sequencing experiment a library of molecules is prepared and loaded on to a flow cell. As the molecules pass through the nanopores the current difference across the membrane is recorded at regular intervals. This process is continuous for the duration of the sequencing run. In the event that a molecule blocks the channel or cannot continue sequencing the applied voltage across the membrane can be inverted to reverse the direction the molecule is travelling; sending it back out the way it came in. Here I will define some terms to differentiate between unblocking reads during the normal course of sequencing and specifically for selective sequencing applications.

Unblocking reads is a mechanism, used initially to prevent pores from blocking, can also be used to stop sequencing a molecule at any point. This process is called "Read Until", which allows a single molecule to be sequenced until the voltage is reversed to remove the strand from the nanopore. Not only can the change in voltage can be activated by MinKNOW, but the user of the sequencing device can be given control to unblock individual reads on a pore-by-pore basis.

Selective sequencing uses pre-defined, immutable, conditions to make real-time decisions about currently sequencing molecules. MinKNOW's "active unblock" is an example of selective sequencing as it aims to detect nanopores that have become blocked and clear them by reversing the voltage. Likewise, selecting specific genomic regions (as demonstrated in Loose et al. (2016)) is selective-sequencing.

Adaptive sampling is the process by which the experimental conditions are updated as sequencing progresses — directly in response to the data generated by the sequencer. Thus far true adaptive sampling has only been shown by Loose et al. where specific viral amplicons were sequenced until they had reached sufficient coverage for a consensus sequence to be generated (Loose et al., 2016).

For selective sequencing to work molecules must be analysed in real time. As a strand is progressing through a nanopore the current is streamed from the sequencer to the controlling computer. Through inspecting these live current traces the molecule present in a channel can be classified and a decision can be made about whether to continue collecting data, allowing the read to end naturally, or whether to eject the molecule and sample another molecule from the available pool.

4.1.2 Current selective sequencing implementations

Selective sequencing is dependent on the ability to match molecules currently progressing through a nanopore with a reference sequence. This requires either converting the live signal data from the nanopore to nucleotides and aligning to a biological reference or converting the biological reference into a signal-like representation.

Signal based methods

Matching un-basecalled signal with a simulated reference is the most common method used to implement selective sequencing (Loose et al., 2016; Masutani and Morishita, 2018; Kovaka et al., 2020). First demonstrated by Loose et al. in 2016 using a modified audio sampling algorithm called Dynamic Time Warping (DTW) (Kruskal, 1983). This process matches the raw electrical signal from the DNA in a nanopore

to a simulated reference signal. The process of simulating signal from a reference sequence is accomplished by pre-calculating the mean current value for every 5 base window (5-mer) in the reference genome. Each simulated reference squiggle is unique to the *k*mer model that it was generated from as each model is dependent on the sequencing chemistry (nanopore version) that it was created with. In addition to matching raw current this DTW approach needed to compensate for noise introduced by the sequencing environment, thus incoming read fragments were *z*-score normalized to overcome these variations between the original *k*mer model used to simulate the reference and the *k*mer values reported by nanopores during sensing. Loose et al. were able to enrich for specific genomic regions and then prioritise alternate regions when a target coverage depth had been reached (Loose et al., 2016). Though this was limited to smaller genomes ($<$ 5 Mb in length) and required a 22-core server to process the data fast enough.

The DTW approach was refined by Masutani and Morishita who applied refined DTW algorithms such as Sparse-DTW (Al-Naymat et al., 2012) and Fast-DTW (Salvador and Chan, 2007) among others to increase the throughput of the naive DTW approach (Masutani and Morishita, 2018). Despite improved algorithms, due to the time complexity of DTW being quadratic[1] (Kruskal, 1983; Loose et al., 2016) longer assembled contigs were hard to place reads within optimally.

Finally, in techniques utilizing raw signal is, UNCALLED (Kovaka et al., 2020), which employs efficient index and seeding techniques to reduce the computational time of matching signal. Specifically, Kovaka et al. use a Ferragina-Manzini (FM) index built from approximate *k*mer — called events — generated from the reference sequence which is then queried using the most probable *k*mer from the live signal. Their event detector is based on Scrappie which uses rolling t-tests to detect the sudden changes in signal that define event boundaries. Like in Loose et al. each event is represented by the mean of the signal that it covers. These events are are also normalized so that the mean and standard deviation match the *k*mer model. After normalization, UNCALLED calculates the probability that each event matches each possible *k*mer from ONT's *k*mer model. Lastly, in the seed-mapping stage, short but perfect alignments between the most probable read and FM-index reference are sought which are then used to create longer alignments used in the selective sequencing process.

[1]Given an input of size n, it will take n^2 steps to complete the task

Nucleotide based methods

It is also possible to replace signal matching with a two-step process of basecalling and alignment. This technique was concurrently attempted by Edwards et al. in 2019. Their software, RUBRIC, used the (now obsolete) nanonet basecaller and the LAST aligner (Kiełbasa et al., 2011). RUBRIC demonstrated benefit compared to non-selective sequencing, by filtering unwanted reads, but did not provide any enrichment. This approach required considerable computational resources that were provided by an additional computer to the one controlling the sequencer.

4.1.3 Aims

A recurring theme in selective sequencing software is that it often takes more than a single computer to process the signal stream for real-time inspection. This is the case in DTW (Loose et al., 2016), UNCALLED (Kovaka et al., 2020), and RUBRIC (Edwards et al., 2019); and in the case of DTW and UNCALLED large high-performance servers were used. For this reason a primary goal was to utilise reasonable computational resources, ideally using a single computer such as a laptop or one that fits in MinKNOW's computational requirements (Table 2.1).

In addition, the reference genome constraints seen in signal based methods initially limited reference length to ~5 kb (Loose et al., 2016). As such, signal based methods are limited in both the size of the reference genome they can use and, by extension, the number of target regions they can consider.

Finally, with the advent of fast basecalling on GPU it has become easier to generate nucleotide data in real-time. With the addition of fast read alignement using minimap2 (Li, 2018) a completely real-time process should be possible. Leveraging this data would allow for reference genomes and target sets to be updated during the course of sequencing in response to data that has already been generated; which would make these runs adaptive as they change in response to the sample. An example of this kind of adaptive sampling is for both reference and targets, of any size, to be added or removed during an experiment.

Overall our goals for this Read Until software was to: work with a reference genome of any size, work with any number of genomic targets, allow the reference genome to be updated during an experiment, allow the targets to be updated during an experiment, work on a single computer.

4.1.4 Work contribution

The majority of the work in this chapter was done by the author apart from DNA library preparation and flow cell flushing and reloading which was carried out by Nadine Holmes in Deep Seq. The initial design and selection of target panels was done in collaboration with Matt Loose.

4.2 Results

4.2.1 Application Programming Interfaces

The MinION device is controlled by ONT's sequencing control software, MinKNOW. MinKNOW provides an Application Programming Interface (API) that enables real-time interactivity between the controlling computer and the sequencer (ONT, 2021c); a subset of this API has been curated as the Read Until API (ONT, 2020).

These APIs all use Google's Remote Procedural Call (gRPC; grpc.io) framework which standardises communication between applications without needing specific details of how they are connected. These operate fastest when both applications are on the same computer, but can create a seamless interface between distinct computers. An overview of how data is passed between applications can be seen in Figure 4.1.

Read Until API

Read Until requires bidirectional communication with the sequencing device, this is provided by the Read Until API (Figure 4.1). This API provides chunks of signal from every sequencing pore on the flow cell continuously for the duration of the sequencing experiment.

During the course of developing readfish the Read Until API has been continuously maturing as a result of both community contributions and ONT's development. Some of these developments are as a result of past research; for example both Loose et al. and Edwards et al. found there was a critical need for filtering incoming raw signal such that only data from that represented DNA molecules were served (Loose et al., 2016; Edwards et al., 2019). In response ONT implemented a classification filter so that only reads classified as "strand" or "adapter" were served over the Read Until API. These classifications were chosen as they represent either an in-progress molecule or the very beginning of a molecule, just before it's classification becomes "strand".

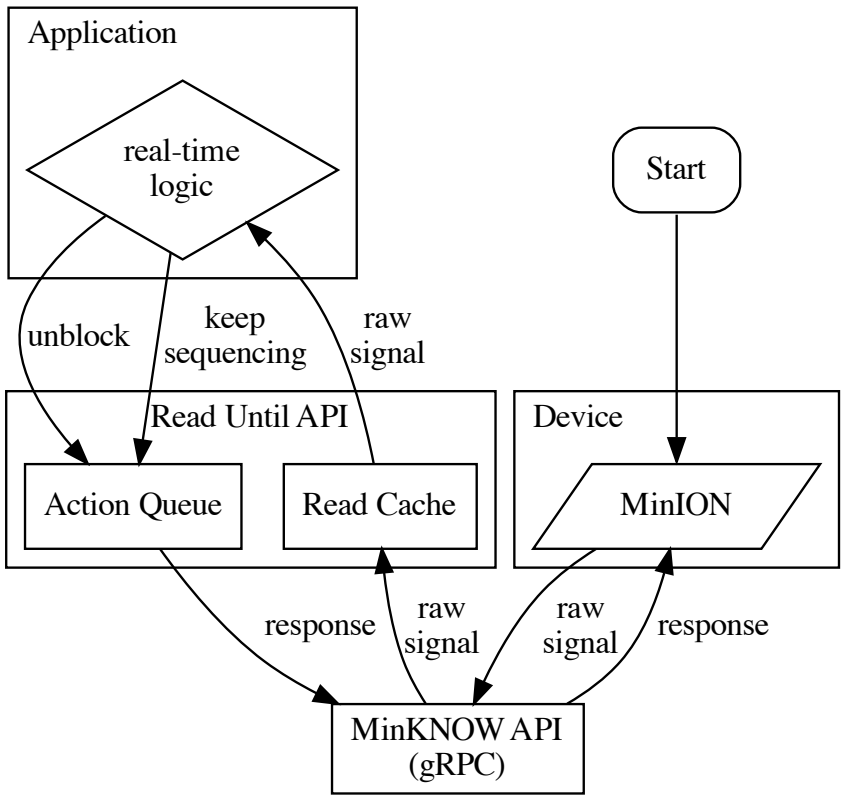


Figure 4.1: Flow diagram of how data passes between the MinION sequencer, MinKNOW and the Read Until API. When the sequencer is started the gRPC system in MinKNOW is accessed by the Read Until API and a cache of raw data (un-basecalled signal) for in-progress reads is created. This cache can be sampled by an application for the purpose of Read Until. The application can then wait for more data or make a decision to either select ("keep sequencing") or reject ("unblock") and the action is relayed back to the Read Until API and stored in the action queue, which is then communicated to the sequencer via the MinKNOW API and effected. In the case of an "unblock" decision the current is reversed and the molecule ejected; in the case of "keep sequencing" the channel will not send data to the read cache for this channel until a new read has begun sequencing.

Initially we were using a fork[2] of the Read Until API that allowed the inclusion of new features and key performance improvements. These improvements have since been incorporated into the stable 3.0 version of the Read Until API[3] (ONT, 2020). The aim of these improvements was to increase the portability of our selective

[2]https://github.com/LooseLab/read_until_api_v2
[3]Additions to the Read Until API were made during my iCASE placement at Oxford Nanopore Technologies

sequencing approach so that it was easier to install.

Most notably, we migrated the API from Python 2 to Python 3 which allowed us to take advantage of newer programming practices, improve the speed and performance of the API, and ensure that the API was not constrained by internal implementation details of MinKNOW. This migration entailed removing obsolete dependencies and enabling the gRPC MinKNOW API.

An essential component of the Read Until API is the read cache. This cache sits between the Read Until code and the MinKNOW API. It runs concurrently with the Read Until application to fetch and store new read chunks as they become available. The read cache was re-implemented to make the transfer of read chunks, from the cache to the selective sequencing code, faster by up to two orders of magnitude. There was a bottleneck in a function (`popitems`) that meant the read cache would repeatedly poll for new data even when it was empty. Each time the empty cache was polled caused an exception to occur in the Read Until API, which is a very expensive operation — especially when it was occurring many times on each attempt to get data from the cache[4]. With this step removed data could be served from the Read Until API much faster allowing better overall performance.

In it's original implementation the read cache would only supply the most recently received read chunk. Therefore, data would be missed in the event that analysing a batch of read chunks takes longer than the cache update period. To address missing chunks of data a new read cache was written[5], the accumulating cache, that does not discard consecutive chunks of data and instead combines them. This accumulating behaviour is essential when converting data from signal to nucleotides as any extra signal can aid in correctly placing a molecule in the correct genomic context (Section 3.2.1). In addition, an obscure effect of discarding missed read chunks is that it is unknowable, to the selective sequencing software, what length of nucleotides have been processed. Knowing the approximate length of a molecule that may be rejected is essential as rejecting molecules longer than ~2 kb is more likely to destroy or block the channel and reduce the overall sequencing capacity of the flow cell (Section 3.2.1).

4.2.2 Alignment

To place basecalled data in their genomic context we opted to use minimap2 (Li, 2018) over other aligners, such as LAST (Kiełbasa et al., 2011) which was used in

[4]https://docs.python.org/3/faq/design.html#how-fast-are-exceptions

[5]This read cache was originally written for readfish to use, and has since been ported into the ONT implementation of Read Until (by me, on my placement)

RUBRIC (Edwards et al., 2019). This was primarily as minimap2 is specifically optimised for use with long-read data from Oxford Nanopore and PacBio, allows for the use of pre-computed indexes that further improve alignment time, and has a mature and stable Python interface, `mappy`. Moreover, minimap2 works with both DNA and RNA reads of any length (e.g. short reads or assembly contigs). It is accurate and efficient and outperforms other alignment tools in terms of both speed and accuracy (Li, 2018).

4.2.3 Basecalling

Basecalling ONT data is the process of translating the raw electrical signal into nucleotides. It is a challenging problem as the number of possible states that a given strand of DNA could have is determined by the number of nucleotides being decoded raised to the power of the number of nucleotides that can fit in the lumen of the nanopore. Currently, for the canonical DNA bases and R9.4 pores (which use ~5 nucleotides) that yields $4^5 = 1\,024$ possible states.

Due to the exponentially increasing complexity of classifying nanopore signal data, machine learning is the primary method used for decoding signal into bases. ONT have released multiple different basecallers over the years, most notably Albacore (now deprecated) and Guppy, which use CPUs and GPUs respectively. In addition, there's Scrappie[6], Flappie[7], and Bonito[8] which are open-sourced "demonstrator" technologies that are used to refine features before their incorporation into Guppy. There are also third-party basecallers that have been developed such as Chiron (Teng et al., 2018), DeepNano, and DeepNano-Blitz(Boža et al., 2017, 2020).

Scrappie

Scrappie is an experimental basecaller, described as a "demonstrator" technology. It provides a Python interface to a basecaller programmed in C. The basecaller makes use of neural network models, which are trained to convert raw signal data into nucleotides. Scrappie's architecture is based on a gated recurrent unit (GRU), which is similar to a long short-term memory architecture, but has better performance on tasks such as speech signal modelling (Ravanelli et al., 2018); and as has been demonstrated by Loose et al. audio processing algorithms are readily applicable to the electrical signal that nanopores produce (Loose et al., 2016).

As Scrappie is a basecaller that can be called programmatically it is ideal candidate for incorporating into a programme that requires basecalling. For Read Until to

[6]https://github.com/nanoporetech/scrappie; deprecated
[7]https://github.com/nanoporetech/flappie; deprecated
[8]https://github.com/nanoporetech/bonito

be effective it needs to make *good* decisions about where reads are from in their target genome; and it needs to do this quickly. Therefore, when assessing Scrappie for use in Read Until we must gauge both the speed at which it can basecall sequences and the accuracy of the alignments those basecalls produce.

The models[9] that Scrappie provides for basecalling vary in how the parameters used in their generation. For example models `rgr_94` and `rgrgr_r94` use alternating reverse GRU (`rgr`) and GRU (`gr`) layers. Other networks, using the `k3_...` or `k5_...` naming scheme also use a reverse GRU and GRU layers, but the parameters used to generate them have been modified and are encoded in the name. These parameters are the *k*mer model, either 3 or 5; the window and stride, which alters how much contextual data a model uses when considering raw input; and finally the layers and size of the hidden data layers, which are used internally by the model.

Scrappie's models generally have good alignment accuracy, with the best models having mean accuracy of ~90%. The two best performing models had mean accuracies of 0.896 (`rgrgr_r94`) and 0.882 (`k5_w11_s5_l3_u96`) (Figure 4.2).

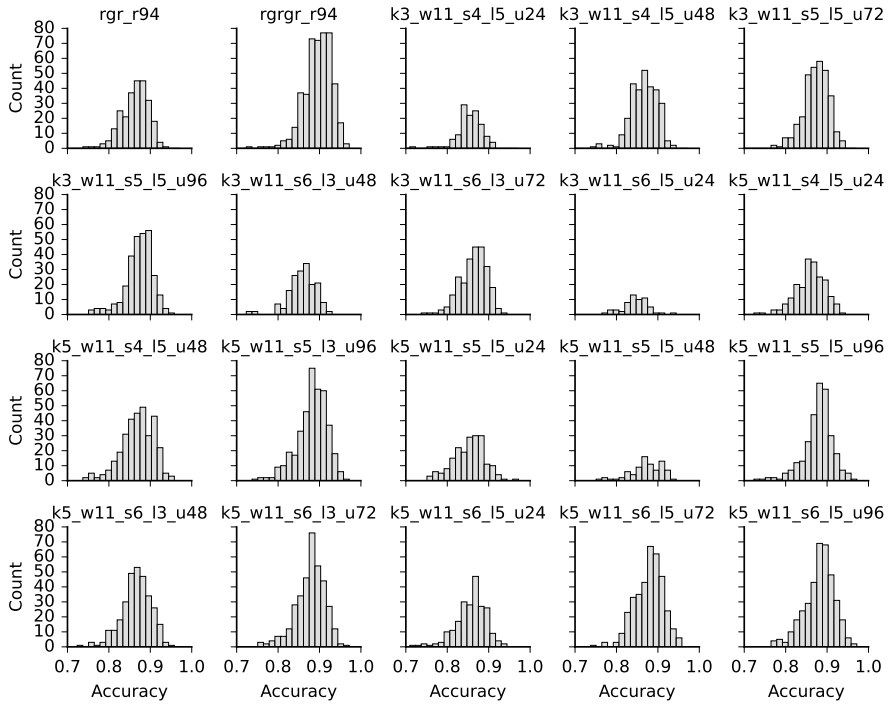


Figure 4.2: Scrappie alignment accuracy. Comparison of 750 reads basecalled with each available Scrappie model. Calculated from minimap2 alignments using matches and indels in the CIGAR string. The two best performing models were `rgrgr_r94` (row 1, col 2) and `k5_w11_s5_l3_u96` (row 3, col 2) with overall mean accuracy of 0.905 and 0.899 respectively.

[9] Provided with Scrappie and from ONT

Scrappie's speed of basecalling varies with the model being used, with the fastest model being `k3_w11_s6_l5_u24` with an average time of 0.026 seconds per read and the slowest being `rgrgr_94` with an average time of 0.117 seconds per read (Figure 4.3). The speed of basecalling with Scrappie has been quite consistent over time, but it struggles to exceed 10^5 bases per second.

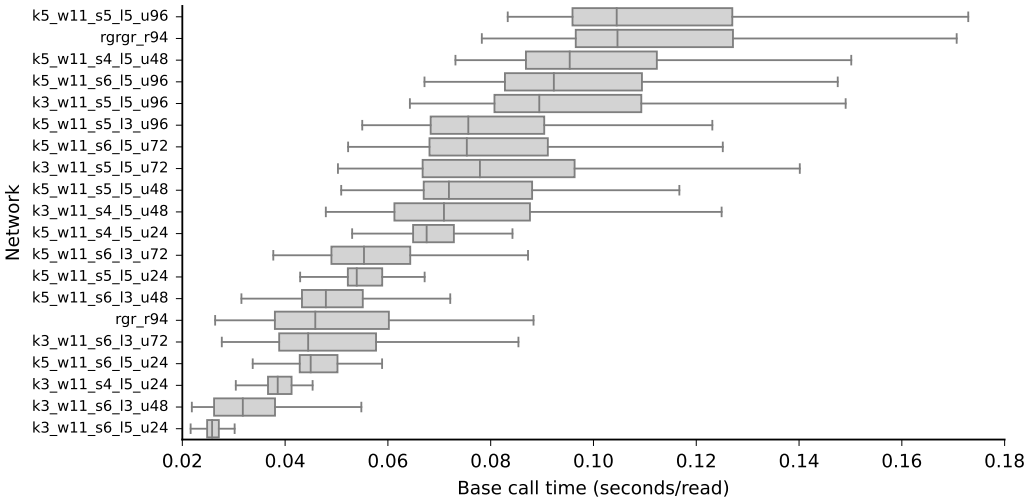


Figure 4.3: Mean Scrappie basecalling speed per model. Different Scrappie base calling models have varying basecalling speeds, ranging from 0.026 to 0.117 seconds per read. Some models, particularly those using the 5-mer model, have greater variability and take longer. While Scrappie is capable of calling reads quickly it still may not be fast enough, as at the fastest speed of 0.026 seconds per read a full flow cell (512 sequencing channels) of data would take > 13 seconds to basecall.

4.2.4 readfish

In tandem to assessing the performance of Scrappie, the initial scripts that would become readfish were being written. These earliest scripts only attempted basic enrichment or depletion. Eventually, a flexible configuration schema was implemented that allowed running different experiments on the same sequencing library simultaneously.

Readfish operates by receiving a series of targets for the experiment, supplied in a configuration TOML file. This file specifies what basecaller to use and the required parameters, what reference genome to use, and what regions of the reference are to be selected for (enriched) or selected against (depleted).

With a suitable configuration readfish has the required information to start selective sequencing. In general this follows the procedure (also outlined in Figure 4.4):

1. Initialise a connection to the MinKNOW API via the Read Until API

2. Initialise a connection to a basecaller
3. Initialise a minimap2 aligner with a pre-computed reference file
4. Begin streaming live data from the live sequencing experiment
5. For each iteration of the Read Until API chunk cycle:
   a) Stream raw signal from the current batch of read chunks to the basecaller
   b) Stream the returned basecalled data directly to the aligner
   c) Stream the alignment results back to the readfish programme
   d) Parse the alignments against the experiment configuration, determining whether the read is on or off target, then pass the decision to the Read Until API for it to be effected on the sequencer
   e) (Optionally) Check for updated configuration parameters with a new reference or targets

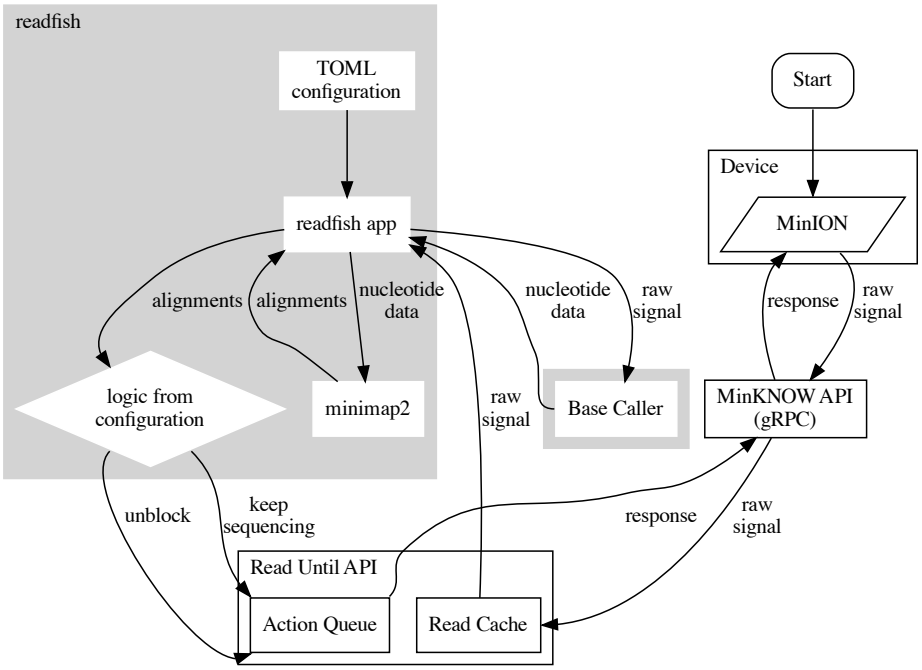


Figure 4.4: Flow diagram showing the additional components that readfish includes which are highlighted in grey. The readfish application draws raw signal from the read cache where it is packaged for basecalling. As base called data are received back they are immediately dispatched to minimap2 which aligns them with the reference supplied in the TOML configuration. These alignments are then passed through the selective sequencing logic that is determined from the TOML configuration and a decision is sent to the action queue and on to MinKNOW and the sequencing device.

Experimental configuration — TOML files

The TOML configuration requires the experiment settings to be defined at the beginning of the experiment. In this schema a single reference genome is used for the entire flow cell and then experiment regions (conditions) are specified. The number of regions must be a factor of the flow cell dimensions so that an even number of pores can be assigned to each region. Each region is either a control region or specifies it's own strategy for selective sequencing. By setting a strategy per region different targets can be used in each experimental condition. Moreover, each region can set the actions that happen in response to each of the available classifications (Table 4.1).

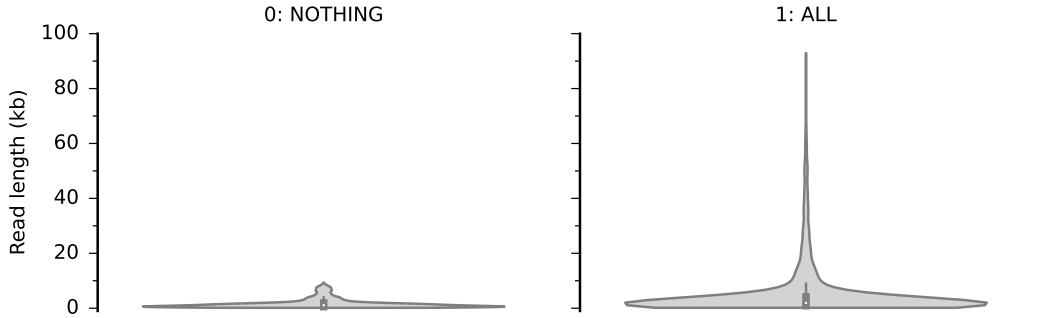
Table 4.1: Possible classification states for a read chunk in readfish. All non-control read chunks will be assigned one of these states depending on how many alignments to the reference genome they (individually) have and whether the locus of any of the alignments is within a region specified in a target list. `no_seq` is a specific case that only occurs when basecalling fails.

Classification	Alignment	
	number	in targets list
`single_on`	= 1	Yes
`single_off`	= 1	No
`multi_on`	> 1	Yes
`multi_off`	> 1	No
`no_map`	0	N/A
`no_seq`	N/A	

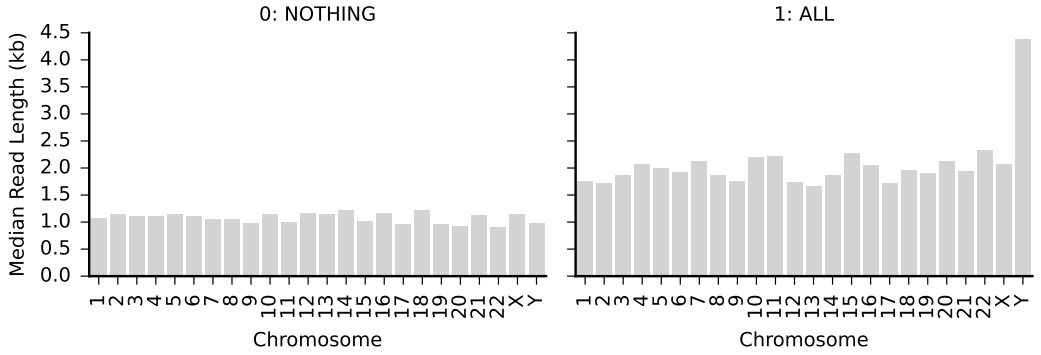
Unblocking half a flow cell

To test the performance of real-time basecalling with Scrappie for enrichment and depletion a playback experiment was setup. The flow cell was divided into two halves based on each channel's position on the flow cell surface. On the left half of the flow cell all reads would be rejected, while on the right half all reads would accepted. As all reads are basecalled and aligned during this experiment the unblock efficiency can be observed by looking at just the left half (`NOTHING`). This is the case as instead of being assigned as a `control` region the `NOTHING` region processed all read chunks as normal and always sent an unblock response.

As a proof-of-concept, this experiment worked with a clear difference in the read-length distributions between the conditions (Figure 4.5a). However, there was little extra data gathered from the accept all portion of the flow cell as the median read length was ~1 kb longer than those seen in the rejected condition (Figure 4.5b).



(a)

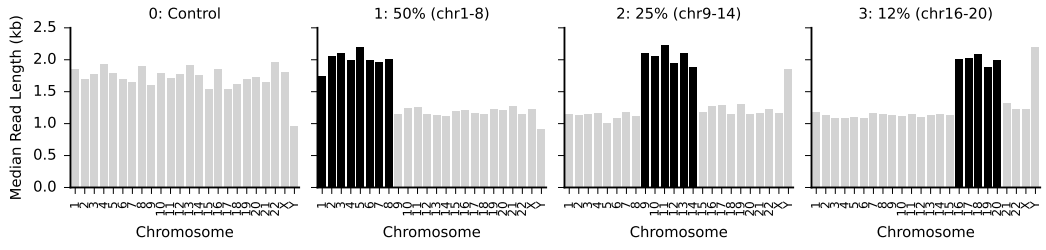


(b)

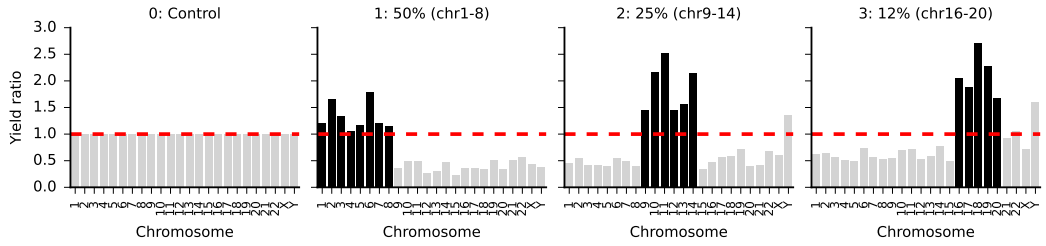
Figure 4.5: (a) Violin plot of read lengths per flow cell condition. On the left is the reject all condition that sends an unblock signal after basecalling and aligning the read chunks; on the right is the accept all condition which accepts all reads after basecalling and aligning. The reject all condition has a mean read length of 1 672 bases while accept all has a mean of 5 449 bases. (b) Median read length in the two conditions. The median read length in the accept all condition is not far from the reject all condition.

Scaling to the human genome

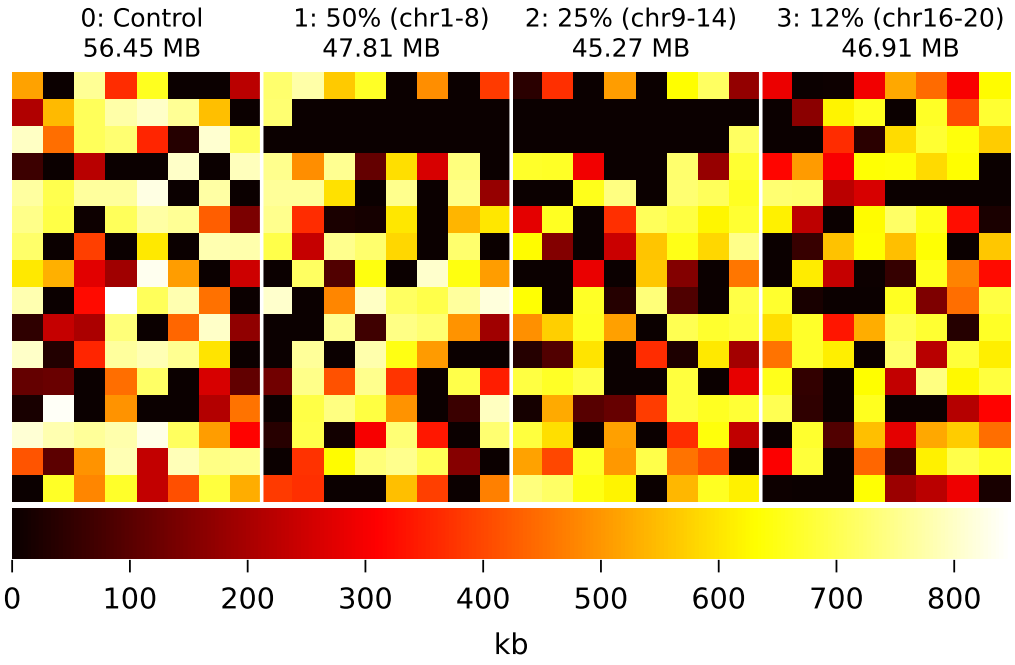
As an initial test of using real-time basecalling for selective sequencing using a gigabase-sized reference, an experiment was setup that divided the flow cell surface into four quadrants. Each of these quadrants was assigned a condition: control (no selection), chromosomes 1–8 (50% of reads accepted), chromosomes 9–14 (25% of reads accepted), and finally chromosomes 16–20 (12.5% of reads accepted). The Read Until API allows channels to be excluded from selective sequencing, however these can only be specified as a range of included channels. Because the actual flow cell layout is not contiguous this method of setting aside control channels is incompatible with how readfish divides the flow cell. Therefore, all data were processed from all channels throughout the duration of the experiment. As read chunks were made available they were basecalled by Scrappie and aligned to the human reference genome by minimap2 (hg38, excluding alternate and unplaced chromosomes). After the run finished all completed FASTQ data were aligned to the same reference used during selective sequencing and the median read lengths per chromosome were plotted (Figure 4.6a). The median read length for reads on targeted contigs closely matches the median lengths seen in the control region (all ~2 kb) while the rejected reads in conditions half, quarter, and eighth are shorter (~1–1.5 kb) (Figure 4.6a). In addition to subsets of chromosomes having the expected median read length as the control Figure 4.6b shows the yield ratio for each chromosome compared to the control region. All targeted chromosomes sequenced as well as those in the control region, with most sequencing $> 1.5\times$ the amount seen on the control.



(a)



(b)



(c)

Figure 4.6: (a) Median read lengths for reads in each quadrant aligned to hg38. The panels are organised as the quadrants were on the flow cell. In the control, all reads are sequenced; in the second, third and fourth quadrants, reads mapping to chromosomes 1–8, 9–14 and 16–20, respectively, are sequenced. The combined length of each of these target sets equates to approximately $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ of the human genome, respectively. The chromosomes that were targeted in each section are highlighted in black. (b) The yield ratio for each chromosome in each condition normalized against the yield observed for each chromosome in the control quadrant. (c) Channel heat map of throughput for each sequencing channel on the flow cell surface.

Guppy

About the time that these experiments with Scrappie and readfish were being developed, Guppy version 3.0 was released with wider GPU support. As Guppy is a much more performant basecaller than Scrappie (Wick et al., 2019) we sought to use it for the real-time basecalling step in readfish. Like Scrappie, Guppy uses recurrent neural networks (RNNs) for their base calling models; however, the architecture of Guppy's models is proprietary and unknown. Guppy's speed of basecalling is also dependent on the model being used, either "fast" or "hac" (high accuracy). Guppy's models are tied to it's version so can vary between upgrades. Moreover, as Guppy utilises GPUs it's performance is tied to the "compute capability" of the hardware in use (Figure 4.7). An upgrade from Guppy version `3.4.5` to `3.6.0` increased the size of the underlying models, leading to larger read batches in Read Until. These larger read batches take longer to basecall and therefore accumulate more data with each read batch. As more read chunks become available for analysis basecalling progressively becomes slower until it cannot keep up with real-time sequencing. The chunks in these larger batches also contain more data and so take even longer to basecall (Figure 4.7 and Table 4.2).

Table 4.2: Descriptive statistics for basecalling time. The NVIDIA GTX 1080 and Quadro GV100 GPUs have different compute capabilities (6.1 and 7.0 respectively) which allows the GV100 to process more data faster using the same underlying RNN model.

			Time (seconds)			
			mean	SD	min	max
GPU	Guppy	Model				
1080	3.4.5	fast	0.139	0.083	0.025	0.413
		hac	1.631	0.470	0.686	3.293
	3.6.0	fast	1.459	1.887	0.059	6.354
		hac	19.713	22.127	1.147	87.555
GV100	3.4.5	fast	0.070	0.026	0.014	0.183
		hac	0.393	0.178	0.107	0.858
	3.6.0	fast	0.553	0.441	0.058	1.404
		hac	3.723	6.033	0.232	46.602

As Guppy is the current state of the art basecaller for ONT data we opted to use Guppy for the real-time basecalling in readfish. This is made possible through the `ont-pyguppy-client-lib` (ONT, 2021), a Python library that enables interactive basecalling. As Guppy is used through a different library interface to Scrappie it

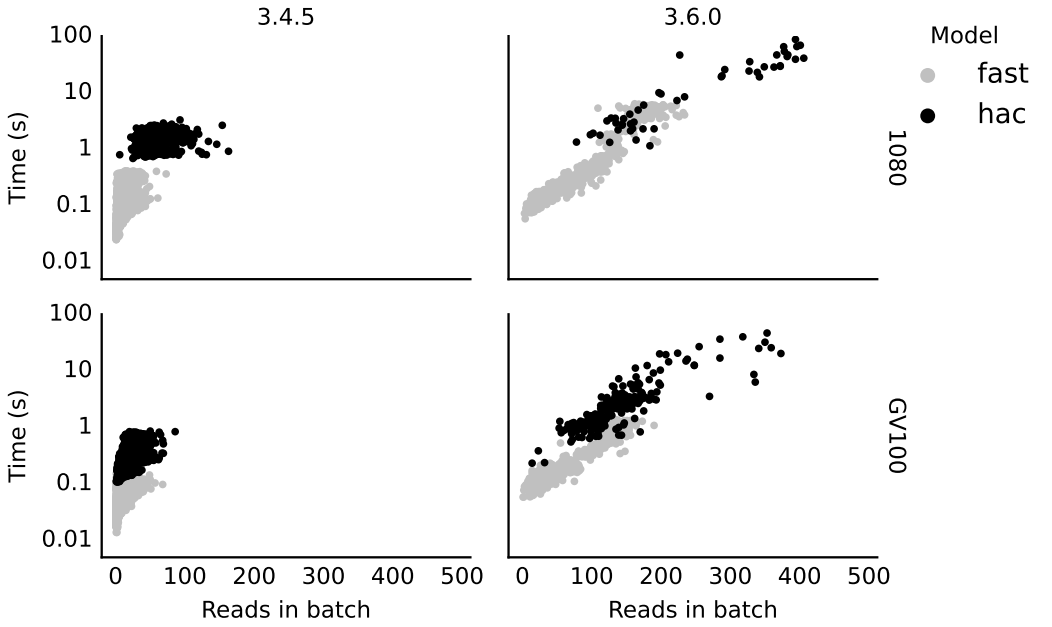
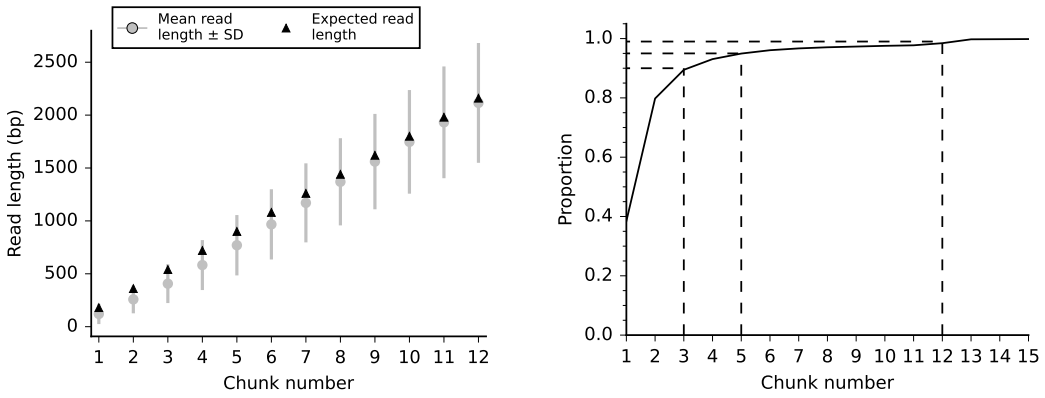


Figure 4.7: Comparison of Guppy `v3.4.5` and `v3.6.0` on the NVIDIA GTX 1080 and Quadro GV100 GPUs. Using playback and the Read Until API batches of reads were retrieved as they would be during a normal experiment and basecalled. The time to basecall all the read chunks in each batch was recorded. Different calling models are shown hac (high accuracy) in black and fast in grey. The size of the basecalling models increased between these two Guppy versions and this can be seen in the time that it takes to basecall batches of reads. With Guppy `v3.4.5` all batches took less than 10 seconds, in contrast, batches of reads took between 10 and 100 seconds to call with Guppy `v3.6.0`.

required restructuring and refactoring parts of readfish to allow the use of `ont-pyguppy-client-lib`.

Getting smaller chunks of data from MinKNOW

When running selective sequencing the requirement to inspect molecules at frequent intervals needs to be balanced with what the computer is capable of supplying. Chunks of live signal, by default, are one second in duration. This however, can be altered prior to beginning an experiment. We found that a chunk duration of 0.4 seconds was ideal when sequencing with a MinION flow cell (512 pores) as this yields ~180 bases of nucleotides per iteration (Figure 4.8a). This allowed reads to be placed within the first few chunks that are inspected (Figure 4.8b). In typical experiments 90% of reads are processed (called, mapped, and a decision made) within three chunks, ~1.2 seconds (Figure 4.8b).



(a) (b)

Figure 4.8: (a) Mean read length per chunk, error bars show standard deviation from the mean, and red triangles show expected read length for a given chunk, calculated as *chunk number* × *chunk duration* × *bases per second*. So a chunk duration of 0.4 seconds and a sequencing speed of ~450 b/s results in $\{180, 360, 540, \ldots, N\}$. (b) Proportion of read fragments processed in a given number of chunks. 90% of reads are processed in 3 chunks, 95% in 5 chunks and 99% in 12 chunks.

4.2.5 Human chromosome enrichment

Once Guppy had been integrated with readfish it's performance needed to be quantified. To do this the previous experiment, splitting the flow cell into fractions of the human genome (Section 4.2.4), was repeated. As before, the flow cell surface was divided into the same four quadrants: control (no selection), chromosomes 1–8 (50% of reads accepted), chromosomes 9–14 (25% of reads accepted), and finally chromosomes 16–20 (12.5% of reads accepted). Selectively sequenced reads have a median read length of ~15 kb (Figure 4.9a); while rejected reads have a median read length of ~500 bases, equating to ~1.1 seconds of sequencing time (Figure 4.8b).

This run generated 9.5 Gb of sequence data, which was unevenly distributed across the quadrants; 3.47 Gb in the control, 2.79 Gb at 50% acceptance,1.84 Gb at 25% acceptance and only 1.22 Gb at 12% (Figure 4.9c). For each quadrant the optimal enrichment is 2-fold, 4-fold and 8-fold but observed enrichment is lower, most likely due to reduced yield (Figures 4.9b and 4.9c).

Analysis of available channels contributing to data generation shows that sequencing capacity is lost faster as more reads are rejected(Figure 4.9d). We did not nuclease flush the flow cell, as this was not currently available. Though, this should increase throughput and enrichment as it recovers lost sequencing capacity due to blocked pores.

4.2.6 *trans*-nuclease flow cells

An alternative to washing the *cis* surface of the flow cell is a *trans*-nuclease. A *trans*-nuclease is the application of nuclease enzymes to the *trans* well of the flow cell. This nuclease would act on strands of nucleotides that have already begun to progress through a nanopore in a similar manner to the enzyme in Section 2.1.6. These sequenced strands would be cleaved in the *trans* compartment of the flow cell. This would, in principle, reduce the length of the molecules that need to be ejected from a nanopore when unblocking and reduce the number of pores going into the "recovering" or "unavailable" states (Table 3.1). This would have the effect of increasing flow cell throughput without the need to flush and reload — reducing the amount of sequencing library needed.

To test this out, ONT provided a custom flow cell that included a *trans*-nuclease. We used our initial human-chromosome quadrants experiment as a template. In this instance the experimental conditions are reversed with "control" on the right and 12% on the left.

This run generated 1.184 Gb of data in 18 hours. With yields comparable to the initial experiment with Guppy in 12% and 25% conditions (Figure 4.10b). The 50%

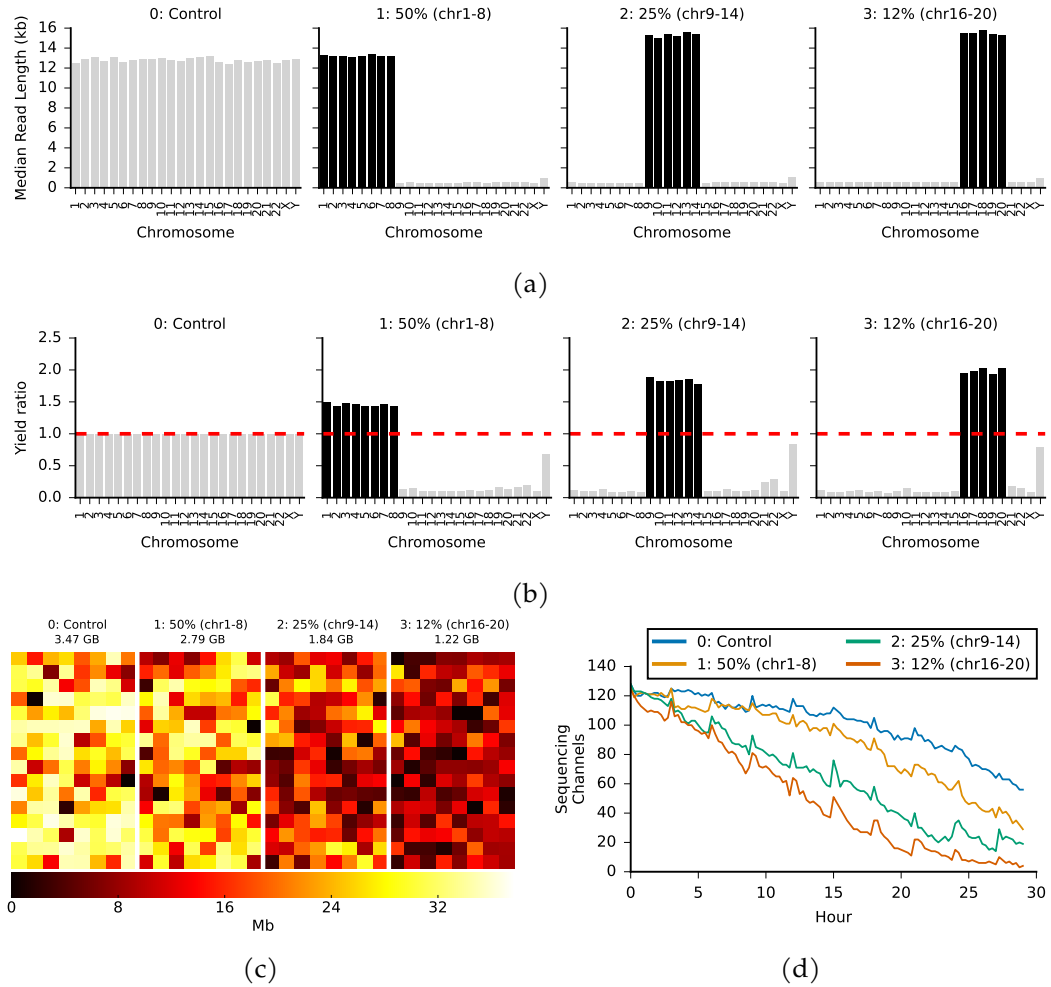


Figure 4.9: (a) Median read lengths for reads sequenced from NA12878 and mapped against hg38 excluding alt chromosomes. The panels are organised as the quadrants were on the flow cell. In the control, all reads are sequenced; in the second, third and fourth quadrants, reads mapping to chromosomes 1–8, 9–14 and 16–20, respectively, are sequenced. The combined length of each of these target sets equates to approximately $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ of the human genome, respectively. (b) Yield ratio for each chromosome normalised against the yield observed in the control quadrant. (c) Heat map of throughput per channel in each quadrant on the flow cell. As the proportion of the genome being rejected increases (left to right) the yield decreases. (d) A plot of the number of channels contributing sequence data over the course of the sequencing run. Channels are lost at a greater rate when more reads are rejected.

condition, however, sequenced less relative to the control, and so did not enrich for these targets at all. In addition the median read length for on-target reads was variable in across the experimental conditions (Figure 4.10a) with the 12% condition having a median on-target read length of ~5.8 kb, while the 25% and 50% conditions had ~8.8 kb and ~8.4 kb respectively.

Due to the extremely low throughput compared to the other iterations of this experiment (Section 4.2.5) *trans*-nuclease flow cells are not a viable alternative to nuclease flushing and reloading at the moment.

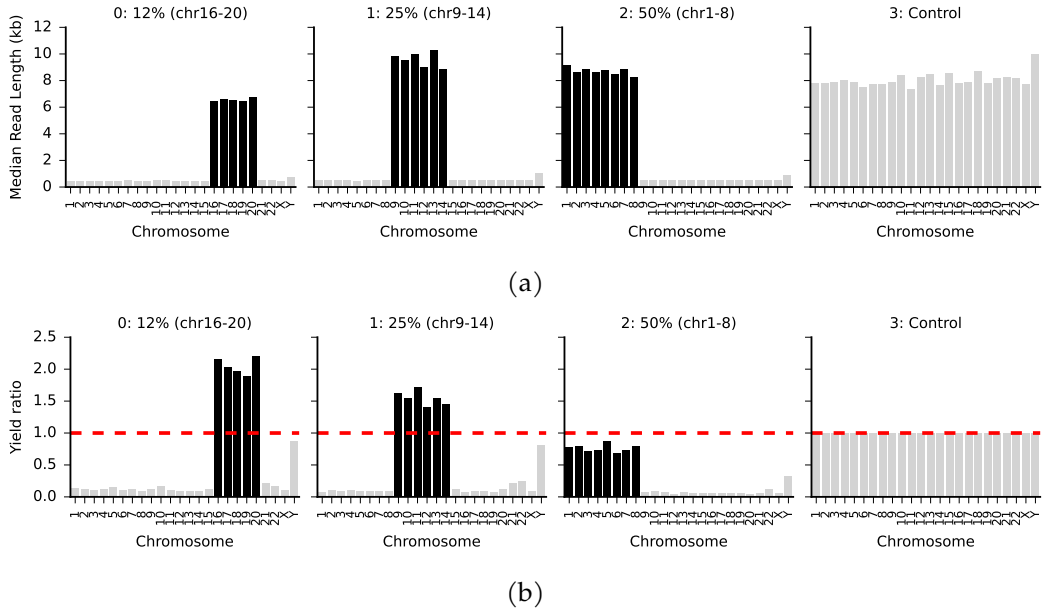


Figure 4.10: *trans*-nuclease flow cell: (a) Median read lengths for reads sequenced from NA12878 and mapped against hg38 excluding alt chromosomes. The panels are organised as the quadrants were on the flow cell. In regions 0, 1, and 2 reads mapping to chromosomes 16–20, 9–14 and 1–8, respectively, are sequenced. In the control region all reads are sequenced. (b) Yield ratio for each chromosome normalised against the yield observed in the control quadrant.

4.2.7 Nuclease flushed flow cell

Another approach to improve throughput and yield on a sequencing run is to nuclease flush and wash the flow cell (Section 2.1.6). Here the quadrants experiment was repeated with a nuclease flush and library reload every 24 h. The same quadrants were used generating a total of 30.5 Gb of data in 72 h. Like the initial run using Guppy (Section 4.2.5) the yield was spread unevenly across the flow cell (Figure 4.11c) with 9.37 Gb in the control region; 8.70 Gb at 50% acceptance; 6.38 Gb at 25% acceptance; and 4.78 Gb at 12.5% acceptance.

Across the entire flow cell, reads that were rejected had a median read length of 458 b while selected reads had a median read length of ~6.6 kb (Figure 4.11a).

4.2.8 Exon enrichment

Despite the cost of sequencing dropping year on year, it is still too expensive to carry out whole genome re-sequencing at a large scale; for example the cost of sequencing a human genome is roughly $1,000[10]. Instead, looking at a relevant genomic subset is more cost effective. One genome-wide target set is the exome, all the protein coding regions, which encompasses ~1–2 % of the human genome (International Human Genome Sequencing, 2001, 2004; Venter et al., 2001).

Initial efforts to sequence human exomes required the generation of exome capture arrays with 164,007 regions (Ng et al., 2009). In contrast, selective sequencing of exonic regions using readfish only requires the curation of target coordinates. These target coordinates were selected by identifying protein coding genes from the human genome (GRCh38) excluding X, Y, and alternate chromosomes (Section 2.2.1). In total, 19,296 genes were identified. Of these genes ~10,000 were selected as they are situated on odd numbered chromosomes. Each target region was expanded by 3 kb both upstream and downstream; overlapping targets were then merged into single regions. This resulted in 25,600 targets covering a selected region of ~176 Mb, which is ~5 % of the human genome.

Two sequencing runs were conducted using a single MinION flow cell, with a nuclease flush performed at 24 h. Both runs used MinKNOW (for GridION) version 3.6.0 and Guppy (GPU) version 3.2.8. Readfish was run as normal, with only one required change to MinKNOW's configuration: setting `break_reads_after_seconds` to 0.4; to ensure that reads can be unblocked before they have sequenced too much. The initial run started with 1,640 pores available for sequencing and finished with 286 pores at the last mux scan. Nuclease flushing and reloading additional library restored 791 pores (1,077 pores total). This complete sequencing run yielded 11.68 Gbp of sequence data; of which 8.5 Gbp were selectively sequenced and 3.18 Gbp were unblocked (Table 4.3). Both the sequenced and unblocked subsets are similar between each run with a mean read length of 7,794.2 and 7,941.4 bases in the sequenced groups and 509.6 and 521.6 bases in the unblocked groups (Table 4.3 and Figure 4.12a). The unblocked subsets consist of ~5–6× more reads than the sequenced subset; which has ~2.5–3× greater yield concentrated on exons on odd chromosomes.

[10]https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

Table 4.3: Exon enrichment run summary statistics from NanoStat for both the run before the nuclease flush and reload (Run 1) and after (Run 2). Each run is split into the sequenced and unblocked subsets and the whole run. Both Run 1 and Run2 performed similarly as can be seen by the consistent read length and quality values.

	Run 1		
	Sequenced	Unblocked	Complete run
Active channels:	512	512	512
Mean read length:	7,794.2	509.6	1,527
Median read length:	7,059.0	414.0	440.0
Mean read quality:	11.1	11.4	11.4
Median read quality:	12.2	11.8	11.9
Number of reads:	607,000	3,737,288	4,344,288
Read length N50:	11,641.0	500.0	8,949.0
Total bases:	4,731,092,828	1,904,399,825	6,635,492,653
	Run 2		
Active channels:	506	499	506
Mean read length:	7,941.4	521.6	1,731.1
Median read length:	7,241.0	412.0	444.0
Mean read quality:	10.9	11.0	11.0
Median read quality:	11.7	11.4	11.5
Number of reads:	474,564	2,436,819	2,911,383
Read length N50:	11,463.0	511.0	9,238.0
Total bases:	3,768,718,814	1,271,162,488	5,039,881,302

Exons had a average median coverage of 17.23× (mean 17.40×) and 0.98× (mean 1.22×) for target and control exons respectively (Figures 4.13a and 4.13b and Table 4.4). 99% of exon targets had a coverage >7.17× compared with 99% of control regions having a coverage of ≤4.34× (Figure 4.12b and Table 4.4).

Table 4.4: Mean coverage for exon targets (odd chromosomes) and exon controls (even chromosomes). Quantile 50% represents the median. Exon targets had much greater coverage, with 99% of targets having at least 7.17× coverage.

	Mean coverage				
	mean	std	1%	50%	99%
Exon Controls	1.22	1.09	0.00	0.98	4.34
Exon Targets	17.40	5.02	7.17	17.23	29.77

Using just computational selective sequencing we were able to select ~5% of the human genome, by only sequencing exonic regions on odd numbered chromosomes. With the addition of nuclease flushing and reloading lost sequencing capacity can be recovered, this can be seen in Figure 4.14 as the experiment progresses fewer reads are seen in each successive read batch until the restart. Readfish is also able to keep up with the rate of live data generation on the device as the mean time for processing a read batch never exceeds the 0.4 s period that they are sent on (Figure 4.14e).

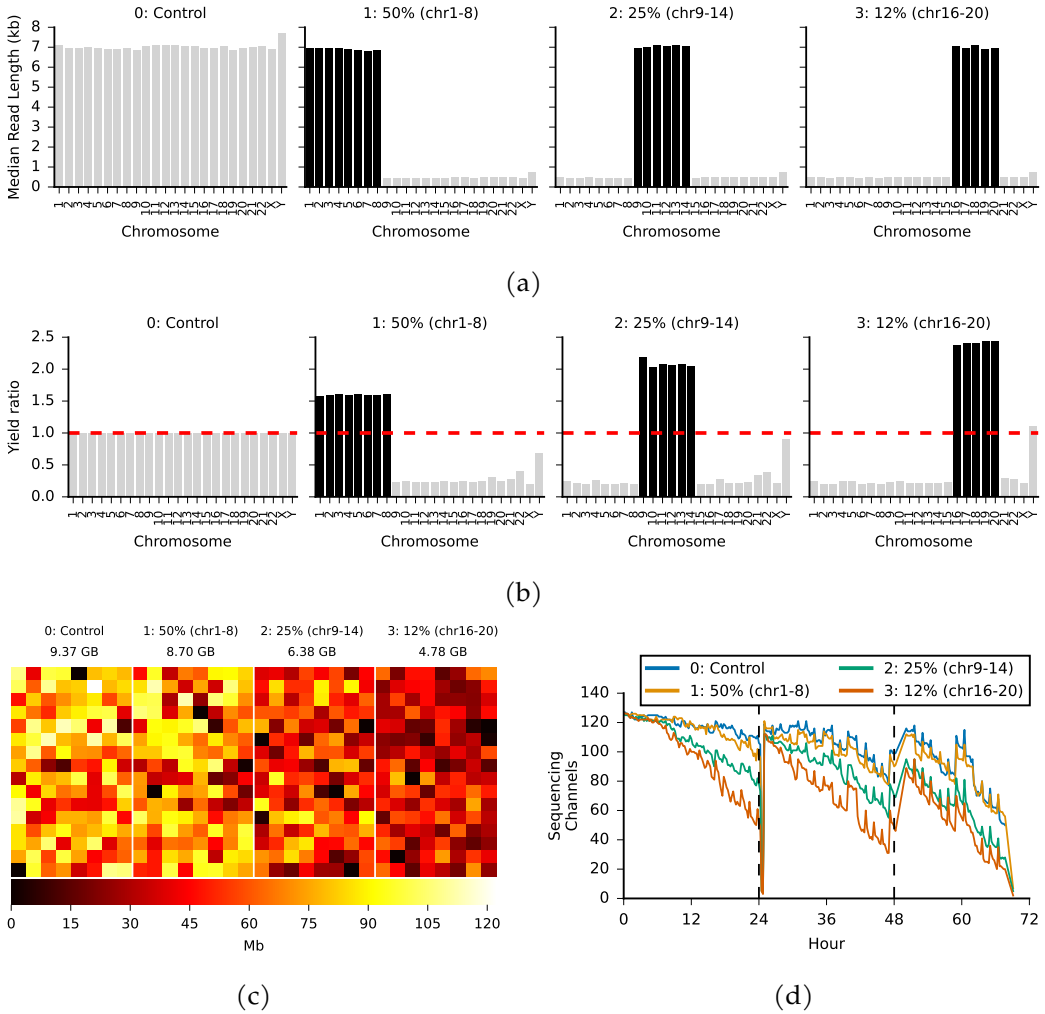
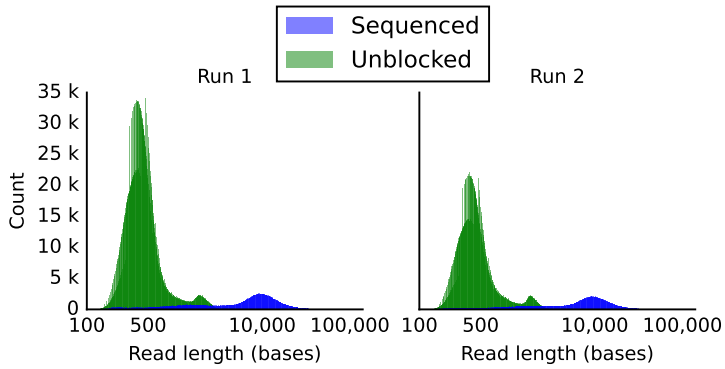
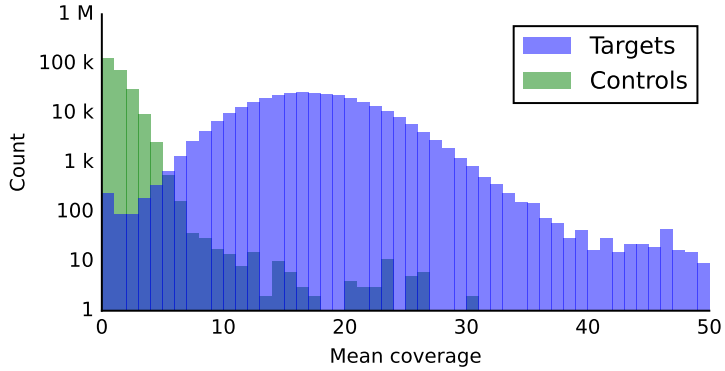


Figure 4.11: (a) Median read lengths for reads sequenced from NA12878 and mapped against hg38 excluding alt chromosomes. The panels are organised as the quadrants were on the flow cell as in Section 4.2.5. (b) Yield ratio for each chromosome normalised against the yield observed in the control quadrant. (c) Heat map of throughput per channel in each quadrant on the flow cell. As the proportion of the genome being rejected increases (left to right) the yield decreases. (d) A plot of the number of channels contributing sequence data over the course of the sequencing run. Channels are lost at a greater rate when more reads are rejected. But flushing and reloading (dashed lines) restore all regions to the same level.

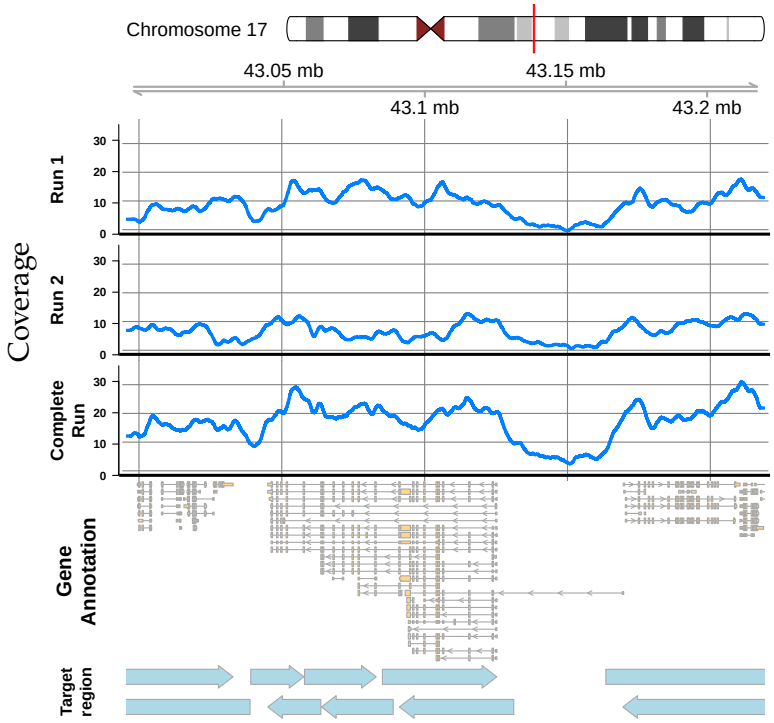


(a)

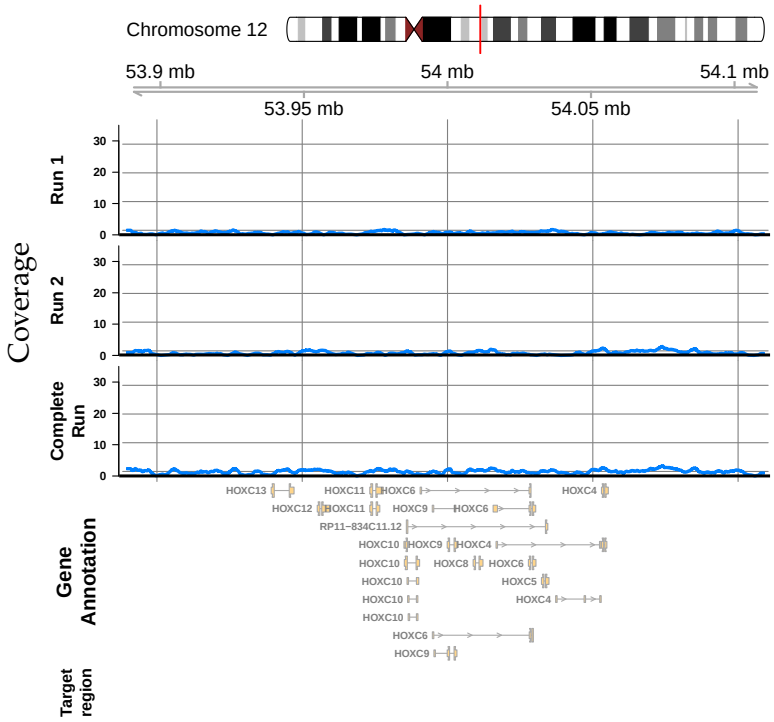


(b)

Figure 4.12: (a) Read length distribution for sequenced and unblocked reads from both runs in the exome enrichment experiment. Here, x-axis is read length (log scale) and y-axis is the count for each bin. As in Table 4.3 the unblocked reads have a much shorter distribution than the sequenced reads, showing that readfish is able to make it's decisions quickly. (b) Distribution of mean coverage for Read Until exon targets (odd chromosomes) and control exons (even chromosomes). >99% of target regions had at least 7.17× coverage, while only 1% of control regions had >4.34× coverage. (Table 4.4)



(a) BRCA1



(b) HOXC4–13

Figure 4.13: Coverage plots for two example loci, BRCA1 and the HOX cluster (covering HOXC4–13). (a) BRCA1 was enriched for as it resides on chromosome 17 and shows greater coverage over exonic regions, shown in the bottom two tracks displaying exons and readfish targets respectively. There is also a visble reduction in coverage (on both Run 1 and Run 2) at ~43.15 mb where there no exons or targets. (b) HOX cluster, which resides on chromosome 12, has coverage close to 0× in both exonic and intergenic regions. Note there are no target regions in (b) as this is an even numbered chromosome.

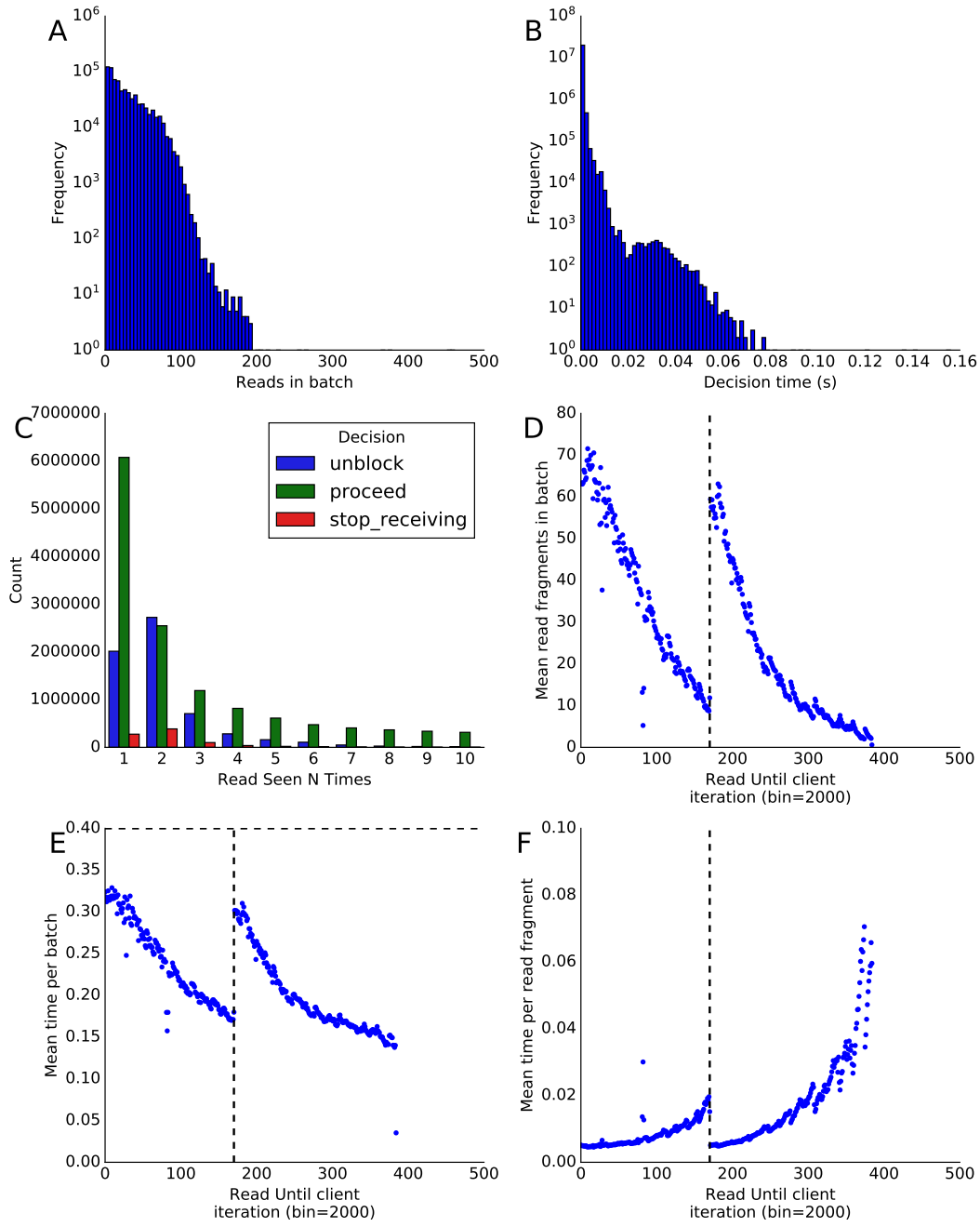


Figure 4.14: Human Exome (a) Histogram of read batch size throughout the selective sequencing program. (b) Histogram of decision times (time to choose unblock, stop receiving, or proceed from an alignment). (c) Counts of decision classifications for read fragments seen a given number of times. (d) Mean batch size, in bins of 2000, seen throughout the selective sequencing program. (e) Mean process time, in bins of 2000, for batches of read fragments throughout the run. (f) Mean decision time per read fragment, in bins of 2000, throughout the run. As the number of reads in a batch reduces, the overhead time of calling becomes more apparent. The vertical dashed lines mark flushing and restart of the run and illustrate the benefits of flushing.

4.2.9 DeepNano-Blitz — CPU basecalling

To allow readfish to be more broadly usable a suitably performant CPU basecaller was required. For this, DeepNano-Blitz was selected as the fastest mode is 100× faster than Guppy (`v3.4.4`, CPU) using the high accuracy model and ~13× faster than Guppy using the fast model (Bož a et al., 2020). DeepNano-Blitz is able to maintain read accuracy while processing much more data, dropping only 4.5% points of median read accuracy comparing DeepNano-Blitz's fastest model with Guppy's fast model. As DeepNano-Blitz does not require a GPU for basecalling this expanded the compatible platforms from just linux to include MacOS and Windows.

With the exception of how the basecaller is initialised there is no difference to how readfish operates when using DeepNano-Blitz as the basecaller. In these experiments we ran the DeepNano-Blitz basecaller on a subset of the human exome (Section 4.2.8) panel that consists of 717 gene targets that are implicated in cancer, from the COSMIC panel (Tate et al., 2018). These experiments were carried out on a variety of platforms that have little or no support for GPU accelerated basecalling, most notably MacOS and Microsoft Windows.

For MacOS there is no specific setup required as DeepNano-Blitz can be compiled natively on this platform. However, on Windows compilation is challenging so a "Windows Subsystem Linux" installation is required. This is a virtual container that allows a linux installation to be operated on a Windows computer. Once WSL2 is setup readfish installation occurs as normal for a UNIX computer.

To compare CPU basecalling with GPU basecalling we set up six experiments. Two utilised Guppy on the NVIDIA Quadro GV100 and GTX 1080 Ti on a GridION Mk1 and a linux workstation respectively. The remaining four CPU runs were split between the GridION Mk1, the two workstations (linux and Windows) and a MacBook Pro (from 2018, with a ~3 GHz i7 processor). In all cases enrichment is comparable to that seen with GPU accelerated basecalling (Table 4.5).

The differences in coverage are mostly the result of differences in yield for each experiment. This can be seen in the GPU experiments, which both have a yield of < 10 Gb, but have the lowest mean and median read lengths (Table 4.5). As a result of more efficient rejection more on-target coverage is achieved. This is further exemplified by the Windows run, which sequenced ~17 Gb in total, but achieves similar coverage to GridION GPU run that had a roughly half the yield.

Table 4.5: Comparison of GPU and CPU basecalling. Mean and Median read length are a proxy for readfish efficiency as when these are lower they correspond to unwanted reads being ejected faster. Basecalling on GPU is the most efficient, but CPU basecalling is performant and delivers enrichment of targets.

	Mean Read Length	Median Read Length	Yield Gb	Mean Coverage of Targets	Coverage SD	Flushes
GridION GPU	735.6	423.0	9.08	31.30	5.54	2
GridION CPU	878.9	662.0	14.93	29.78	5.30	2
Linux GPU	683.7	402.0	5.90	19.11	3.23	2
Linux CPU	771.2	564.0	14.31	27.78	5.09	2
MacBook CPU	1085.0	745.0	14.03	29.08	5.24	2
Windows CPU	1146.9	823.0	17.27	34.47	6.62	3

4.2.10 Selective sequencing with barcoded samples

Enrichment and depletion by barcode

Barcoding is useful when the amount of data needed for a sample is less than the throughput of a single flow cell. By attaching unique tags to ends of dsDNA during library preparation many samples can be sequenced simultaneously (multiplexed). Then, following sequencing each sample can be de-multiplexed by decoding the attached barcodes. By pooling samples on a single flow cell the sequencing capacity is shared between libraries making more efficient use of the flow cell and reducing the cost of sequencing a single sample.

Guppy v4.0 and newer allows a barcode kit to be specified when passing read chunks for basecalling. As such readfish was further developed so that individual barcodes can be configured to be included or excluded instead of selecting based on genomic alignment. This configuration was trialled over three hour-long experiments using four barcoded samples of *Clostridioides difficile*. Initially a control run was conducted to ascertain a baseline for each sample. Then two subsequent experiments were carried out; the first sought to enrich barcodes 8 and 11 while depleting barcodes 9 and 10. The final experiment was the inverse of the second, enriching barcodes 9 and 10 while depleting barcodes 8 and 11.

The control gave ~180 Mb of sequence data, the enrichment for barcodes 8 and 11 gave ~91 Mb, and barcodes 9 and 10 gave ~103 Mb (Table 4.6). The distribution of read lengths in the sequenced conditions is similar to that seen in the control run (Table 4.6 and Figure 4.15a). None of the enriched targets exceeded the yield

seen in the control experiment, however depleted targets were greatly reduced (Figure 4.15b). The read count for each enriched barcode is similar to that seen in the control, with depleted targets diminished to almost zero (Figure 4.15c). In the enrich and deplete conditions the "unclassified" read count is greatly increased due to using a "barcode at both ends" configuration that requires a read to have both a forwards and reverse barcodes to be fully classified. If a read is successfully ejected in response to an unblock request then it will, by definition, not have both barcodes and so will be classed as "unclassified". This is why the "unclassified" category is increased in the experimental conditions.

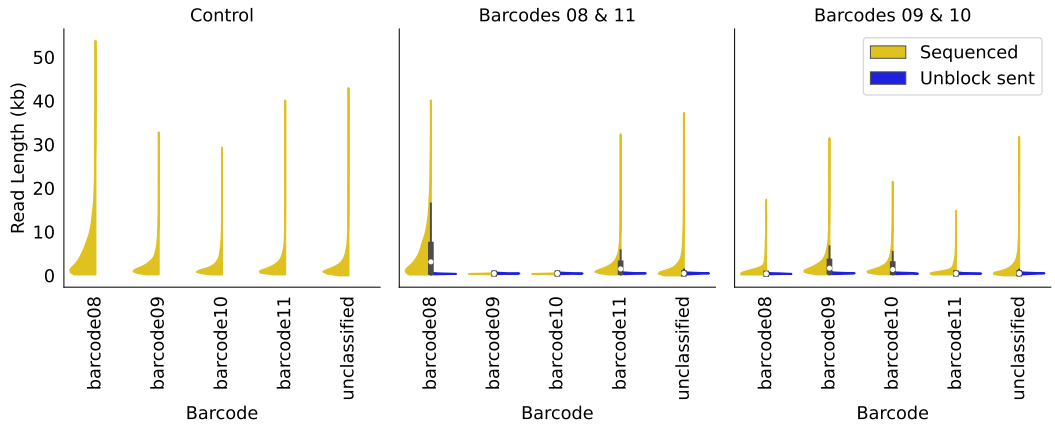
Table 4.6: Read length statistics for each of the enrichment/depletion experiments, subset on whether the reads were enriched ("Sequenced") or depleted ("Unblocked"). The mean and N50 of the sequenced groups are similar to that of the control showing that sub-samples can be enriched without impacting the sequencing library characteristics.

Experiment	Sequenced	Read length		
		Yield (Mb)	Mean	N50
Control	Sequenced	180.75	2,590.70	4267
Barcodes 08 & 11	Sequenced	62.62	2,523.89	4600
	Unblocked	28.67	469.53	498
Barcodes 09 & 10	Sequenced	72.30	2,179.14	3672
	Unblocked	31.38	471.24	501

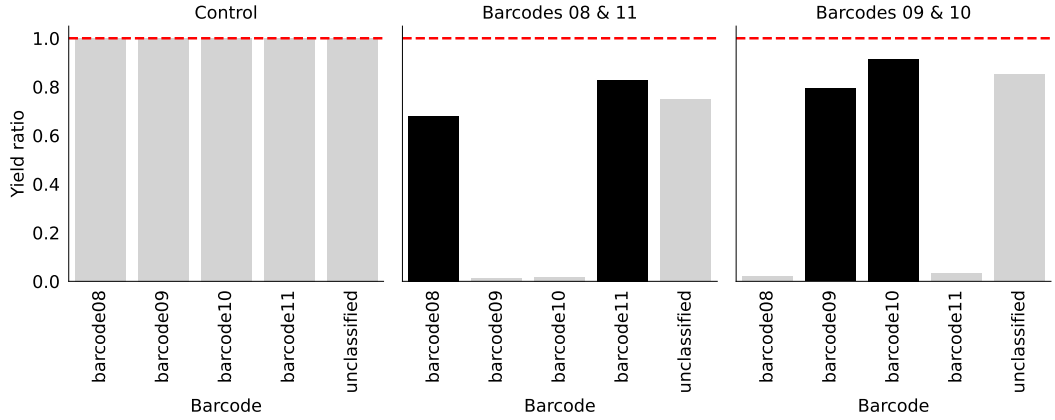
Selective sequencing — barcode specific targets

While it is convenient to be able to stop sequencing entire barcoded samples after a run has started it is more likely that each barcoded sample requires it's own targeting that is more nuanced than on or off. To address this readfish was modified to enable multiple conditions, similar to the quadrants (Sections 4.2.4 and 4.2.5) configuration, but instead all channels are considered and the identified barcode determines the selective sequencing criteria; read chunks that do not receive a barcode classification within four chunks are unblocked as they will likely remain unclassified.

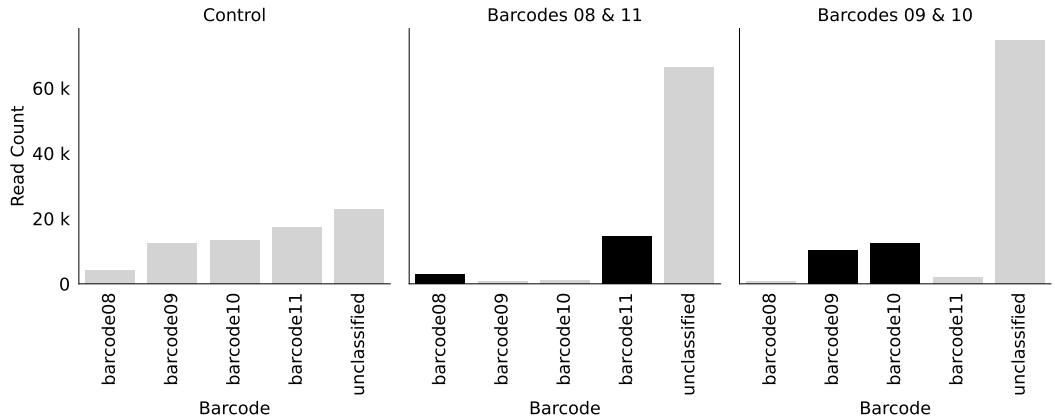
An experiment was set up, using the *C. difficile* sample aiming to enrich quarters of the ~4 Mb genome on each barcode. That is, barcodes 8, 9, 10, and 11 consider the regions 0–1, 1–2, 2–3, and 3–4 Mb respectively. In one hour this experiment generated 384.38 Mb of data which was basecalled and aligned to the *C. difficile* genome used during selective sequencing. The per-base read depth was computed using `samtools depth` (v1.11) for each barcode group subdivided into "sequenced" and



(a)



(b)



(c)

Figure 4.15: Barcode enrichment and depletion. In each figure columns left to right are control, select barcodes 8 and 11, and select barcodes 9 and 10. (a) Read length distributions for each barcode classification (including `unclassified`) split by whether the read was selected ("Sequenced") or rejected ("Unblock sent"). (b) Yield for each experiment and barcode compared to the control experiment. No enriched targets exceeded the yield seen in the control experiment, but depleted targets are greatly reduced compared to the control. (c) Read count for each barcode over each experiment. Enriched targets have similar read count to that seen in the control while depleted targets are greatly reduced.

"unblocked" reads. In barcodes 9, 10, and 11 the mean read depth for the targeted region was ~5× greater in the sequenced categories compared with the unblocked off-target categories (Table 4.7). The target regions are clearly enriched compared to the rest of the genome (Figure 4.16); moreover the target regions have lower read depth in the "Unblocked" category, suggesting that read chunks are correctly identified with both their barcode and genomic position.

Table 4.7: Mean coverage over the *C. difficile* genome split by barcode and by whether reads were sequenced or unblocked.

Barcode	Sequenced	Mean coverage depth			
		on-target		off-target	
		split	total	split	total
barcode08	sequenced	0.197	0.198	0.056	0.162
	unblocked	0.001		0.106	
barcode09	sequenced	7.059	7.220	0.610	1.510
	unblocked	0.161		0.900	
barcode10	sequenced	7.821	7.931	0.750	1.572
	unblocked	0.111		0.821	
barcode11	sequenced	12.328	12.490	0.803	1.815
	unblocked	0.162		1.013	

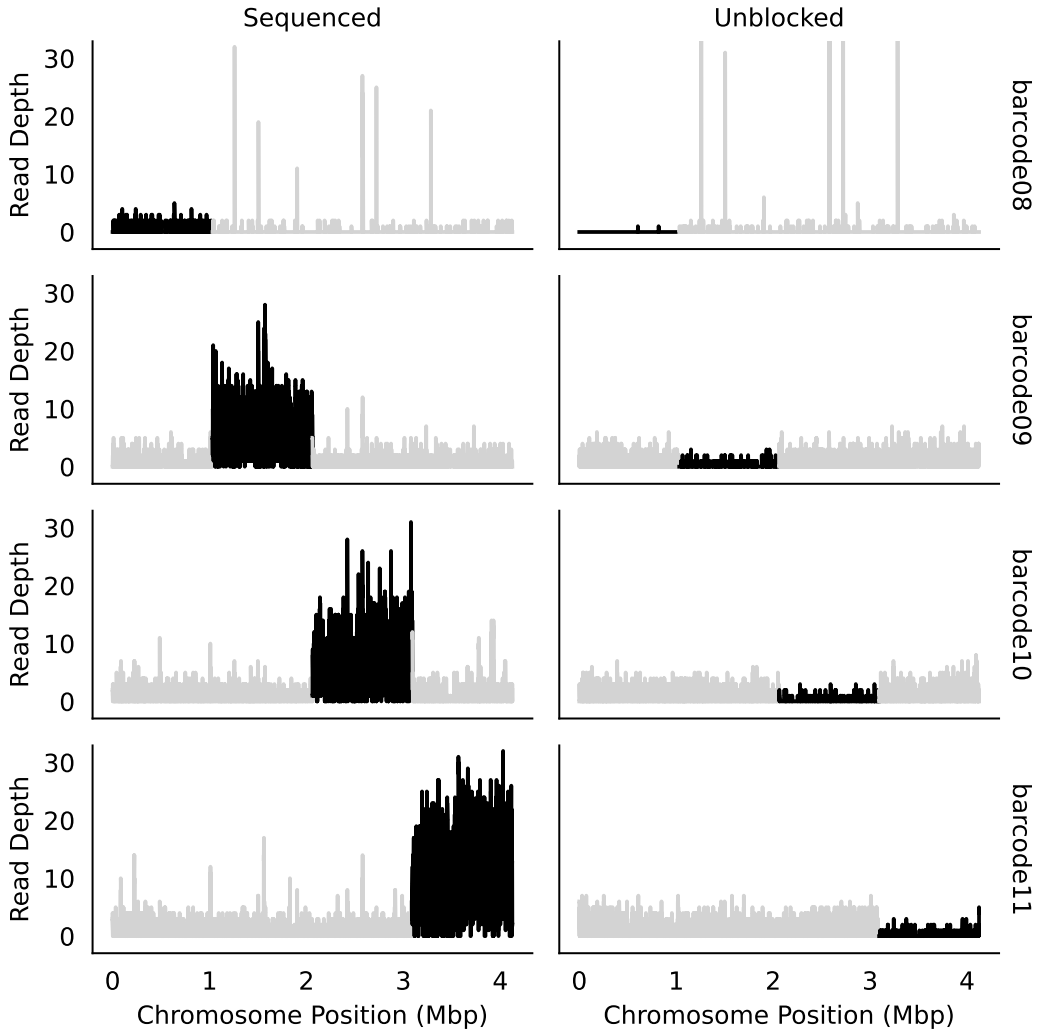


Figure 4.16: Coverage depth over *Clostridioides difficile* per barcode (rows) and split by whether the reads were sequenced (left column) or unblocked (right column). Each barcode was selecting approximately $\frac{1}{4}$ of the *C. difficile* genome; with barcodes 8, 9, 10, and 11 targeting coordinates 0.00–1.03 Mb, 1.03–2.06 Mb, 2.06–3.09 Mb, and 3.09–4.12 Mb respectively. Target regions are labelled in black for each barcode while off-target regions are labelled in grey.

4.3 Conclusion

Selectively sequencing individual molecules using only computational methods is a unique capability of nanopore sequencing. Using real-time basecalling instead of raw signal allows selective sequencing experiments to be carried out on computers capable of live basecalling, rather than using network connected servers for analysis (Loose et al., 2016; Edwards et al., 2019; Kovaka et al., 2020). The careful management of the time that readfish takes while processing real-time data was essential as handling the real-time stream too slowly results in a backlog of unprocessed chunks. Having an unprocessed backlog, in turn, means that readfish is no longer synchronised with the sequencer and cannot effectively carry out selective sequencing. Using a sufficiently fast basecalling and a performant aligner are essential for this approach to work. As demonstrated in the difference between using Scrappie, a CPU basecaller (Figure 4.6) and Guppy, a GPU accelerated basecaller (Figure 4.9). Though, as DeepNano-Blitz demonstrates (Section 4.2.9 and Table 5.3) an optimised CPU basecaller can still be used on platforms where GPU basecalling is not yet enabled.

Throughout the development of readfish we have demonstrated that software-based enrichment is possible and results in real enrichment (Figure 4.11). For selective sequencing in this form to be effective though efficiency is everything. That is, the faster reads can be identified and unblocked, the better; and standard techniques for increasing yield, such as flushing and reloading library on to your flow cell can help recover capacity and improve enrichment. This is particularly evidenced by Scrappie, which shows very little enrichment (Figure 4.6b) due to the speed of basecalling (Figure 4.3); and, to a certain degree, DeepNano-Blitz, which shows acceptable levels of enrichment but is still slower than GPU accelerated basecalling (Table 4.5).

Most importantly, though, readfish has done what it set out to do. Using the processing power available on a single computer — whether it is GPU equipped or not — we are able to enrich for an arbitrary number of targets across gigabase-sized genomes.

In addition the inclusion of real-time barcode demultiplexing allows readfish to work in more common sequencing workflows where samples are pooled. It is not necessary for all barcoded samples to be under selection as this approach is flexible to many different sequencing configurations.

Signal based methods, such as UNCALLED (Kovaka et al., 2020) and Sigmap (Zhang et al., 2021) are able to utilise longer signal references; that is, >5 kb. Though,

UNCALLED's performance still degrades with reference genome length and repetitiveness; using intensive masking procedures on subsets (rather than whole) of the reference genome are essential for UNCALLED to work. Moreover, due to the nature of the pre-processing UNCALLED carries out — at least 30 iterations of *k*mer masking — it is unlikely that UNCALLED will be able to achievable real-time reference or target updates. Sigmap performs raw signal alignment faster than UNCALLED but does not have an interface with real-time data yet (Zhang et al., 2021). It is not likely to be useful for large scale selective sequencing as the signal index generated is ~30–34× larger than the corresponing genomic index.

Chapter 5

Applications of readfish

5.1 Introduction

Readfish is able to enrich arbitrary targets within a library of molecules using a two-step basecalling and alignment process. During the course of developing readfish this approach was trialled by enriching fractions of the human genome, half of the human exome, and a panel of target loci implicated in cancer (COSMIC). The advantage of this approach, over signal based methods, is that other existing tools — such as barcode demultiplexers and metagenomic classifiers — can be incorporated to provide greater information for selective sequencing.

In this chapter we will take a look at more real-world uses for readfish. Considering panels of targets genes, barcode specific gene panels, adaptive control of sequence depth both with and without *a priori* knowledge of the sample.

5.1.1 Work contribution

The majority of the work in this chapter was done by the author apart from DNA library preparation and flow cell flushing and reloading which was carried out by Nadine Holmes in Deep Seq. The initial design and selection of target panels was done in collaboration with Matt Loose. Real-time centrifuge analysis was written in collaboration with Rory Munro and Thomas Clarke. All of the bioinformatics analysis was carried out by the author.

5.2 Gene panels

Making targeted panels with molecular methods, such as CRISPR-Cas9 or PCR amplification, is both time consuming and expensive. With software-based selective sequencing target panels can easily be customised or changed by selecting a new reference genome and target coordinates.

These two experiments, whole exome and a panel of genes implicated in cancer (COSMIC) demonstrate just that. We can provide a minimal set of targets and go from there, then you can re-evaluate and try again.

5.2.1 COSMIC panel

With the ability to target and enrich thousands of genomic loci at a low percentage of the human genome, we sought another target set. We settled on the Catalogue of Somatic Mutations in Cancer (COSMIC) (Tate et al., 2018). This panel consists of 717 genes implicated in somatic cancers. To prepare the targets, they were downloaded and all loci with genomic coordinates had 5 kb added both upstream and downstream. This resulted in a panel of 678 genes covering 89.9 Mb, or ~2.7% of the human genome.

Initially, two sequencing runs were conducted using a single MinION flow cell, with a nuclease flush performed within 24 h. Both runs used MinKNOW (for GridION) version 3.6.0 and Guppy (GPU) version 3.2.8. Readfish was run as normal, setting `break_reads_after_seconds` to 0.4. The initial run started with 1,724 pores available for sequencing and finished with 250 pores at the last mux scan. The first run generated 3.70 Gb of sequence data. After nuclease flush and reload the flow cell generated a further 6.33 Gb, resulting in a total yield of ~10 Gb (Table 5.1).

For these 678 targets, the average coverage was 30.89× while the depleted portion of the genome was at 3.38×; 99% of COSMIC targets are covered at 3.65× while 99% of the rest of the genome is covered at ~1× (Table 5.2).

Table 5.1: COSMIC panel run summary statistics from NanoStat for experiment "ml_032". Split into the sequenced and unblocked subsets for the complete run. Extended table in Table B.1 (Page 145)

	Sequenced	Unblocked	Complete run
Active channels:	511.0	511.0	511.0
Mean read length:	5,848.0	505.6	764.6
Mean read quality:	9.6	11.0	10.9
Median read length:	3,098.0	424.0	430.0
Median read quality:	10.7	11.3	11.3
Number of reads:	636,138.0	12,487,287.0	13,123,425.0
Read length N50:	11,191.0	501.0	855.0
STDEV read length:	6,730.2	334.0	1,902.2
Total bases:	3,720,138,912.0	6,314,097,902.0	10,034,236,814.0

With the COSMIC panel being easy to use for both sequencing and analysis we chose to use it for benchmarking and testing other platforms and with alternate base callers. These other experiments encompass six sequencing runs. Two experiments utilise GPU base calling with Guppy running on a GridION MK1 and a linux workstation; they differ in the GPU used, with the GridION using an NVIDIA Quadro GV100 while the linux workstation was equipped with an NVIDIA GTX 1080 Ti. The

Table 5.2: Coverage of COSMIC panel targets and the rest of the genome. 99% of COSMIC targets are covered at 3.65× while 99% of the rest of the genome is covered at ~1×

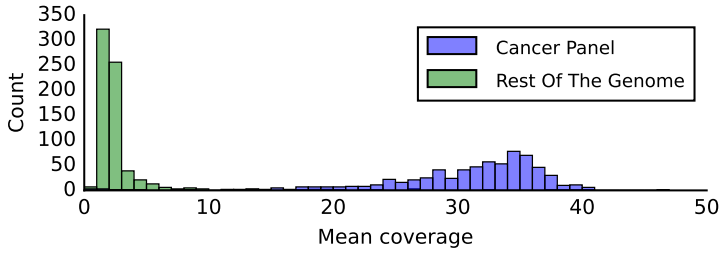
	count	mean	std	1%	50%	99%
Cancer Panel	678	30.89	6.63	3.65	32.28	40.16
Rest of the genome	700	3.38	9.28	1.02	2.03	26.54

four other experiments utilise CPU base calling with DeepNano-Blitz, with each being run on a different computer. Two CPU runs used Ubuntu 16.04 on the GridION MK1 and linux workstation, one run used Ubuntu 18.04 running using Windows Subsystem for Linux (WSL) on a Windows 10 workstation; finally one run used macOS running on a 2018 MacBook Pro.

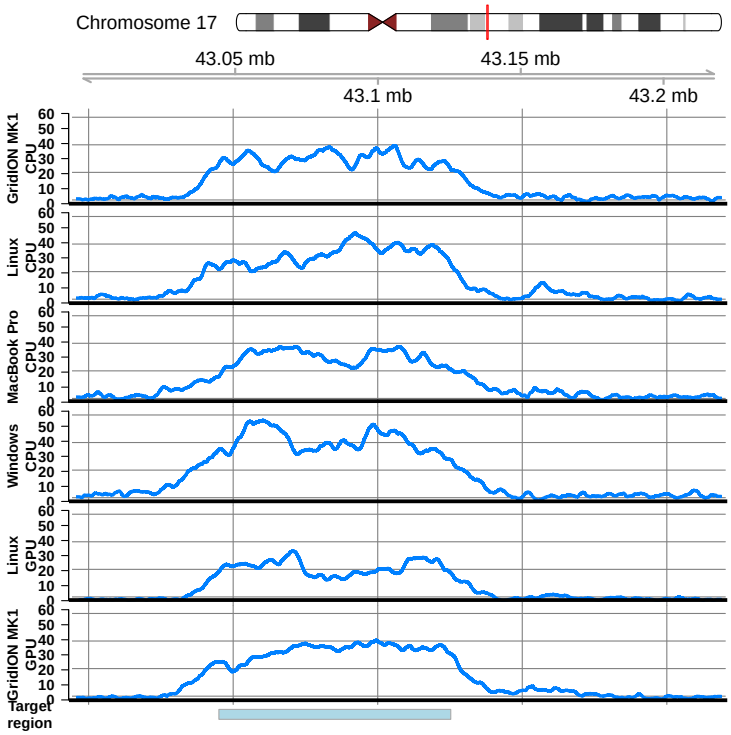
DNA for each run was extracted as in Section 2.1.1. Following extraction DNA was sheared to be in the range 10–20 kb. Each sequencing library was prepared using SQK-LSK109 sequencing kits. Each run was carried out as normal for a targeted sequencing run, with a nuclease flush and library reload every 24 h.

With the exception of the "Linux GPU" run all experiments had good yield, ranging from ~10–18 Gb on GridION using GPU to WSL using CPU respectively (Table 5.3). In addition, each run showed efficient rejection read lengths, indicating that readfish was able to keep up with data generation in real-time and make timely repsonses (Table 5.3). Mean coverage of targets was consistent with flow cell yields, as the worst performing flow cell (Linux GPU) also had the lowest coverage of these runs (Table 5.3). Despite this, mean on-target coverage ranged from 19.19× (Linux GPU) to 34.64× (WSL) (Tables B.7 and 5.3). All platforms were able to enrich target loci compared to the rest of the genome (Figures 5.1b to 5.1e).

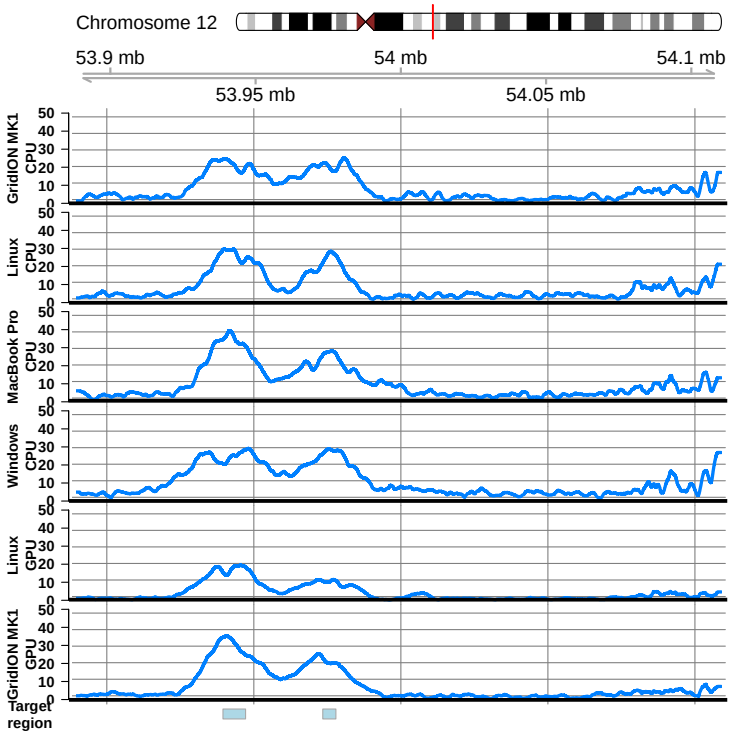
The difference in coverage between each run is primarily as a result of yield for each flow cell. However, both the runs utilising GPU base calling have much lower yield but have comparable target enrichment to CPU base calling runs. This is down to the time it takes to unblock a read, as the faster an unwanted read can be ejected the more other molecules can be sampled. In addition, faster unblock times likely reduce the wear on flow cells that repeatededly unblocking can cause as there is less opportunity for molecules to become tangled and block the nanopore from further sequencing.



(a)

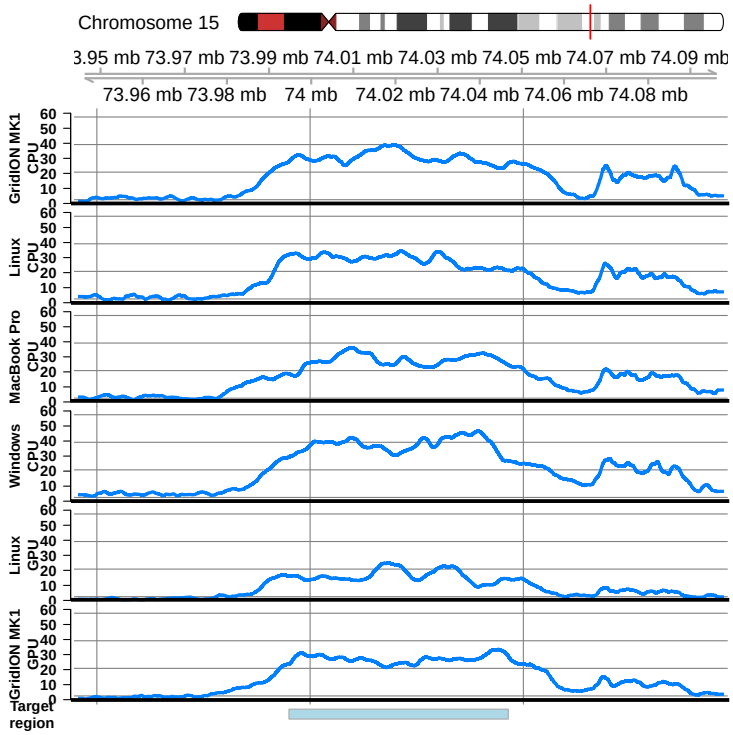


(b) BRCA1

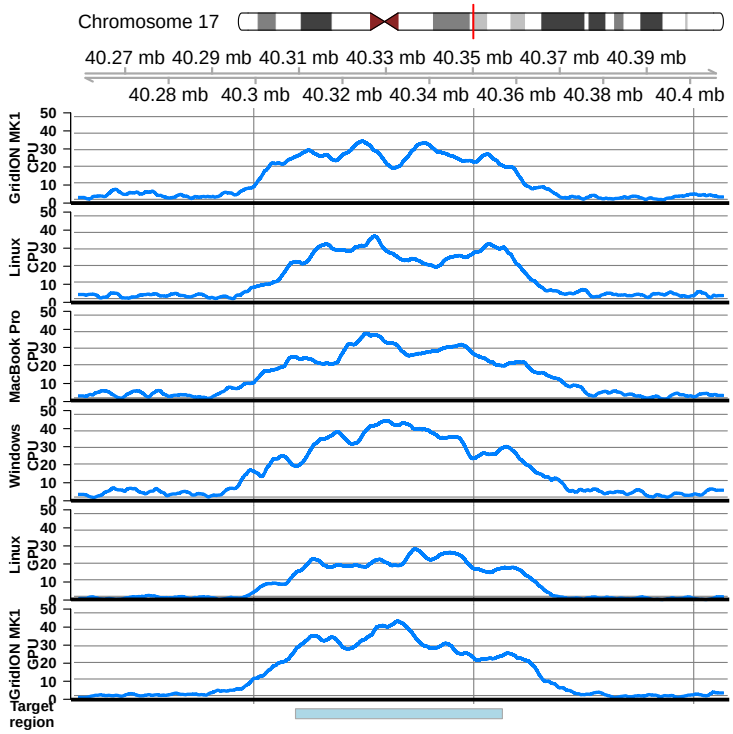


(c) HOXC11 & 13

Figure 5.1: Continued of the following page.



(d) PML



(e) RARA

Figure 5.1: (a) Mean coverage over all genes found in the COSMIC panel and the rest of the human genome. Panel targets have a mean coverage of 30.89× compared with a mean coverage of 3.38× in off-target regions (for a single GridION run). Coverage plots for four example loci, BRCA1 (b), HOXC11 & HOXC13 (c), PML (d), and RARA (e). Coverage plots in descending order are: GridION CPU, Linux CPU, MacBook Pro CPU, Windows CPU, Linux GPU, and GridION GPU. (b,c) both have a window of ~220 kbp and (d, e) both have a window of ~150 kbp; coverage aligns well with the selected region (bottom track on each plot) and in each run coverage over the targets is approaching 40×.

Table 5.3: Run statistics for GPU and CPU base calling experiments, calculated using NanoStat. Mean read length is split into sequenced and unblocked subsets as determined by readfish logs. Rejection is most efficient when using GPU base calling (highlighted), but CPU base calling is still performant enough for targeted sequencing.

	Mean Read Length		Yield	Mean	Flushes
	Sequenced	Unblocked	(Gb)	Coverage	
GridION MK1 CPU	3,773.8	667.7	14.93	29.78	2
GridION MK1 GPU	5,848.0	505.6	9.08	31.30	2
Linux CPU	2,517.9	625.5	14.31	27.78	2
Linux GPU	4,792.7	486.8	5.90	19.11	2
MacBook Pro CPU	3,524.0	891.3	14.03	29.08	2
Windows CPU	3,003.9	975.9	17.27	34.47	3

5.3 Barcoded samples

Readfish can be configured to handle barcodes in two ways. For simple experiments, the user can identify a list of barcodes to be either rejected or accepted. In this way users can exclude or include a subset of barcodes on a sequencing run (Section 4.2.10 and Figure 4.15). For more complex experiments, the user can configure a set of targets for each individual barcode in a library and so sequence specific regions from each (Section 4.2.10 and Figure 4.16). There is no requirement for each sample to be from the same organism and so readfish can target multiple references in a single genomic index.

To test this approach, we used three previously described cell lines: GM12878, from the Utah/CEPH pedigree (Jain et al., 2018a); NB4, a cell line carrying a fusion between PML and RARA representing an acute promyelocytic leukemia (APL) (Mozziconacci et al., 2002); and 22Rv1, a prostate cancer derived cell line containing significant chromosomal abnormalities (Liu et al., 2010). Each sample used a specific panel of gene targets based on known variation (Table 5.4). GM12878 used the TruSight 170 Tumor panel (Na et al., 2019). The NB4 cell line used TruSight RNA Fusion Panel (Siegfried et al., 2018) as it contains an APL fusion. Finally, 22Rv1 being a prostate cancer line we used the previously described COSMIC panel (Tate et al., 2018).

Table 5.4: Run metric performance per barcode and over the entire flow cell.

Barcode	Sample	Panel	Gene Number
01	GM12878	TruSight 170 Tumor Panel	170
02	NB4	TruSight RNA Fusion Panel	508
03	22Rv1	COSMIC	717

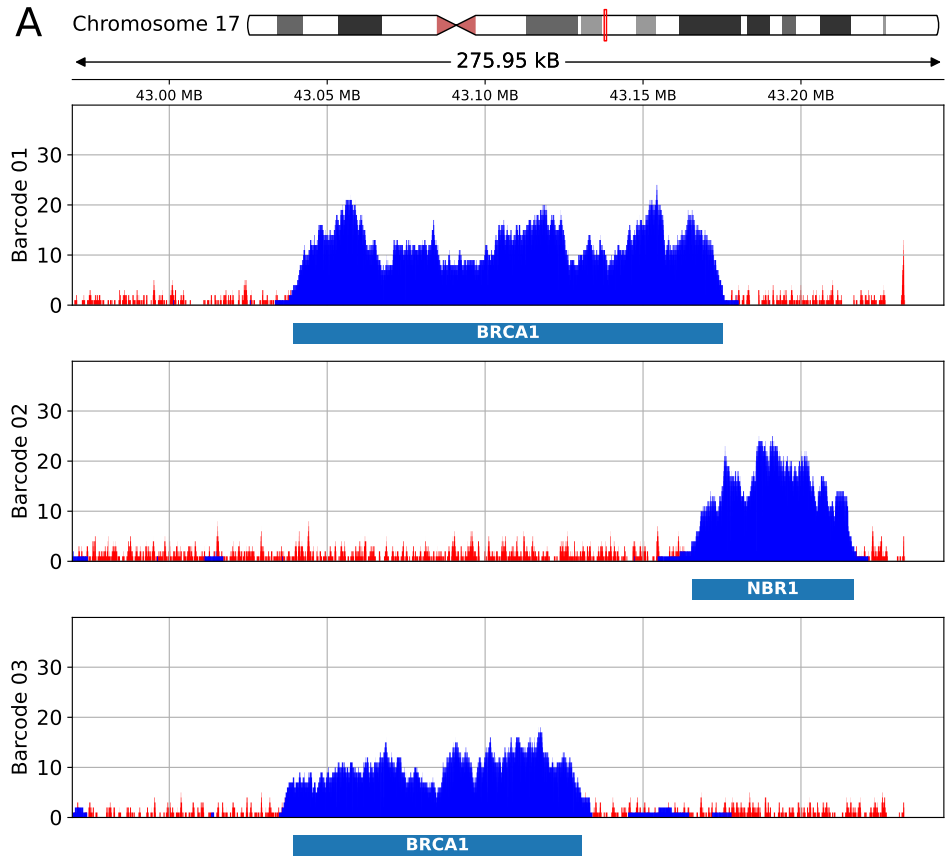
Samples were barcoded and sequenced on a single flow cell, and run for 72 h, including a nuclease flush and reload every 24 h. In a single experiment using a flow cell with 1,330 pores, 18.1 Gb of data were generated, with a total of 15.0 Gb being barcoded successfully (Table 5.5).

Across the whole experiment, the on target reads had an N50 of ~7 kb, with the rejected read N50 being 579 b, or approximately 1.3 s of sequencing which is fully in-line with the observed classification time in non barcoded samples (Figure 4.8b). This results in mean read coverage on target regions of between 11–15 ×. Inspection of individual targets including BRCA1, NBR1, PML and RARA demonstrates

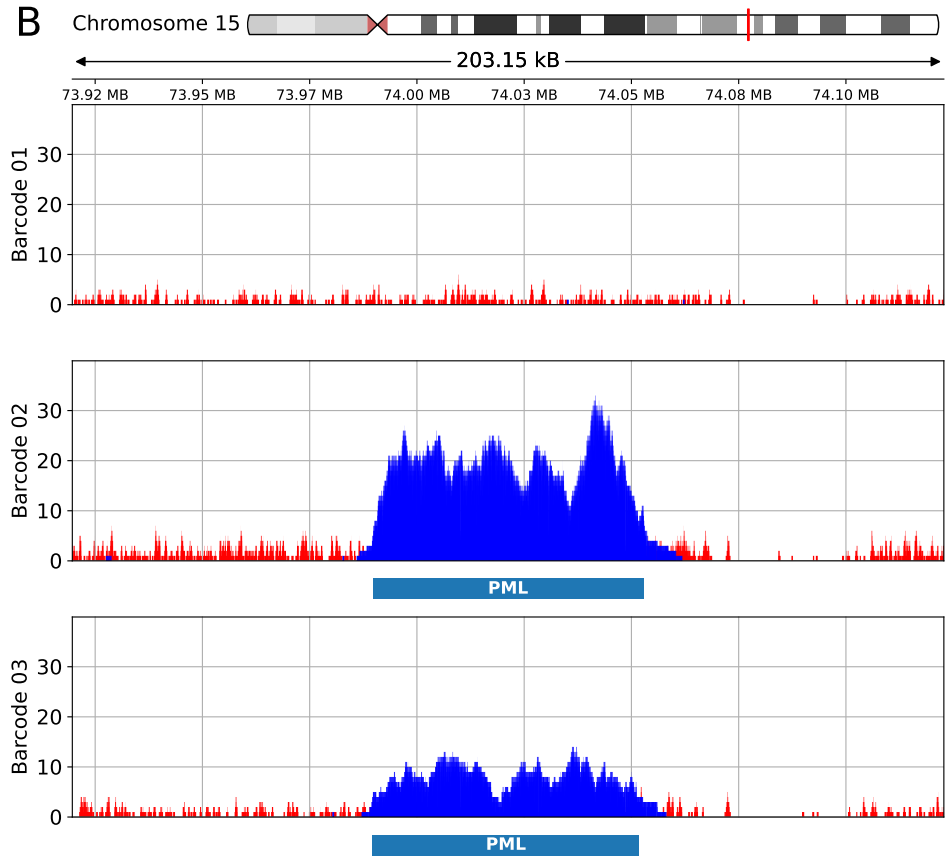
the ability to specifically target unique regions on each sample (Figure 5.3). Current best practice for variant calling requires higher minimal depth than we achieve when looking at three samples. However, long range structural variants can be measured and so we used cuteSV (Jiang et al., 2020) to analyse these three samples. As expected, multiple reads supporting the detection of a fusion between PML and RARA were detected in the NB4 cell line (Figure 5.3). In contrast, this rearrangement was not found in the 22Rv1 line and it cannot be excluded in GM12878 as neither PML nor RARA were within the gene panel used for this cell line (Figures 5.2 and 5.3).

Table 5.5: Run metric performance per barcode and over the entire flow cell.

Barcode	Sample	Yield (Gb)	On Target (Gb)	On Target N50	On Target Mean	Off Target Mean	Mean Target Coverage
01	GM12878	3.80	0.355	8,149	1,926	554	11.0
02	NB4	6.10	1.240	7,191	4,203	551	15.0
03	22Rv1	5.10	1.250	6,858	5,065	556	11.5
Unclassified		3.10				736	
Total		18.79			3,221	587	

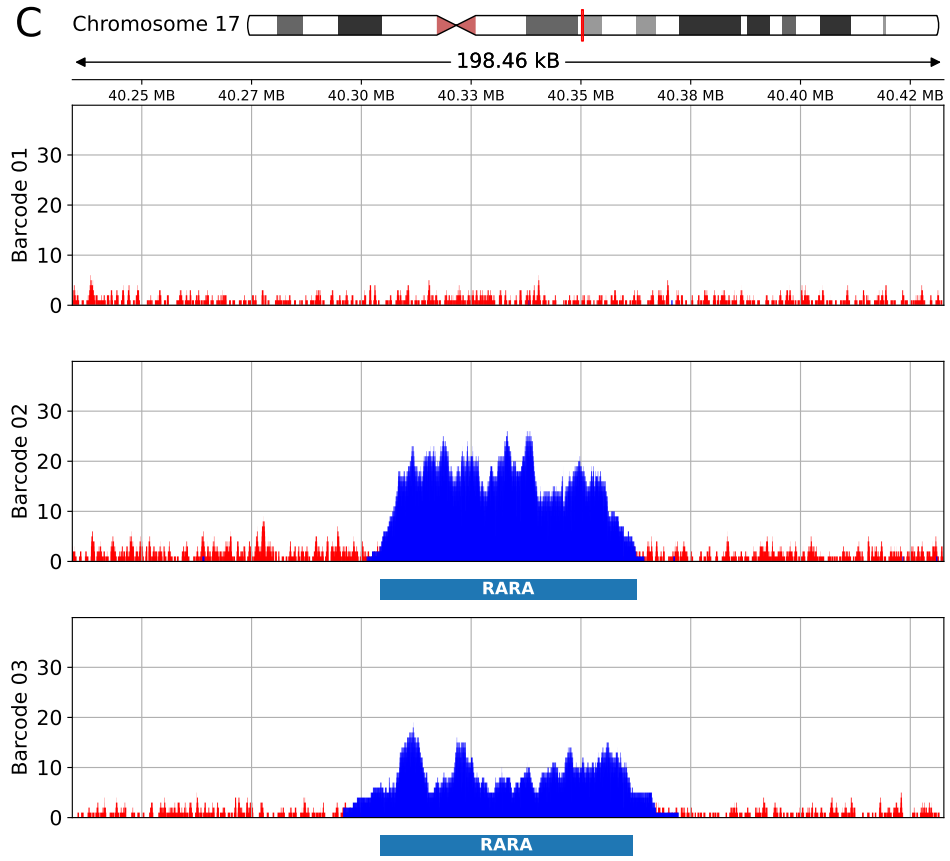


(a) BRCA1



(b) PML

Figure 5.2: Continued on the following page.



(c) RARA

Figure 5.2: Illustration of coverage over each barcoded sample for each target in the panel. Blue is sequenced read coverage, red illustrates coverage of rejected reads. (a) shows coverage over BRCA1 and the adjacent gene NBR1. BRCA1 was a target for barcode 1 and 3, but not 2. The targeted regions are illustrated below the coverage plots. Note that the region representing BRCA1 differs in barcode 1 and 3 by design. NBR1 was only targeted on barcode 2. (b, c) illustrate coverage over PML and RARA respectively, which were only targeted on barcodes 2 and 3.

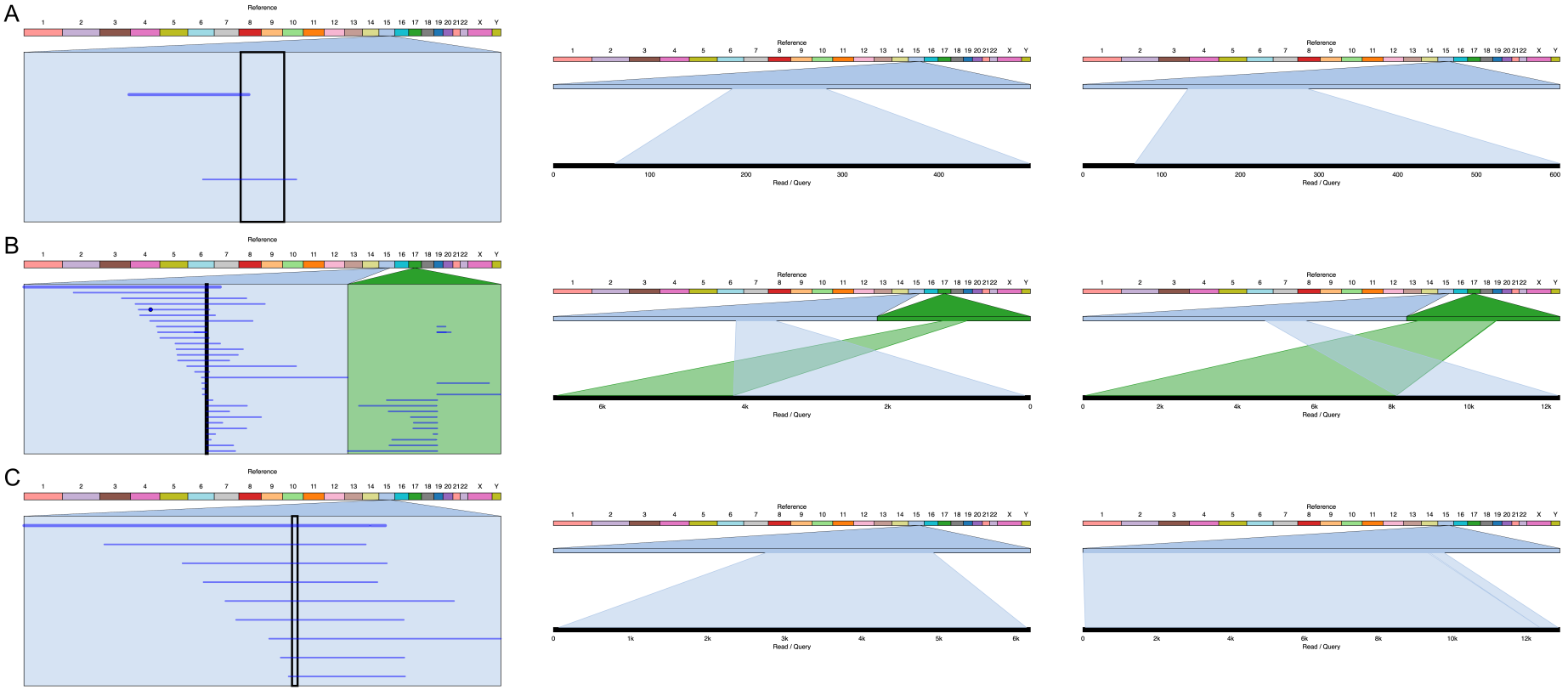


Figure 5.3: Using Ribbon, we visualise reads covering PML (chromosome 15) and any known fusions. (a) Barcode 01, GM12878, has only two reads in the candidate region as PML is not included within the targets for this sample. (b) Barcode 02, NB4, shows multiple reads spanning PML and linking to RARA (chromosome 17) as expected for this fusion cell line. (c) Barcode 03, 22Rv1, also had PML within the target gene list, but had no structural variant in this region as expected. SVs were identified using CuteSV

5.4 Adaptive sampling or "Run Until"

5.4.1 Iterative Alignment

A typical metagenomic scenario involves the amplification of all DNA from a sample. This is usually achieved using 16S amplicon sequencing which cannot achieve the same level of taxonomic assignment as full-length sequencing (Johnson et al., 2019). However, using full-length sequencing there are still questions about the appropriate read depth that is needed to answer a particular question. There have been attempts to calculate a suitable read depth (Ni et al., 2013); however, what must be sequenced to answer different questions may vary considerably. For example, the identification of sample composition will require broad coverage over as much of the sample as possible, whereas detecting specific single nucleotide polymorphisms in specific genes will require concentrated coverage over the genes of interest.

To simulate metagenomics questions, we utilised the ZymoBIOMICS high-molecular-weight DNA standard (D6322). As this sample consists of high-quality extracted DNA with a mean read length ≥24 kb it will de facto improve sequencing and subsequent analysis. This standard mixture can be used to benchmark the performance of sequencing approaches for microbiomics and metagenomics analysis. The theoretical composition of this sample is seven bacterial species, each at 14% (*Pseudomonas aeruginosa*, *Escherichia coli*,, *Salmonella enterica*, *Enterococcus faecalis*, *Staphylococcus aureus*, *Listeria monocytogenes*, and *Bacillus subtilis*) and a single fungal species at 2% (*Saccharomyces cerevisiae*).

In a similar metagenomics benchmarking experiment, Nicholls et al. generated a reference dataset using the similar ZymoBIOMICS Microbial Community Standards. This DNA standard included two extra species, a bacteria *Lactobacillus fermentum* and a yeast *Cryptococcus neoformans*; all bacterial species were present at 12% while both yeast species were present at 2%. The data generated using the even community, sequencing on GridION, enabled Nicholls et al. to create *de novo* assemblies of the bacterial species. However, neither of the eukaryotic genomes could not be reconstructed reliably as they were present at too low abundance. This resulted in the coverage depth for *Saccharomyces cerevisiae* being 17× and *Cryptococcus neoformans* being 10×.

For this reason, using a mock metagenomic community is an ideal experiment as it allows for the simulation of depleting host genomic material that is highly abundant (~98%). In addition to "depleting" host material, this programme aims

for each target to meet a coverage threshold, such as 40×. The readfish align programme (Figure 5.4) watches for base called data from completed reads, aligning them to the reference used for readfish targets and calculates read depth over all contigs present in the reference. Once a user-defined threshold is met, readfish targets' experiment criteria is updated with the contigs that have sufficient coverage depth and they are depleted.

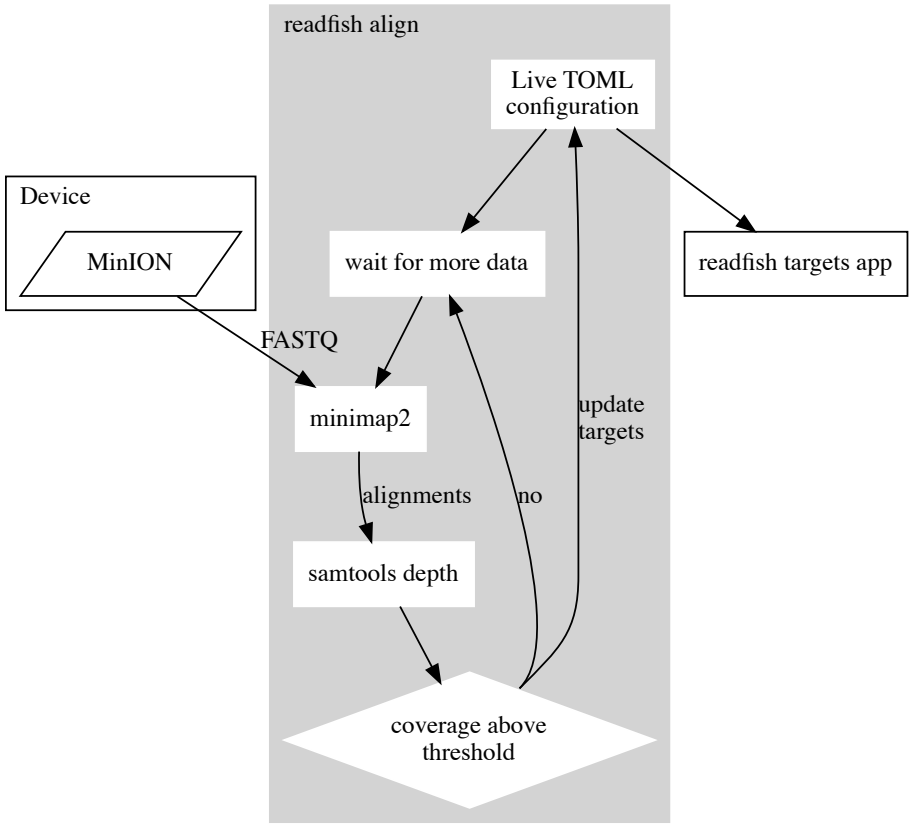


Figure 5.4: Flow diagram of iterative alignment programme. As completed reads are base called by MinKNOW they are written to disk. Readfish iteralign polls the output folder for the current run. When FASTQ files are written, they are aligned and coverage depth calculated using the defined reference in the readfish TOML file. Chromosome targets that reach or surpass a defined level are then added as targets for depletion, which are picked up by the readfish targets app and effected.

In addition to dynamically depleting species that have reached 40× readfish align implements a "Run Until" condition. That is, once all targets in the reference genome are being depleted (all targets have reached the coverage goal) the sequencing run can be stopped. This can be seen in Figure 5.5, mean read length reduces as the cov-

erage target is reached. Plotting coverage over time for reads not rejected by readfish (middle column) shows a decrease in coverage accumulation for completed genomes with an increase in sequencing potential for the least abundant sample, *S. cerevisiae* (Figure 5.5).

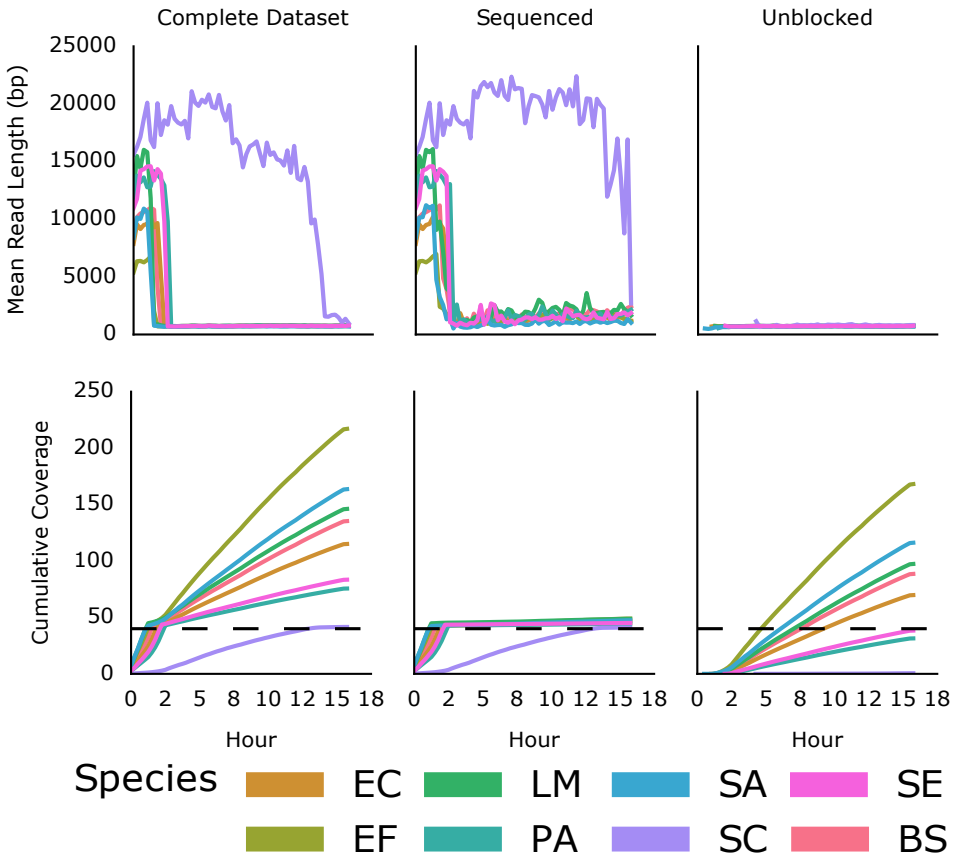


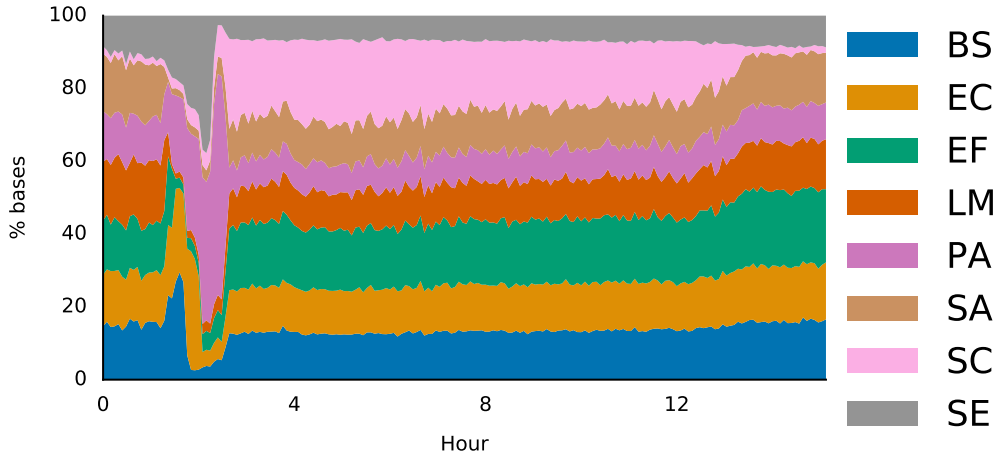
Figure 5.5: Mean read length and cumulative coverage of readfish align. Each row is split into all data, sequenced reads, and unblocked reads. Top row is mean read length, below is cumulative coverage. As species reach the coverage threshold of 40× the mean read length reduces as these targets are now being depleted. This frees sequencing capacity for the low abundance *S. cerevisiae* targets. This is clearly visible as the rate at which coverage accumulates for *S. cerevisiae* increases at ~2 h in the sequenced subset.

The proportion of bases mapping to each constituent genome in the sample shows how sequencing capacity increases for *S. cerevisiae* as other targets are depleted (Figure 5.6a). This gradually tapers back to roughly the proportions of each genome in the sample as the sequencing run concludes (Figure 5.6a). Notably, however, relative abundance of the constituent species can still be determined by observing the proportion of reads aligning to each genome in the sample (Figure 5.6b) as these remain consistent throughout the duration of the experiment. The run automati-

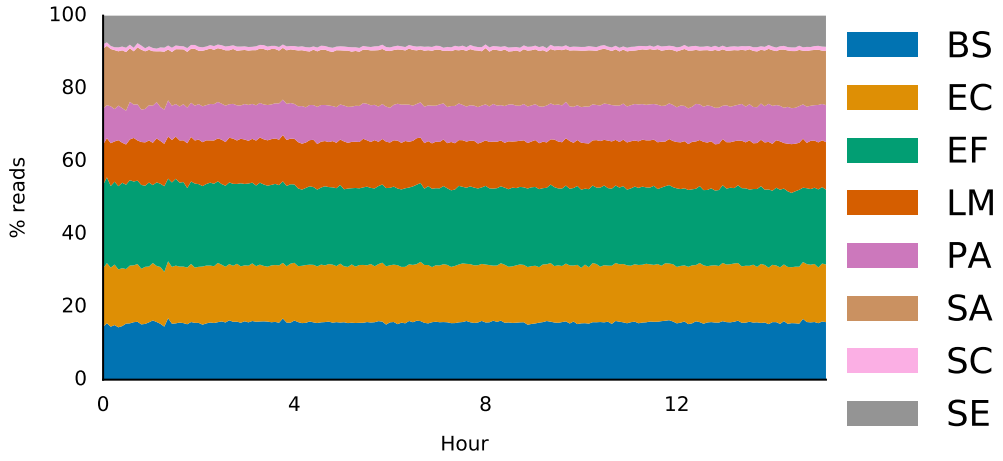
cally stops once each genome reaches 40×, taking ~16 h and 4.4 Gb of sequence data (Table 5.6 and Figure B.2).

Table 5.6: Iteralign NanoStat Summary — This run was conducted before the `unblocked_read_ids.txt` file was collected, therefore summary stats are only available for the entire run.

	Complete run
Active channels	504.0
Mean read length	1,247.1
Mean read quality	12.0
Median read length	675.0
Median read quality	12.6
Number of reads	3,540,936.0
Read length N50	1,544.0
Total bases	4,415,735,206.0



(a)



(b)

Figure 5.6: (a) Stacked area plot, as individual species reach 40× coverage they are rejected. As such, the proportion of bases mapping to each species changes over time. This is particularly evident in the SC band as it starts out as at ~2% of the sample, but just after 2 hours the proportion greatly increases as sequencing capacity is redirected to only this organism. (b) Conversely, the proportion of reads mapping to each species over time does not change during the run. Species: BS, *Bacillus subtilis*; EF, *Enterococcus faecalis*; EC, *Escherichia coli*; LM, *Listeria monocytogenes*; PA, *Pseudomonas aeruginosa*; SC, *Saccharomyces cerevisiae*; SE, *Salmonella enterica*; SA, *Staphylococcus aureus*.

5.4.2 Iterative Centrifuge

The approach used with readfish align assumes total knowledge of the sample composition *a priori*, as such it is impractical. Instead, by integrating a metagenomic classifier we can simulate no prior knowledge.

In this programme, readfish centrifuge, a broad centrifuge (Kim et al., 2016) index is used to classify completed reads. These classifications are accumulated and tracked, when a single classification has been made more than 2,000 times the corresponding RefSeq genome is dynamically retreived from NCBI; then an index for minimap2 is generated and passed to readfish targets to carry out selective sequencing. This process iterates until a coverage threshold is achieved (Figure 5.7).

Using this method, we generated 5.99 Gb of sequence data, identifying all bacterial genomes in the sample. Although we observed enrichment, readfish centrifuge struggled to keep up with data generation (Figure B.3), likely due to the intensive background classification process. In addition, the flow cell became completely blocked after 24 h before reaching 40× on the final species, *S. cerevisiae* (Figure 5.8).

This is due to the entirety of a read being considered for selection rather than just the first few chunks. As a result reads were sequenced when they should have been unblocked and some reads were unblocked too late, potentially damaging the flow cell surface (Figure 5.8; top center and right). This can also be seen in the mean read length of unblocked reads (Table 5.7). This experiment was completed within 24 h, illustrating the benefits in terms of time-to-answer. As expected, improved coverage depth results in almost complete assemblies using MetaFlye (Figure 5.10), this is in part due to improved read lengths compared with Nicholls et al..

Table 5.7: Itercent NanoStat Summary

	Sequenced	Unblocked	Complete run
Active channels:	464.0	451.0	467.0
Mean read length:	7,707.1	1,005.3	2,160.6
Mean read quality:	10.2	11.3	11.1
Median read length:	2,548.0	869.0	905.0
Median read quality:	11.2	11.6	11.5
Number of reads:	478,349.0	2,296,491.0	2,774,840.0
Read length N50:	22,704.0	1,079.0	8,268.0
Total bases:	3,686,701,649.0	2,308,737,840.0	5,995,439,489.0

Similarly to readfish align, readfish centrifuge shows the same behaviour with proportion of reads being consistent with sample composition throughout the entire duration of the run Figure 5.9b. However, the proportion of bases starts out as

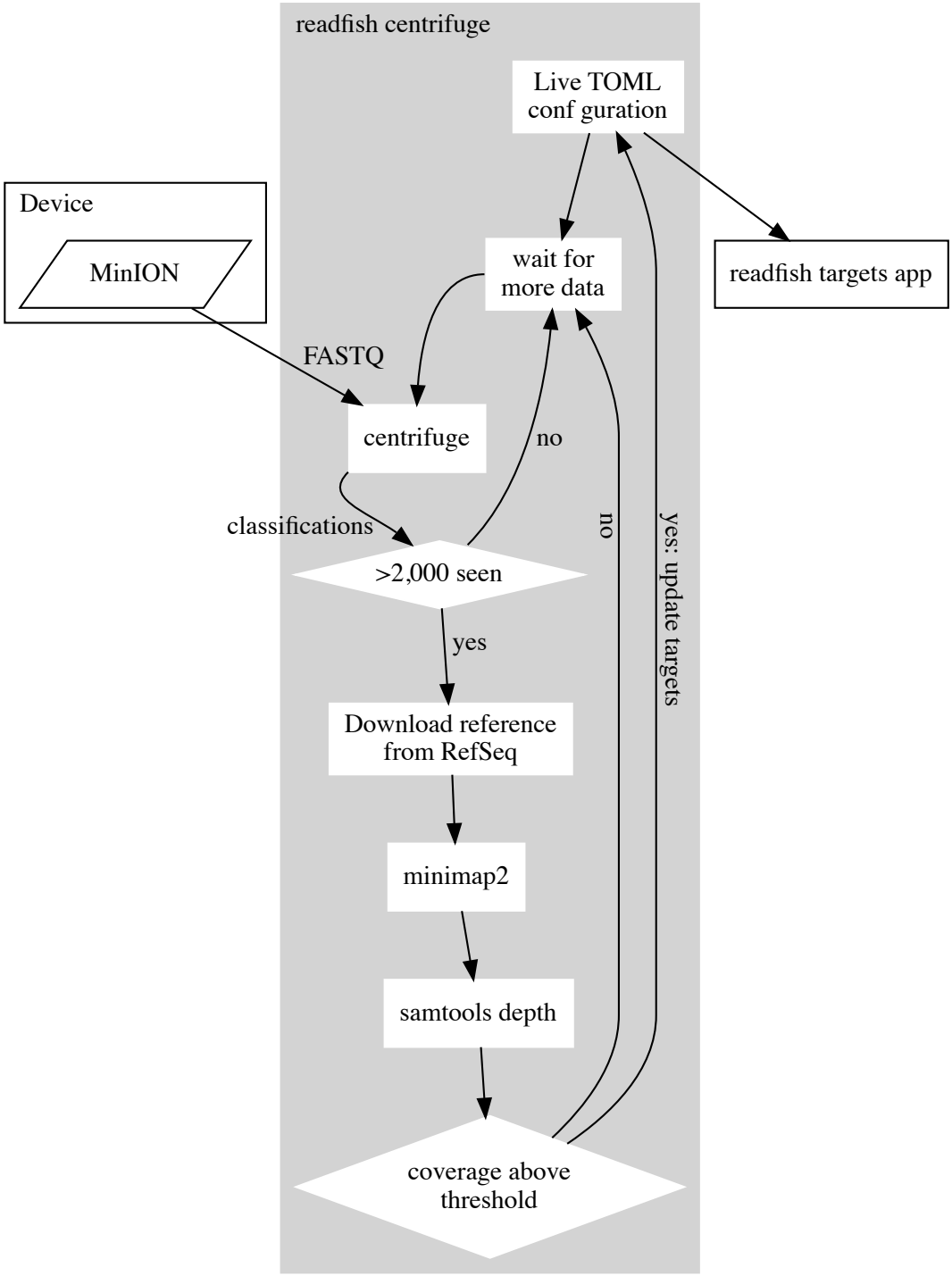


Figure 5.7: Flow diagram of iterative centrifuge programme. As completed reads are base called by MinKNOW they are written to an output folder. Readfish centrifuge polls the specific folder for the run that it is monitoring. When files are written, they are classified using centrifuge (Kim et al., 2016). Once 2000 reads have been seen for any individual species the reference genome is retrieved from NCBI/RefSeq. Using these downloaded genomes a new reference is generated for readfish to use. All targets in the multi-reference index are monitored for read-depth. Once a threshold has been reached for any given species it is depleted using readfish targets.

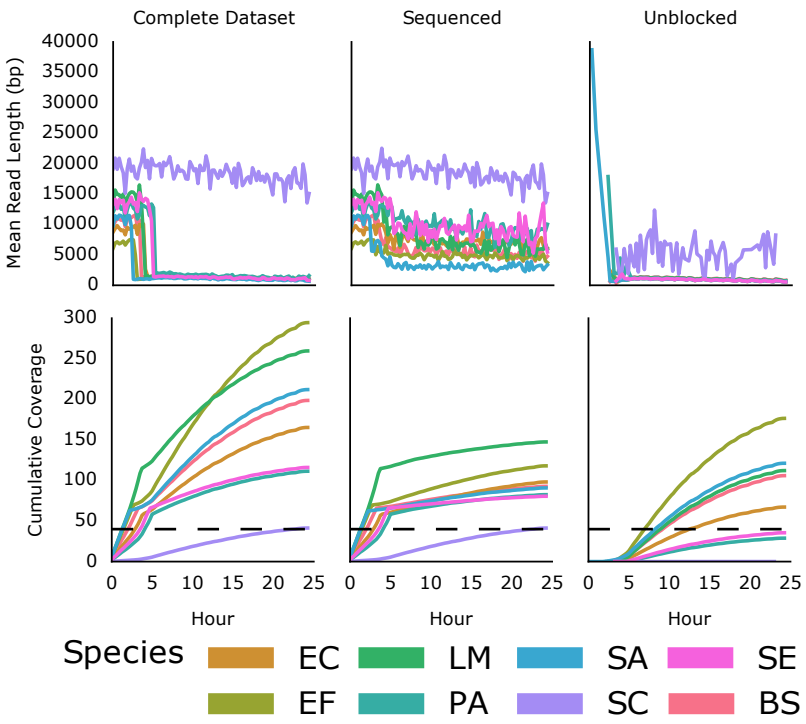
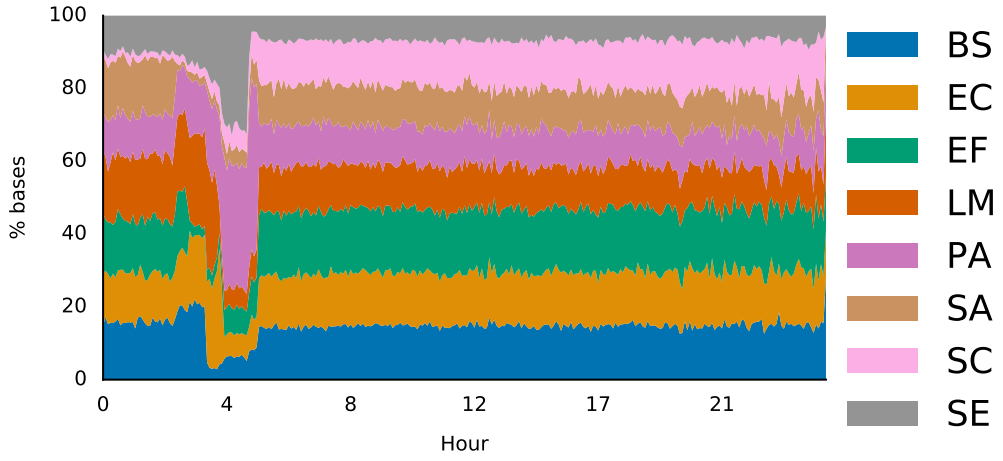
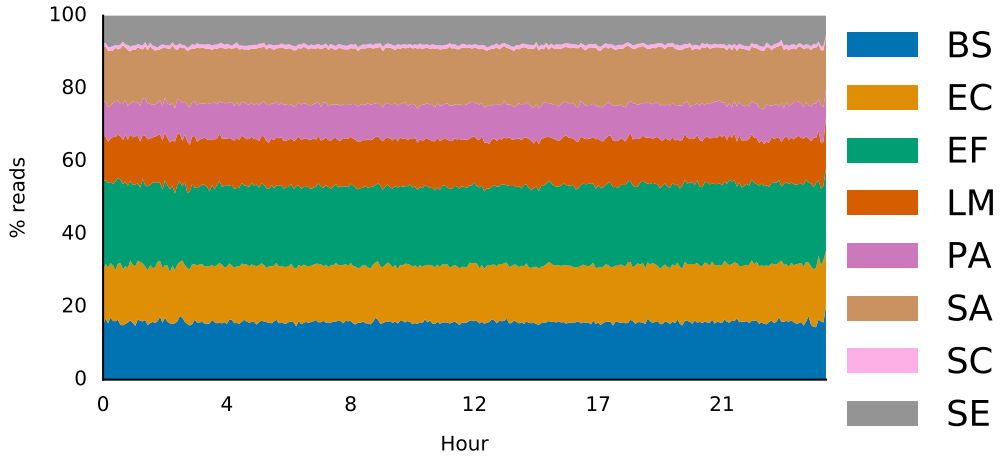


Figure 5.8: Mean read length and cumulative coverage of readfish centrifuge. Each row is split into all data, sequenced, and unblocked reads. Top row is mean read length, below is cumulative coverage. Despite bacterial species reaching the target coverage quickly, they were not effectively depleted. This results in coverage continuing to climb (bottom left) reducing available capacity for low abundance targets.

expected, but does not return to this composition at the end of the run (Figure 5.9a). This is due to the fact that *S. cerevisiae* did not reach the target threshold and was still being enriched for when the flow cell stopped sequencing at 24 h.



(a)



(b)

Figure 5.9: (a) Stacked area plot, as individual species reach 40× coverage they are rejected. As such, the proportion of bases mapping to each species changes over time. This is particularly evident in the SC band as it starts out as at ~2% of the sample, but just after 2 hours the proportion greatly increases as sequencing capacity is redirected to only this organism. (b) Proportion of reads mapping to each species over time does not change during the run. Species: BS, *Bacillus subtilis*; EF, *Enterococcus faecalis*; EC, *Escherichia coli*; LM, *Listeria monocytogenes*; PA, *Pseudomonas aeruginosa*; SC, *Saccharomyces cerevisiae*; SE, *Salmonella enterica*; SA, *Staphylococcus aureus*.

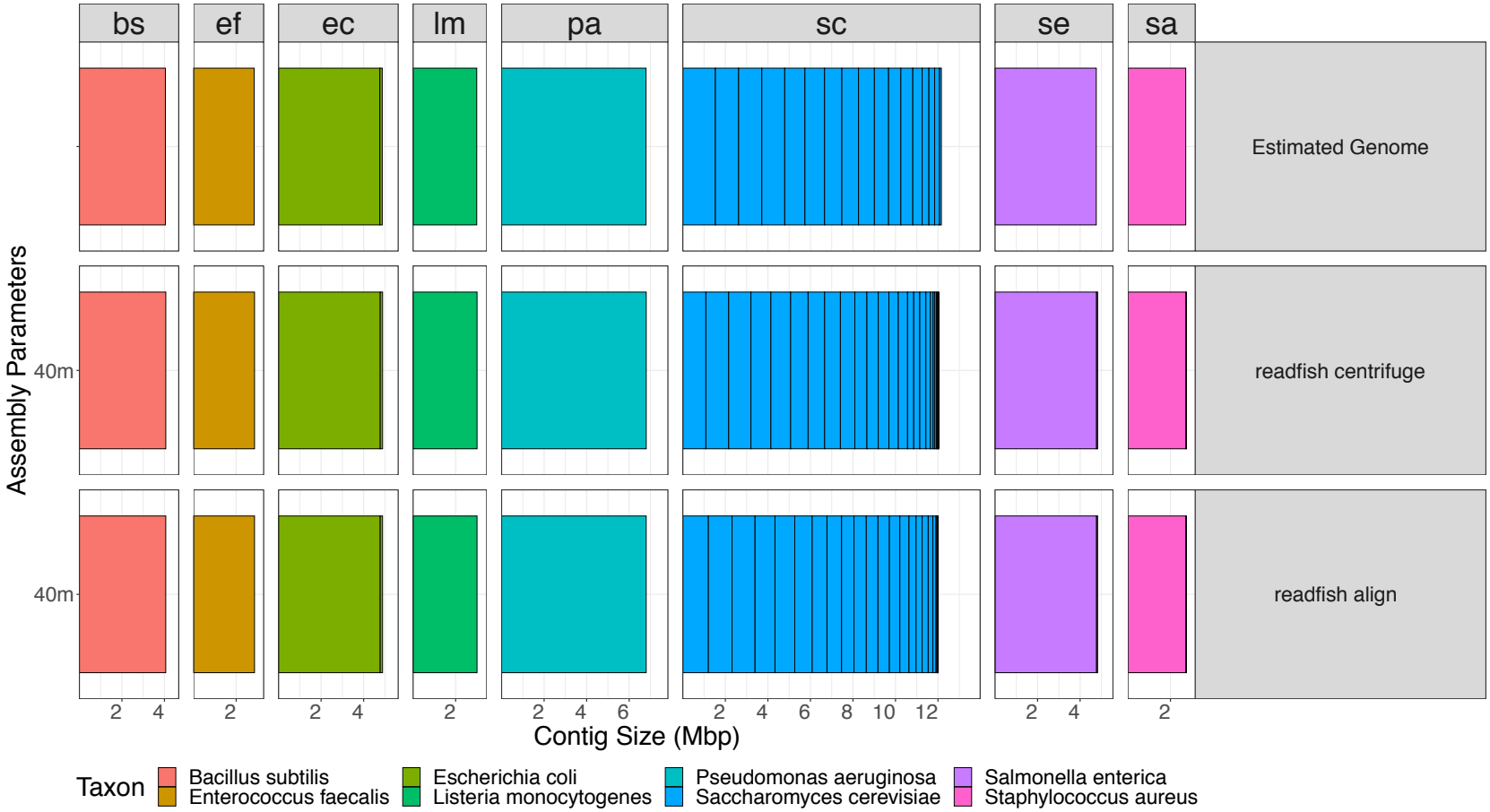


Figure 5.10: Assemblies for ZymoBIOMICS mock microbial community using data from readfish align and readfish centrifuge. Data assembled using MetaFlye using an estimated genome size of 40 Mb. Bacterial genomes are comparable to the estimated genomes for each species. Eukaryotic (SC) is more fragmented than the estimated genome, but is close in overall size.

5.5 Discussion

Real-time selective sequencing is an appropriate approach in situations with well defined targets in a well characterised sample. The primary benefits of software based selective sequencing is easy curation of target databases, as these are extremely flexible allowing custom targets to be generated on a whim. In addition to the flexibility of the panel format, readfish allows target panels to be updated mid-run such that an experiment can react to data in real-time. This allows an external process to inform on selective sequencing without the need to stop and start the analysis, particularly this process can be automated as in readfish align and centrifuge. These automated processes allow sequencing runs to continue until they would be producing too much data for the question at hand. By stopping early, readfish improves on the time to answer freeing up sequencing infrastructure for other experiments. There is also a reduction in the cost of sequencing as a run that may, by shotgun sequencing, have required multiple runs to yield on-target data can now be accomplished using fewer consumables such as flow cells and library preparation reagents. Costs can further be reduced by employing standard efficiency steps such as multiplexing samples by barcoding them. This allows many samples to be sequenced concurrently and with readfish allows arbitrary selection of barcode specific panels.

Real-time selective sequencing is not a magic method. It is an appealing technique but is very sensitive to *good* sequencing libraries. That is, an experiment that has problems with low-yield, low occupancy, not enough computational power, or a short library read length — to name a few — will struggle to acheive any meaningful enrichment of targets.

Chapter 6

Discussion

6.1 Conclusion

Sequencing DNA is a central part of modern molecular biology. Ranging from whole genome sequencing and assembly to just confirming plasmid features during an experiment. Year-on-year the cost of sequencing reduces, enabling the inexpensive production of large volumes of data. This volume of data can quickly become unmanageable as well as difficult to analyse and distribute. Targeted sequencing approaches aim to address these issues either through upfront filtering and amplification, using molecular methods such as CRISPR-Cas9 or PCR; or by real-time selective sequencing on single molecule sequencing platforms, such as the MinION. All these techniques can reduce sequencing costs and achieve high coverage over regions of interest. Though the molecular methods (CRISPR-Cas9 and PCR) have low throughput, high input requirements and may result in loss of extra data such as detecting nucleotide modifications.

The aims of this project were to: increase the scale of useable reference genomes for selective sequencing; and (by extension) increase the number of target regions that could be considered simultaneously; reduce the computational requirements so that a single workstation or laptop is capable of running Read Until; and finally, enable true adaptive sampling allowing real-time feedback throughout an experiment.

Analysis of bulk FAST5 data allows detailed inspection of current traces that would be expected during sequencing. These are the data that must be processed and considered during Read Until. In post-sequencing data the interpretation of each molecule, as determined by MinKNOW, is presented. Though, this is not necessarily the complete reading of that single molecule as there are extra useable data that can be rescued from reads (Payne et al., 2018; Workman et al., 2019).

This deeper view of useable signal allows for tuning real-time processes to enable better and faster classification. Moreover, this extra information can be utilised for better post-sequencing data recovery.

Building on the initial work of Loose et al. and with improvements to the Read Until API, a basecalling and alignment approach for selective sequencing was developed. This initially utilised open source CPU base calling in the form of Scrappie, but took advantage of the GPU accelerated Guppy basecaller. As Guppy requires GPU acceleration for efficient basecalling another CPU basecaller, DeepNano-Blitz, was also integrated. The Guppy and DeepNano-Blitz basecallers were both performant enough to allow for real enrichment of thousands of target loci in the human genome. Guppy requires the use of a GPU, so has higher computational requirements, but these are modest compared to the multi-core servers that other selective sequencing tools require (Loose et al., 2016; Kovaka et al., 2020); with readfish only requiring a workstation computer for sequencing that meets ONT's recommendations, such as the GridION (Table 2.1). Such a relatively small computational footprint makes this approach quite practical in most sequencing labs.

Having a system that works at the scale of gigabase-sized references and *many* target regions increases the practicality of this targeted sequencing approach. Evaluating this system with hypothetical real-world scenarios help ensure that readfish is a feasible solution. For example, applying different gene panels in the form of the COSMIC panel (Tate et al., 2018) and TruSight 170 Tumor (Na et al., 2019) and TruSight RNA Fusion panels (Siegfried et al., 2018) These panels have been applied to both single experiments addressing different operating systems (Linux, MacOS, and Windows) and hardware configurations (with and without GPU) as well as to different experimental configurations such as the inclusion of multiplexed (barcoded) samples on a single flow cell. These experiments demonstrate that target loci are sequenced at a greater depth than the rest of the (off-target) genome, therefore enriching these samples.

Using the ZymoBIOMICS mock microbial community allowed for the assessment of readfish to distinguish and select for microbial genomes from a mixed background. These experiments also applied truly adaptive sampling techniques to assess genome coverage in real-time. These adaptive examples evaluate genome coverage using either *a priori* knowledge or real-time classifications of the sample composition. In the first case minimap2 (Li, 2018) is used alongside the official ZymoBIOMICS reference sequence. In the second case completed reads are evaluated using a broad centrifuge classification index and dynamic retrieval of reference genomes from NCBI/refseq, which are then used to create a reference for selective sequencing. This approach aims to gradually deplete all samples from the library focusing available pores on under-represented sequences. Finally, these adaptive

approaches reduced the time to sequence each constituent genome to a target depth of ~50×.

6.2 Current uses of readfish

Readfish has been open source and available on GitHub since February 2020. Since then the Read Until API improvements and underlying basecalling and alignment approach have been incorporated into MinKNOW. The version of Read Until that MinKNOW implements[1] is a subset of the features that readfish has such as, enrichment or depletion of specific target regions of a reference or enrichment or depletion of entire references.

Readfish provides more fine-grain control over the selective sequencing logic. This allows for greater customisation of how each molecule seen will be considered. For example, readfish allows setting both a minimum and maximum number of times a single molecule should be evaluated before being selected or rejected. In addition, there are rules regarding specific cases, such as multiple alignment, are handled.

Papers that use readfish or ONT adaptive sampling broadly fall into two categories: human diagnostic and metagenomic communities. Here I will consider publications that have used readfish or ONT adaptive sampling for targeted enrichment.

Targeted long-read sequencing identifies missing disease-causing variation (Miller et al., 2021) aims to increase genetic diagnosis of patients using targeted long read sequencing. Here the targeted sequencing aims to replace multiple other steps in the process include microarray and whole exome sequencing, saving both time and costs. Miller et al. intend to address complex copy number variant changes, specifically multiple deletions or duplications on one or more chromosomes. Target panels consisted of clinically relevant genes with flanking sequence of up to 100 kb up- and down-stream added. In addition, other regions of non-target chromosomes were enriched to serve as internal copy number and coverage controls. These panels routinely yielded 7–40 × coverage (Miller et al., 2021).

Rapid-CNS^2: Rapid comprehensive adaptive nanopore-sequencing of CNS tumors, a proof of concept study (Patel et al., 2021) aims for a comprehensive mutational, methylation, and copy number profiling of central nervous system tumours with a single, cost effective sequencing assay. This assay selects for a panel of brain-tumor related regions and CpG sites. They found complete concordance with the

[1] Called "ONT Adaptive Sampling"

EPIC array[2] for copy number profiles and consistent classification for *MGMT* promoter status and methylation. This pipeline has a complete turnaround time of ~5 d, with a long-term goal of integrating adaptive nanopore sequencers into hospitals and care locations for faster diagnosis.

Comprehensive genetic diagnosis of tandem repeat expansion disorders with programmable targeted nanopore sequencing (Stevanovski et al., 2021) used targeted long read sequencing in combination with high-depth PromethION shotgun sequencing. They sampled ~1.6 % of the human genome for Short Tandem Repeats (STRs) and other clinically relevant regions for 27 individuals. Similarly to Miller et al., a ~4.5 × increase in sequencing depth was observed for target regions yielding 7–32 × median coverage.

All these applications, so far, have been using readfish directly and are only considering human genomics. The human reference genome is extensively studied and very well characterised. This level of detail allowed these approaches to rapidly progress from designing a panel of targets to selective sequencing.

With metagenomic communities, these samples are typically a mock community (Martin et al., 2022) or a clinical sample with a host (human) background (Marquet et al., 2021; Zhao et al., 2021). Reducing the abundance of these host sequences is difficult due to short read fragments, mixed samples, and (potentially) lower basecall accuracy. For example, Marquet et al. found that they could reduce human background from 87.9% to 34.7% by depleting human-aligning sequences when using readfish. This reduction can go even further when enriching for a subset of the population (87.9% to 8.3%) but this also rejects 96% of all reads (Marquet et al., 2021). Similarly, Martin et al. saw 40% of on-target reads getting rejected incorrectly when using ONT adaptive sampling.

6.3 Other approaches to selective sequencing

6.3.1 Mapping raw signal

UNCALLED is one such technique, which aims to map raw signal without basecalling (Kovaka et al., 2020). It builds an FM-index from a reference genome and converts the raw signal into "events" that represent *k*mers based on the pore/*k*mer model from ONT. The high-probability *k*mers are used to query the FM-index, each successive query refines the mapping location until there is one location that is significantly better than the others. UNCALLED is able to work with small genomes, such as the ZymoBIOMICS mixed community, without issue. However, UNCALLED

[2] https://emea.illumina.com/products/by-type/microarray-kits/infinium-methylation-epic.html

cannot make use of larger genomes without subsetting to the target region and intensive repeat masking.

Another tool is Sigmap that aims to implement a signal streaming method, similar to UNCALLED (Zhang et al., 2021). Unlike UNCALLED, Sigmap converts the reference genome into a simulated squiggle that is indexed using an optimised k-d tree data structure. This signal reference is queried and used in the same fashion as minimap2, using specific "seed" matches that are chained together. Sigmap is benchmarked against UNCALLED, showing a 4.4 × speedup when mapping yeast sequences. However, Sigmap is not presently implemented in any real-time selective sequencing applications and generates simulated references that are ~26–35 × larger than their corresponding FASTA reference. Finally, Sigmap requires a large amount of external[3] computational resources that greatly reduces it's portability.

6.3.2 Bloom filter

A recent tool, ReadBouncer, implements a bloom filter that acts on basecalled data rather than using an aligner (Ulrich et al., 2022). Like readfish, ReadBouncer uses exactly the same basecalling routines (both Guppy and DeepNano-Blitz) for decoding raw signal. However, instead of using minimap2 for read alignment a bloom filter is used.

Briefly, a bloom filter makes use of *k*mer hashing (similar to minimap2's seeding step) but forgoes chain extension between seeds. An index for the bloom filter is created by using successive hash functions on the unique *k*mers in the target sequence. Therefore, ReadBouncer can only yield two possible results: "not in the target set" and "possibly in the target set". No mapping location is given. In this regard, ReadBouncer and bloom filter approaches maybe appropriate for host depletion or binary classification experiments, but this depends on the hash functions chosen.

6.3.3 Other approaches

A recent, SARS-CoV-2 inspired, approach (SquiggleFilter) created a hardware accelerated DTW implementation for selectively sequencing SARS-CoV-2 only (Dunn et al., 2021). This technique acts as a binary filter that only targets viral reads from SARS-CoV-2. Similarly, a neural network based binary filter (SquiggleNet) attempts to separate human from bacterial reads (Bao et al., 2021). This requires the specific training of a model for the purpose of selective sequencing.

[3] from the sequencing workstation

6.4 Future directions

Readfish is functional and actively maintained, but requires some proactive improvements as newer dependencies become available. For example, upgraded versions of Guppy[4] introduced API changes that broke the basecalling step when using Guppy. Likewise, the classification step, using minimap2 (Li, 2018), works extremely efficiently even with large (gigabase-sized) genomes. However, when assessing a mixed sample using minimap2 can become more risky as the chance of false positives when assessing small read chunks can increase (Martin et al., 2022; Marquet et al., 2021). In these scenarios, particularly when attempting to deplete a host or background, a broad classifier such as those used by centrifuge (Kim et al., 2016) or kraken (Wood et al., 2019) would be more appropriate and allow for a many more samples to be included.

6.4.1 Adaptive sampling

There needs to be an expansion of truly adaptive sampling workflows. These are ones where meaningful analysis is done iteratively as the experiment generates output and this analysis then re-informs the sequencer with updated targets. This has been attempted already for smoothing coverage over a genome and assigning targets where there is the greatest benefit (e.g. low coverage areas) (Maio et al., 2020). Other sequencing schemes should be considered or explored especially for tasks like *de novo* assembly, for example generating a high-coverage dataset using a large single molecule platform such as a PacBio Sequel II or an ONT PromethION. This large dataset can then be assembled and specific gaps in the assembly targeted using a single MinION flow cell.

6.4.2 Copy number variation

While nanopore sequencing experiments typically aim for extending read length, copy number variation (CNV) can be accessed by sequencing many (millions) of short reads (Baslan et al., 2021). This was achieved by optimising the sequencing library for short reads. Alternatively, this can be done by using the selective sequencing features of nanopore sequencers; unblocked (off-target) reads are very short (typically <1 kb). These short reads will still align to a reference genome and can be binned to approximate copy number across the genome. This technique is likely going to be a useful complement to methods such as cytogenetic testing and karyotyping.

[4] https://community.nanoporetech.com/posts/guppy-v6-0-0-release

6.4.3 Barcode balancing

Real-time selective sequencing lets the user of a nanopore device control and modify the result of a sequencing experiment by applying arbitrary rules. The outcome of this control is, in part, determined by the initial library loaded on the flow cell. That is, a library with a low concentration of DNA molecules will have lower throughput compared with a higher concentration library. And, as previously mentioned, readfish and Read Until are not "magic methods" they work within the constraints of the sequencing experiment being carried out.

With that in mind, multiplexed samples add an extra dimension of complexity as the overall library composition will affect overall sequencing efficiency while individual samples (barcodes) in the library will be present at their own concentration. Barcode balancing aims to normalise the amount of data that each sub-sample in a barcoded library produces, whether it is read number or overall yield. This makes this system extremely sensitive to the unique composition of each barcoded library. For these reasons, selective sequencing of barcoded samples is not a trivial problem unless the overall goal is to filter barcodes once they reach set thresholds.

6.5 Closing remarks

So should I use readfish or Read Until? Yes, no, maybe...[5] It really depends on what your end goal is with this sequencing. For whole exome sequencing without target capture or controlling coverage over an entire genome, readfish could work exceptionally well (Payne et al., 2020, 2021; Miller et al., 2021; Patel et al., 2021; Stevanovski et al., 2021). For samples that are not so highly studied there are methods in development to dynamically target areas of most interest (Maio et al., 2020). Though, selective sequencing should not be blindly applied without some care and thought into the desired result.

This dissertation has presented a new implementation of real-time selective sequencing, readfish. Hopefully, the work here makes selective sequencing practical and accessible.

[5] https://youtu.be/O8SMmG6sW9k

Bibliography

Adessi, C. (2000). Solid phase DNA amplification: characterisation of primer attachment and amplification mechanisms. *Nucleic Acids Research* 28(20): 87e–87. doi:10.1093/nar/28.20.e87.

Al-Naymat, G., Chawla, S., and Taheri, J. (2012). Sparsedtw: A novel approach to speed up dynamic time warping. *CoRR* abs/1201.2969.

Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., and Walter, P. (2017). *Molecular Biology of the Cell*. W.W. Norton & Company. doi:10.1201/9781315735368.

Alkan, C., Sajjadian, S., and Eichler, E. E. (2010). Limitations of next-generation genome sequence assembly. *Nature Methods* 8(1): 61–65. doi:10.1038/nmeth.1527.

Athanasopoulou, K., Boti, M. A., Adamopoulos, P. G., Skourou, P. C., and Scorilas, A. (2021). Third-generation sequencing: The spearhead towards the radical transformation of modern genomics. *Life* 12(1): 30. doi:10.3390/life12010030.

Avery, O. T., MacLeod, C. M., and McCarty, M. (1944). STUDIES ON THE CHEMICAL NATURE OF THE SUBSTANCE INDUCING TRANSFORMATION OF PNEUMOCOCCAL TYPES. *Journal of Experimental Medicine* 79(2): 137–158. doi:10.1084/jem.79.2.137.

Backman, J. D., Li, A. H., Marcketta, A., Sun, D., Mbatchou, J., Kessler, M. D., Benner, C., Liu, D., Locke, A. E., Balasubramanian, S., Yadav, A., Banerjee, N., Gillies, C. E., Damask, A., Liu, S., Bai, X., Hawes, A., Maxwell, E., Gurski, L., Watanabe, K., Kosmicki, J. A., Rajagopal, V., Mighty, J., Jones, M., Mitnaul, L., Stahl, E., Coppola, G., Jorgenson, E., Habegger, L., Salerno, W. J., Shuldiner, A. R., Lotta, L. A., Overton, J. D., Cantor, M. N., Reid, J. G., Yancopoulos, G., Kang, H. M., Marchini, J., Baras, A., Abecasis, G. R., Ferreira, M. A. R., and and (2021). Exome sequencing and analysis of 454, 787 UK biobank participants. *Nature* 599(7886): 628–634. doi:10.1038/s41586-021-04103-z.

Bao, Y., Wadden, J., Erb-Downward, J. R., Ranjan, P., Zhou, W., McDonald, T. L., Mills, R. E., Boyle, A. P., Dickson, R. P., Blaauw, D., and Welch, J. D. (2021). SquiggleNet: real-time, direct classification of nanopore signals. *Genome Biology* 22(1). doi:10.1186/s13059-021-02511-y.

Baslan, T., Kovaka, S., Sedlazeck, F. J., Zhang, Y., Wappel, R., Tian, S., Lowe, S. W., Goodwin, S., and Schatz, M. C. (2021). High resolution copy number inference in cancer using short-molecule nanopore sequencing. *Nucleic Acids Research* 49(21): e124–e124. doi:10.1093/nar/gkab812.

Beuf, K. D., Schrijver, J. D., Thas, O., Criekinge, W. V., Irizarry, R. A., and Clement, L. (2012). Improved base-calling and quality scores for 454 sequencing based on a hurdle poisson model. *BMC Bioinformatics* 13(1). doi:10.1186/1471-2105-13-303.

Beyter, D., Ingimundardottir, H., Oddsson, A., Eggertsson, H. P., Bjornsson, E., Jonsson, H., Atlason, B. A., Kristmundsdottir, S., Mehringer, S., Hardarson, M. T., Gudjonsson, S. A., Magnusdottir, D. N., Jonasdottir, A., Jonasdottir, A., Kristjansson, R. P., Sverrisson, S. T., Holley, G., Palsson, G., Stefansson, O. A., Eyjolfsson, G., Olafsson, I., Sigurdardottir, O., Torfason, B., Masson, G., Helgason, A., Thorsteinsdottir, U., Holm, H., Gudbjartsson, D. F., Sulem, P., Magnusson, O. T., Halldorsson, B. V., and Stefansson, K. (2021). Long-read sequencing of 3, 622 icelanders provides insight into the role of structural variants in human diseases and other traits. *Nature Genetics* 53(6): 779–786. doi:10.1038/s41588-021-00865-4.

Bezrukov, S. (2000). Ion channels as molecular coulter counters to probe metabolite transport. *Journal of Membrane Biology* 174(1): 1–13. doi:10.1007/s002320001026.

Boemo, M. A. (2021). DNAscent v2: detecting replication forks in nanopore sequencing data with deep learning. *BMC Genomics* 22(1). doi:10.1186/s12864-021-07736-6.

Bokeh Development Team (2018). *Bokeh: Python library for interactive visualization*. **URL:** *https://bokeh.pydata.org/en/latest/*

Boža, V., Brejová, B., and Vinař, T. (2017). DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. *PLOS ONE* 12(6): e0178,751. doi:10.1371/journal.pone.0178751.

Boža, V., Perešíni, P., Brejová, B., and Vinař, T. (2020). DeepNano-blitz: a fast base caller for MinION nanopore sequencers. *Bioinformatics* 36(14): 4191–4192. doi:10.1093/bioinformatics/btaa297.

Branton, D. (2019). Nanopore structure, assembly, and sensing. In *Nanopore Sequencing*, pages 49–58. WORLD SCIENTIFIC. doi:10.1142/9789813270619_0004.

Branton, D. and Deamer, D. (2019). *Nanopore Sequencing*. WORLD SCIENTIFIC. doi:10.1142/10995.

Branton, D., Deamer, D. W., Marziali, A., Bayley, H., Benner, S. A., Butler, T., Di Ventra, M., Garaj, S., Hibbs, A., Huang, X., Jovanovich, S. B., Krstic, P. S., Lindsay, S., Ling, X. S., Mastrangelo, C. H., Meller, A., Oliver, J. S., Pershin, Y. V., Ramsey, J. M., Riehn, R., Soni, G. V., Tabard-Cossa, V., Wanunu, M., Wiggin, M., and Schloss, J. A. (2008). The potential and challenges of nanopore sequencing. *Nat Biotechnol* 26(10): 1146–53. ISSN 1546-1696 (Electronic) 1087-0156 (Linking). doi: 10.1038/nbt.1495.

Brown, C. (2015). AGBT 2012 presentation (oxford nanopore technologies). doi: 10.7490/f1000research.1110935.1.

Byrd, A. K., Matlock, D. L., Bagchi, D., Aarattuthodiyil, S., Harrison, D., Croquette, V., and Raney, K. D. (2012). Dda helicase tightly couples translocation on single-stranded DNA to unwinding of duplex DNA: Dda is an optimally active helicase. *Journal of Molecular Biology* 420(3): 141–154. doi:10.1016/j.jmb.2012.04.007.

Byrd, A. K. and Raney, K. D. (2019). Helicases and DNA motor proteins. In *Nanopore Sequencing*, pages 59–74. WORLD SCIENTIFIC. doi:10.1142/9789813270619_0005.

Castro-Wallace, S. L., Chiu, C. Y., John, K. K., Stahl, S. E., Rubins, K. H., McIntyre, A. B. R., Dworkin, J. P., Lupisella, M. L., Smith, D. J., Botkin, D. J., Stephenson, T. A., Juul, S., Turner, D. J., Izquierdo, F., Federman, S., Stryke, D., Somasekar, S., Alexander, N., Yu, G., Mason, C. E., and Burton, A. S. (2017). Nanopore DNA sequencing and genome assembly on the international space station. *Scientific Reports* 7(1). doi:10.1038/s41598-017-18364-0.

Cherf, G. M., Lieberman, K. R., Rashid, H., Lam, C. E., Karplus, K., and Akeson, M. (2012). Automated forward and reverse ratcheting of DNA in a nanopore at 5-Å precision. *Nature Biotechnology* 30(4): 344–348. doi:10.1038/nbt.2147.

Chin, F. Y. L., Leung, H. C. M., and Yiu, S. M. (2014). Sequence assembly using next generation sequencing data—challenges and solutions. *Science China Life Sciences* 57(11): 1140–1148. doi:10.1007/s11427-014-4752-9.

Clarke, J. (2019). Development of multipore sequencing instruments. In *Nanopore Sequencing*, pages 75–89. WORLD SCIENTIFIC. doi:10.1142/9789813270619_0006.

Collette, A. (2013). *Python and HDF5*. O'Reilly.

Coulter, W. H. (1953). Means for counting particles suspended in a fluid. US Patent 2,656,508.
URL: *https://patents.google.com/patent/US2656508A*

Crick, F. (1970). Central dogma of molecular biology. *Nature* 227(5258): 561–563. doi:10.1038/227561a0.

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., and Li, H. (2021). Twelve years of SAMtools and BCFtools. *GigaScience* 10(2). ISSN 2047-217X. doi:10.1093/gigascience/giab008.

David, M., Dursi, L. J., Yao, D., Boutros, P. C., and Simpson, J. T. (2016). Nanocall: an open source basecaller for oxford nanopore sequencing data. *Bioinformatics* 33(1): 49–55. doi:10.1093/bioinformatics/btw569.

der Verren, S. E. V., Gerven, N. V., Jonckheere, W., Hambley, R., Singh, P., Kilgour, J., Jordan, M., Wallace, E. J., Jayasinghe, L., and Remaut, H. (2020). A dual-constriction biological nanopore resolves homonucleotide sequences with high fidelity. *Nature Biotechnology* 38(12): 1415–1420. doi:10.1038/s41587-020-0570-8.

Dressman, D., Yan, H., Traverso, G., Kinzler, K. W., and Vogelstein, B. (2003). Transforming single DNA molecules into fluorescent magnetic particles for detection and enumeration of genetic variations. *Proceedings of the National Academy of Sciences* 100(15): 8817–8822. doi:10.1073/pnas.1133470100.

Dunn, T., Sadasivan, H., Wadden, J., Goliya, K., Chen, K.-Y., Das, R., Blaauw, D., and Narayanasamy, S. (2021). Squigglefilter: An accelerator for portable virus detection. doi:10.48550/arXiv.2108.06610.

Edwards, A., Debbonaire, A. R., Nicholls, S. M., Rassner, S. M., Sattler, B., Cook, J. M., Davy, T., Soares, A. R., Mur, L. A., and Hodson, A. J. (2016). In-field metagenome and 16s rRNA gene amplicon nanopore sequencing robustly characterize glacier microbiota. *bioRxiv* doi:10.1101/073965.

Edwards, H. S., Krishnakumar, R., Sinha, A., Bird, S. W., Patel, K. D., and Bartsch, M. S. (2019). Real-time selective sequencing with RUBRIC: Read until with basecall and reference-informed criteria. *Scientific Reports* 9(1). doi:10.1038/s41598-019-47857-3.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., deWinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden, D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J., and Turner, S. (2009). Real-time DNA sequencing from single polymerase molecules. *Science* 323(5910): 133–138. doi:10.1126/science.1162986.

Ewing, B., Hillier, L., Wendl, M. C., and Green, P. (1998). Base-calling of automated sequencer traces UsingPhred. i. accuracy assessment. *Genome Research* 8(3): 175–185. doi:10.1101/gr.8.3.175.

Fedurco, M. (2006). BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies. *Nucleic Acids Research* 34(3): e22–e22. doi:10.1093/nar/gnj023.

Forbes, S. A., Bindal, N., Bamford, S., Cole, C., Kok, C. Y., Beare, D., Jia, M., Shepherd, R., Leung, K., Menzies, A. et al. (2010). Cosmic: mining complete cancer genomes in the catalogue of somatic mutations in cancer. *Nucleic acids research* 39(suppl_1): D945–D950. doi:10.1093/nar/gkq929.

Gilpatrick, T., Lee, I., Graham, J. E., Raimondeau, E., Bowen, R., Heron, A., Downs, B., Sukumar, S., Sedlazeck, F. J., and Timp, W. (2020). Targeted nanopore sequencing with cas9-guided adapter ligation. *Nature Biotechnology* 38(4): 433–438. doi:10.1038/s41587-020-0407-5.

Gnirke, A., Melnikov, A., Maguire, J., Rogov, P., LeProust, E. M., Brockman, W., Fennell, T., Giannoukos, G., Fisher, S., Russ, C., Gabriel, S., Jaffe, D. B., Lander, E. S., and Nusbaum, C. (2009). Solution hybrid selection with ultra-long oligonucleotides for massively parallel targeted sequencing. *Nature Biotechnology* 27(2): 182–189. doi:10.1038/nbt.1523.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet* 17(6): 333–51. ISSN 1471-0064 (Electronic) 1471-0056 (Linking). doi:10.1038/nrg.2016.49.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature* 585(7825): 357–362. doi:10.1038/s41586-020-2649-2.

Heller, D. and Vingron, M. (2019). SVIM: structural variant identification using mapped long reads. *Bioinformatics* 35(17): 2907–2915. doi:10.1093/bioinformatics/btz041.

Heron, A. J. (2019). Molecular engineering DNA and RNA for nanopore sequencing. In *Nanopore Sequencing*, pages 107–146. WORLD SCIENTIFIC. doi:10.1142/9789813270619_0008.

Hert, D. G., Fredlake, C. P., and Barron, A. E. (2008). Advantages and limitations of next-generation sequencing technologies: A comparison of electrophoresis and non-electrophoresis methods. *ELECTROPHORESIS* 29(23): 4618–4626. doi:10.1002/elps.200800456.

Howorka, S., Cheley, S., and Bayley, H. (2001). Sequence-specific detection of individual DNA strands using engineered nanopores. *Nature Biotechnology* 19(7): 636–639. doi:10.1038/90236.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering* 9(3): 90–95. doi:10.1109/MCSE.2007.55.

International Human Genome Sequencing, C. (2001). Initial sequencing and analysis of the human genome. *Nature* 409: 860. doi:10.1038/35057062.

International Human Genome Sequencing, C. (2004). Finishing the euchromatic sequence of the human genome. *Nature* 431(7011): 931–945. doi:10.1038/nature03001.

Ip, C. L. C., Loose, M., Tyson, J. R., de Cesare, M., Brown, B. L., Jain, M., Leggett, R. M., Eccles, D. A., Zalunin, V., Urban, J. M., Piazza, P., Bowden, R. J., Paten, B., Mwaigwisya, S., Batty, E. M., Simpson, J. T., Snutch, T. P., Birney, E., Buck, D., Goodwin, S., Jansen, H. J., O'Grady, J., and Olsen, H. E. (2015). Minion analysis and reference consortium: Phase 1 data release and analysis. *F1000Research* ISSN 2046-1402. doi:10.12688/f1000research.7201.1.

Jain, M., Fiddes, I., Miga, K. H., Olsen, H. E., Paten, B., and Akeson, M. (2015). Improved data analysis for the minion nanopore sequencer. *Nature methods* 12(4): 351–356. ISSN 1548-7091 1548-7105. doi:10.1038/nmeth.3290.

Jain, M., Koren, S., Miga, K. H., Quick, J., Rand, A. C., Sasani, T. A., Tyson, J. R., Beggs, A. D., Dilthey, A. T., Fiddes, I. T., Malla, S., Marriott, H., Nieto, T., O'Grady, J., Olsen, H. E., Pedersen, B. S., Rhie, A., Richardson, H., Quinlan, A. R., Snutch, T. P., Tee, L., Paten, B., Phillippy, A. M., Simpson, J. T., Loman, N. J., and Loose, M. (2018a). Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature Biotechnology* 36(4): 338–345. doi:10.1038/nbt.4060.

Jain, M., Olsen, H. E., Paten, B., and Akeson, M. (2016). The oxford nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biology* 17(1). doi:10.1186/s13059-016-1103-0.

Jain, M., Olsen, H. E., Turner, D. J., Stoddart, D., Bulazel, K. V., Paten, B., Haussler, D., Willard, H. F., Akeson, M., and Miga, K. H. (2018b). Linear assembly of a human centromere on the y chromosome. *Nature Biotechnology* 36(4): 321–323. doi:10.1038/nbt.4109.

Jiang, T., Liu, Y., Jiang, Y., Li, J., Gao, Y., Cui, Z., Liu, Y., Liu, B., and Wang, Y. (2020). Long-read-based human genomic structural variation detection with cuteSV. *Genome Biology* 21(1). doi:10.1186/s13059-020-02107-y.

Johnson, J. S., Spakowicz, D. J., Hong, B.-Y., Petersen, L. M., Demkowicz, P., Chen, L., Leopold, S. R., Hanson, B. M., Agresta, H. O., Gerstein, M., Sodergren, E., and Weinstock, G. M. (2019). Evaluation of 16s rRNA gene sequencing for species and strain-level microbiome analysis. *Nature Communications* 10(1). doi:10.1038/s41467-019-13036-1.

Kasianowicz, J. J., Brandin, E., Branton, D., and Deamer, D. W. (1996). Characterization of individual polynucleotide molecules using a membrane channel. *Proceedings of the National Academy of Sciences* 93(24): 13,770–13,773. doi:10.1073/pnas.93.24.13770.

Kiełbasa, S. M., Wan, R., Sato, K., Horton, P., and Frith, M. C. (2011). Adaptive seeds tame genomic sequence comparison. *Genome Research* 21(3): 487–493. doi:10.1101/gr.113985.110.

Kim, D., Song, L., Breitwieser, F. P., and Salzberg, S. L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Research* 26(12): 1721–1729. doi:10.1101/gr.210641.116.

Kolmogorov, M., Bickhart, D. M., Behsaz, B., Gurevich, A., Rayko, M., Shin, S. B., Kuhn, K., Yuan, J., Polevikov, E., Smith, T. P. L., and Pevzner, P. A. (2020). metaFlye: scalable long-read metagenome assembly using repeat graphs. *Nature Methods* 17(11): 1103–1110. doi:10.1038/s41592-020-00971-x.

Kolmogorov, M., Yuan, J., Lin, Y., and Pevzner, P. A. (2019). Assembly of long, error-prone reads using repeat graphs. *Nature Biotechnology* 37(5): 540–546. doi:10.1038/s41587-019-0072-8.

Koren, S. and Phillippy, A. M. (2015). One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Current Opinion in Microbiology* 23: 110–120. doi:10.1016/j.mib.2014.11.014.

Kovaka, S., Fan, Y., Ni, B., Timp, W., and Schatz, M. C. (2020). Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED. *Nature Biotechnology* doi:10.1038/s41587-020-0731-9.

Kruskal, J. B. (1983). An overview of sequence comparison: Time warps, string edits, and macromolecules. *SIAM Review* 25(2): 201–237. ISSN 00361445.

Leggett, R. M. and Clark, M. D. (2017). A world of opportunities with nanopore sequencing. *J Exp Bot* 68(20): 5419–5429. ISSN 1460-2431 (Electronic) 0022-0957 (Linking). doi:10.1093/jxb/erx289.

Li, H. (2016). Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* 32(14): 2103–2110. doi:10.1093/bioinformatics/btw152.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 34(18): 3094–3100. doi:10.1093/bioinformatics/bty191.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and and, R. D. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics* 25(16): 2078–2079. doi:10.1093/bioinformatics/btp352.

Lieberman, K. R., Cherf, G. M., Doody, M. J., Olasagasti, F., Kolodji, Y., and Akeson, M. (2010). Processive replication of single DNA molecules in a nanopore catalyzed by phi29 DNA polymerase. *Journal of the American Chemical Society* 132(50): 17,961–17,972. doi:10.1021/ja1087612.

Lischer, H. E. L. and Shimizu, K. K. (2017). Reference-guided de novo assembly approach improves genome reconstruction for related species. *BMC Bioinformatics* 18(1). doi:10.1186/s12859-017-1911-6.

Liu, T., Xu, F., Du, X., Lai, D., Liu, T., Zhao, Y., Huang, Q., Jiang, L., Huang, W., Cheng, W., and Liu, Z. (2010). Establishment and characterization of multi-drug resistant, prostate carcinoma-initiating stem-like cells from human prostate cancer cell lines 22rv1. *Molecular and Cellular Biochemistry* 340(1-2): 265–273. doi: 10.1007/s11010-010-0426-5.

Logsdon, G. A., Vollger, M. R., and Eichler, E. E. (2020). Long-read human genome sequencing and its applications. *Nature Reviews Genetics* 21(10): 597–614. doi: 10.1038/s41576-020-0236-x.

Loman, N. J., Quick, J., and Simpson, J. T. (2015). A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature Methods* 12(8): 733–735. doi:10.1038/nmeth.3444.

Loose, M. W. (2017). The potential impact of nanopore sequencing on human genetics. *Human Molecular Genetics* 26(R2): R202–R207. doi:10.1093/hmg/ddx287.

Loose, M., Malla, S., and Stout, M. (2016). Real-time selective sequencing using nanopore technology. *Nature Methods* 13(9): 751–754. doi:10.1038/nmeth.3930.

Maio, N. D., Manser, C., Munro, R., Birney, E., Loose, M., and Goldman, N. (2020). BOSS-RUNS: a flexible and practical dynamic read sampling framework for nanopore sequencing. *bioRxiv* doi:10.1101/2020.02.07.938670.

Manrao, E. A., Derrington, I. M., Laszlo, A. H., Langford, K. W., Hopper, M. K., Gillgren, N., Pavlenok, M., Niederweis, M., and Gundlach, J. H. (2012). Reading DNA at single-nucleotide resolution with a mutant MspA nanopore and phi29 DNA polymerase. *Nature Biotechnology* 30(4): 349–353. doi:10.1038/nbt.2171.

Mardis, E. R. (2013). Next-generation sequencing platforms. *Annual Review of Analytical Chemistry* 6(1): 287–303. doi:10.1146/annurev-anchem-062012-092628.

Marquet, M., Zöllkau, J., Pastuschek, J., Viehweger, A., Schleußner, E., Makarewicz, O., Pletz, M. W., Ehricht, R., and Brandt, C. (2021). Evaluation of microbiome enrichment and host DNA depletion in human vaginal samples using oxford nanopore's adaptive sequencing. *bioRxiv* doi:10.1101/2021.09.15.460450.

Martin, S., Heavens, D., Lan, Y., Horsfield, S., Clark, M. D., and Leggett, R. M. (2022). Nanopore adaptive sampling: a tool for enrichment of low abundance species in metagenomic samples. *Genome Biology* 23(1). doi:10.1186/s13059-021-02582-x.

Masutani, B. and Morishita, S. (2018). A framework and an algorithm to detect low-abundance DNA by a handy sequencer and a palm-sized computer. *Bioinformatics* 35(4): 584–592. doi:10.1093/bioinformatics/bty663.

Maxam, A. M. and Gilbert, W. (1977). A new method for sequencing dna. *Proceedings of the National Academy of Sciences of the United States of America* 74(2): 560–564. ISSN 0027-8424 1091-6490.

McCarty, M. (2003). Discovering genes are made of DNA. *Nature* 421(6921): 406–406. doi:10.1038/nature01398.

McElroy, W. and Green, A. (1956). Function of adenosine triphosphate in the activation of luciferin. *Archives of Biochemistry and Biophysics* 64(2): 257–271. doi:10.1016/0003-9861(56)90268-5.

McIntyre, A. B. R., Alexander, N., Burton, A. S., Castro-Wallace, S., Chiu, C. Y., John, K. K., Stahl, S. E., Li, S., and Mason, C. E. (2017). Nanopore detection of bacterial dna base modifications. *bioRxiv* .

Miller, D. E., Sulovari, A., Wang, T., Loucks, H., Hoekzema, K., Munson, K. M., Lewis, A. P., Fuerte, E. P. A., Paschal, C. R., Walsh, T., Thies, J., Bennett, J. T., Glass, I., Dipple, K. M., Patterson, K., Bonkowski, E. S., Nelson, Z., Squire, A., Sikes, M., Beckman, E., Bennett, R. L., Earl, D., Lee, W., Allikmets, R., Perlman, S. J., Chow, P., Hing, A. V., Wenger, T. L., Adam, M. P., Sun, A., Lam, C., Chang, I., Zou, X., Austin, S. L., Huggins, E., Safi, A., Iyengar, A. K., Reddy, T. E., Majoros, W. H., Allen, A. S., Crawford, G. E., Kishnani, P. S., King, M.-C., Cherry, T., Chong, J. X., Bamshad, M. J., Nickerson, D. A., Mefford, H. C., Doherty, D., and Eichler, E. E. (2021). Targeted long-read sequencing identifies missing disease-causing variation. *The American Journal of Human Genetics* 108(8): 1436–1449. doi:10.1016/j.ajhg.2021.06.006.

Mitra, R. D., Shendure, J., Olejnik, J., Edyta-Krzymanska-Olejnik, and Church, G. M. (2003). Fluorescent in situ sequencing on polymerase colonies. *Analytical Biochemistry* 320(1): 55–65. doi:10.1016/s0003-2697(03)00291-4.

Mozziconacci, M.-J., Rosenauer, A., Restouin, A., Fanelli, M., Shao, W., Fernandez, F., Toiron, Y., Viscardi, J., Gambacorti-Passerini, C., Miller, W. H., and Lafage-Pochitaloff, M. (2002). Molecular cytogenetics of the acute promyelocytic leukemia-derived cell line NB4 and of four all-trans retinoic acid-resistant subclones. *Genes, Chromosomes and Cancer* 35(3): 261–270. doi:10.1002/gcc.10117.

Müller, C. A., Boemo, M. A., Spingardi, P., Kessler, B. M., Kriaucionis, S., Simpson, J. T., and Nieduszynski, C. A. (2019). Capturing the dynamics of genome replication on individual ultra-long nanopore sequence reads. *Nature Methods* 16(5): 429–436. doi:10.1038/s41592-019-0394-y.

Na, K., Kim, H.-S., Shim, H. S., Chang, J. H., Kang, S.-G., and Kim, S. H. (2019). Targeted next-generation sequencing panel (TruSight tumor 170) in diffuse glioma: a single institutional experience of 135 cases. *Journal of Neuro-Oncology* 142(3): 445–454. doi:10.1007/s11060-019-03114-1.

Ng, S. B., Turner, E. H., Robertson, P. D., Flygare, S. D., Bigham, A. W., Lee, C., Shaffer, T., Wong, M., Bhattacharjee, A., Eichler, E. E., Bamshad, M., Nickerson, D. A., and Shendure, J. (2009). Targeted capture and massively parallel sequencing of 12 human exomes. *Nature* 461(7261): 272–276. doi:10.1038/nature08250.

Ni, J., Yan, Q., and Yu, Y. (2013). How much metagenomic sequencing is enough to achieve a given goal? *Scientific Reports* 3(1). doi:10.1038/srep01968.

Nicholls, S. M., Poplawski, R., Bull, M. J., Underwood, A., Chapman, M., Abu-Dahab, K., Taylor, B., Colquhoun, R. M., Rowe, W. P. M., Jackson, B., Hill, V., O'Toole, Á., Rey, S., Southgate, J., Amato, R., Livett, R., Gonçalves, S., Harrison, E. M., Peacock, S. J., Aanensen, D. M., Rambaut, A., Connor, T. R., Loman, N. J., and The COVID-19 Genomics (COG-UK) Consortium (2021). CLIMB-COVID: continuous integration supporting decentralised sequencing for SARS-CoV-2 genomic surveillance. *Genome Biology* 22(1). doi:10.1186/s13059-021-02395-y.

Nicholls, S. M., Quick, J. C., Tang, S., and Loman, N. J. (2019). Ultra-deep, long-read nanopore sequencing of mock microbial community standards. *GigaScience* 8(5). doi:10.1093/gigascience/giz043.

ONT (2019). Scrappie. https://github.com/nanoporetech/scrappie.

ONT (2020). Read Until API. https://github.com/nanoporetech/read_until_api/tree/v3.0.0.

ONT (2021). Continuous development and improvement.
URL: *https://nanoporetech.com/about-us/continuous-development-and-improvement*

ONT (2021). Direct RNA Sequencing Kit. [Online; accessed 17-August-2021].
URL: *https://store.nanoporetech.com/uk/direct-rna-sequencing-kit.html*

ONT (2021). Highlights of clive g brown's technical update.
URL: *https://nanoporetech.com/about-us/news/highlights-clive-g-browns-technical-update*

ONT (2021). Ligation Sequencing Kit. [Online; accessed 17-August-2021].
URL: *https://store.nanoporetech.com/uk/ligation-sequencing-kit.html*

ONT (2021a). Medaka. https://github.com/nanoporetech/medaka.

ONT (2021b). Megalodon. https://github.com/nanoporetech/megalodon.

ONT (2021c). MinKNOW API. https://github.com/nanoporetech/minknow_api/tree/4.2.4.

ONT (2021). Native Barcoding Expansion. [Online; accessed 17-August-2021].
URL: *https://store.nanoporetech.com/uk/native-barcoding-expansion-1-12.html*

ONT (2021). ONT PyGuppy Client Lib. https://pypi.org/project/ont-pyguppy-client-lib/.

ONT (2021). Rapid Sequencing Kit. [Online; accessed 17-August-2021].
URL: *https://store.nanoporetech.com/uk/rapid-sequencing-kit.html*

ONT (2022). Ultra-long dna sequencing kit.
URL: *https://store.nanoporetech.com/uk/ultra-long-dna-sequencing-kit.html*

Pacific Biosciences (2021). Smrt sequencing - pacbio - highly accurate long-read sequencing.
URL: *https://www.pacb.com/smrt-science/smrt-sequencing/*

Parker, M. T., Knop, K., Sherwood, A. V., Schurch, N. J., Mackinnon, K., Gould, P. D., Hall, A. J., Barton, G. J., and Simpson, G. G. (2020). Nanopore direct RNA sequencing maps the complexity of arabidopsis mRNA processing and m6a modification. *eLife* 9. doi:10.7554/elife.49658.

Patel, A., Dogan, H., Payne, A., Sievers, P., Schoebe, N., Schrimpf, D., Stichel, D., Holmes, N., Euskirchen, P., Hench, J., Frank, S., Rosenstiel-Goidts, V., Ratliff, M., Etminan, N., Unterberg, A., Dieterich, C., Herold-Mende, C., Pfister, S. M., Wick, W., Schlesner, M., Loose, M., von Deimling, A., Sill, M., Jones, D. T., and Sahm, F. (2021). Rapid-CNS2: Rapid comprehensive adaptive nanopore-sequencing of CNS tumors, a proof of concept study. *bioRxiv* doi:10.1101/2021.08.09.21261784.

Payne, A., Holmes, N., Clarke, T., Munro, R., Debebe, B. J., and Loose, M. (2020). Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nature Biotechnology* doi:10.1038/s41587-020-00746-x.

Payne, A., Holmes, N., Rakyan, V., and Loose, M. (2018). BulkVis: a graphical viewer for oxford nanopore bulk FAST5 files. *Bioinformatics* 35(13): 2193–2198. doi:10.1093/bioinformatics/bty841.

Payne, A., Munro, R., Holmes, N., Moore, C., Carlile, M., and Loose, M. W. (2021). Barcode aware adaptive sampling for oxford nanopore sequencers. *bioRxiv* doi:10.1101/2021.12.01.470722.

Pedersen, B. S. and Quinlan, A. R. (2017). Mosdepth: quick coverage calculation for genomes and exomes. *Bioinformatics* 34(5): 867–868. doi:10.1093/bioinformatics/btx699.

Petersen, B.-S., Fredrich, B., Hoeppner, M. P., Ellinghaus, D., and Franke, A. (2017). Opportunities and challenges of whole-genome and -exome sequencing. *BMC Genetics* 18(1). doi:10.1186/s12863-017-0479-5.

Quail, M., Smith, M. E., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., Bertoni, A., Swerdlow, H. P., and Gu, Y. (2012). A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina MiSeq sequencers. *BMC Genomics* 13(1): 341. doi:10.1186/1471-2164-13-341.

Quick, J. (2018). Ultra-long read sequencing protocol for RAD004 v3 (protocols.io.mrxc57n). doi:10.17504/protocols.io.mrxc57n.
URL: *https://doi.org/10.17504/protocols.io.mrxc57n*

Quick, J., Loman, N. J., Duraffour, S., Simpson, J. T., Severi, E., Cowley, L., Bore, J. A., Koundouno, R., Dudas, G., Mikhail, A., Ouédraogo, N., Afrough, B., Bah, A., Baum, J. H. J., Becker-Ziaja, B., Boettcher, J. P., Cabeza-Cabrerizo, M., Camino-Sánchez, Á., Carter, L. L., Doerrbecker, J., Enkirch, T., García-Dorival, I., Hetzelt, N., Hinzmann, J., Holm, T., Kafetzopoulou, L. E., Koropogui, M., Kosgey, A., Kuisma, E., Logue, C. H., Mazzarelli, A., Meisel, S., Mertens, M., Michel, J., Ngabo, D., Nitzsche, K., Pallasch, E., Patrono, L. V., Portmann, J., Repits, J. G., Rickett, N. Y., Sachse, A., Singethan, K., Vitoriano, I., Yemanaberhan, R. L., Zekeng, E. G., Racine, T., Bello, A., Sall, A. A., Faye, O., Faye, O., Magassouba, N., Williams, C. V., Amburgey, V., Winona, L., Davis, E., Gerlach, J., Washington, F., Monteil, V., Jourdain, M., Bererd, M., Camara, A., Somlare, H., Camara, A., Gerard, M., Bado, G., Baillet, B., Delaune, D., Nebie, K. Y., Diarra, A., Savane, Y.,

Pallawo, R. B., Gutierrez, G. J., Milhano, N., Roger, I., Williams, C. J., Yattara, F., Lewandowski, K., Taylor, J., Rachwal, P., Turner, D. J., Pollakis, G., Hiscox, J. A., Matthews, D. A., Shea, M. K. O., Johnston, A. M., Wilson, D., Hutley, E., Smit, E., Caro, A. D., Wölfel, R., Stoecker, K., Fleischmann, E., Gabriel, M., Weller, S. A., Koivogui, L., Diallo, B., Keïta, S., Rambaut, A., Formenty, P., Günther, S., and Carroll, M. W. (2016). Real-time, portable genome sequencing for ebola surveillance. *Nature* 530(7589): 228–232. doi:10.1038/nature16996.

Quinlan, A. R. and Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 26(6): 841–842. doi:10.1093/bioinformatics/btq033.

Raina, M. and Ibba, M. (2014). tRNAs as regulators of biological processes. *Frontiers in Genetics* 5. doi:10.3389/fgene.2014.00171.

Rand, A. C., Jain, M., Eizenga, J., Musselman-Brown, A., Olsen, H. E., Akeson, M., and Paten, B. (2016). Cytosine variant calling with high-throughput nanopore sequencing. *bioRxiv* .

Rang, F. J., Kloosterman, W. P., and de Ridder, J. (2018). From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biology* 19(1). doi:10.1186/s13059-018-1462-9.

Ravanelli, M., Brakel, P., Omologo, M., and Bengio, Y. (2018). Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence* 2(2): 92–102. doi:10.1109/tetci.2017.2762739.

Ronaghi, M., Karamohamed, S., Pettersson, B., Uhlén, M., and Nyrén, P. (1996). Real-time DNA sequencing using detection of pyrophosphate release. *Analytical Biochemistry* 242(1): 84–89. doi:10.1006/abio.1996.0432.

Ronaghi, M., Uhlén, M., and Nyrén, P. (1998). A sequencing method based on real-time pyrophosphate. *Science* 281(5375): 363–365. doi:10.1126/science.281.5375.363.

Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M. J., Edwards, M., Hoon, J., Simons, J. F., Marran, D., Myers, J. W., Davidson, J. F., Branting, A., Nobile, J. R., Puc, B. P., Light, D., Clark, T. A., Huber, M., Branciforte, J. T., Stoner, I. B., Cawley, S. E., Lyons, M., Fu, Y., Homer, N., Sedova, M., Miao, X., Reed, B., Sabina, J., Feierstein,

E., Schorn, M., Alanjary, M., Dimalanta, E., Dressman, D., Kasinskas, R., Sokolsky, T., Fidanza, J. A., Namsaraev, E., McKernan, K. J., Williams, A., Roth, G. T., and Bustillo, J. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature* 475(7356): 348–352. doi:10.1038/nature10242.

Saenger, W. (1984). *Principles of Nucleic Acid Structure*. Springer Advanced Texts in Chemistry. Springer New York. ISBN 9780387907611; 0387907610; 9781461251903; 1461251907.

Salvador, S. and Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11(5): 561–580. ISSN 15714128, 1088467X. doi:10.3233/IDA-2007-11508.

Salzberg, S. L., Phillippy, A. M., Zimin, A., Puiu, D., Magoc, T., Koren, S., Treangen, T. J., Schatz, M. C., Delcher, A. L., Roberts, M., Marçais, G., Pop, M., and Yorke, J. A. (2011). GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research* 22(3): 557–567. doi:10.1101/gr.131383.111.

Sambrook, J. and Russell, D. (2001). *Molecular cloning : a laboratory manual*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, N.Y. ISBN 978-0-87969-576-7.

Sanger, F. and Coulson, A. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology* 94(3): 441–448. doi:10.1016/0022-2836(75)90213-2.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America* 74(12): 5463–5467. ISSN 0027-8424 1091-6490.

Schneider, V. A., Graves-Lindsay, T., Howe, K., Bouk, N., Chen, H.-C., Kitts, P. A., Murphy, T. D., Pruitt, K. D., Thibaud-Nissen, F., Albracht, D., Fulton, R. S., Kremitzki, M., Magrini, V., Markovic, C., McGrath, S., Steinberg, K. M., Auger, K., Chow, W., Collins, J., Harden, G., Hubbard, T., Pelan, S., Simpson, J. T., Threadgold, G., Torrance, J., Wood, J. M., Clarke, L., Koren, S., Boitano, M., Peluso, P., Li, H., Chin, C.-S., Phillippy, A. M., Durbin, R., Wilson, R. K., Flicek, P., Eichler, E. E., and Church, D. M. (2017). Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Research* 27(5): 849–864. doi:10.1101/gr.213611.116.

Schreiber, J., Wescoe, Z. L., Abu-Shumays, R., Vivian, J. T., Baatar, B., Karplus, K., and Akeson, M. (2013). Error rates for nanopore discrimination among cytosine, methylcytosine, and hydroxymethylcytosine along individual dna strands. *Proceedings of the National Academy of Sciences* 110(47): 18,910.

Sedlazeck, F. J., Rescheneder, P., Smolka, M., Fang, H., Nattestad, M., von Haeseler, A., and Schatz, M. C. (2018). Accurate detection of complex structural variations using single-molecule sequencing. *Nature Methods* 15(6): 461–468. doi:10.1038/s41592-018-0001-7.

Sharon, D., Tilgner, H., Grubert, F., and Snyder, M. (2013). A single-molecule long-read survey of the human transcriptome. *Nature Biotechnology* 31(11): 1009–1014. doi:10.1038/nbt.2705.

Shendure, J. and Ji, H. (2008). Next-generation DNA sequencing. *Nature Biotechnology* 26(10): 1135–1145. doi:10.1038/nbt1486.

Shendure, J., Porreca, G. J., Reppas, N. B., Lin, X., McCutcheon, J. P., Rosenbaum, A. M., Wang, M. D., Zhang, K., Mitra, R. D., and Church, G. M. (2005). Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* 309(5741): 1728–1732. doi:10.1126/science.1117389.

Shin, G., Grimes, S. M., Lee, H., Lau, B. T., Xia, L. C., and Ji, H. P. (2017). CRISPR–cas9-targeted fragmentation and selective sequencing enable massively parallel microsatellite analysis. *Nature Communications* 8(1). doi:10.1038/ncomms14291.

Siegfried, A., Rousseau, A., Maurage, C.-A., Pericart, S., Nicaise, Y., Escudie, F., Grand, D., Delrieu, A., Gomez-Brouchet, A., Guellec, S. L., Franchet, C., Boetto, S., Vinchon, M., Sol, J.-C., Roux, F.-E., Rigau, V., Bertozzi, A.-I., Jones, D. T. W., Figarella-Branger, D., and Uro-Coste, E. (2018). EWSR1-PATZ1gene fusion may define a new glioneuronal tumor entity. *Brain Pathology* 29(1): 53–62. doi:10.1111/bpa.12619.

Simpson, J. T., Workman, R. E., Zuzarte, P. C., David, M., Dursi, L. J., and Timp, W. (2017). Detecting DNA cytosine methylation using nanopore sequencing. *Nature Methods* 14: 407. doi:10.1038/nmeth.4184.

Siqueira, J. F., Fouad, A. F., and Rôças, I. N. (2012). Pyrosequencing as a tool for better understanding of human microbiomes. *Journal of Oral Microbiology* 4(1): 10,743. doi:10.3402/jom.v4i0.10743.

Stancu, M. C., van Roosmalen, M. J., Renkens, I., Nieboer, M. M., Middelkamp, S., de Ligt, J., Pregno, G., Giachino, D., Mandrile, G., Valle-Inclan, J. E., Korzelius, J., de Bruijn, E., Cuppen, E., Talkowski, M. E., Marschall, T., de Ridder, J., and Kloosterman, W. P. (2017). Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *Nature Communications* 8(1). doi:10.1038/s41467-017-01343-4.

Stevanovski, I., Chintalaphani, S. R., Gamaarachchi, H., Ferguson, J. M., Pineda, S. S., Scriba, C. K., Tchan, M., Fung, V., Ng, K., Cortese, A., Houlden, H., Dobson-Stone, C., Fitzpatrick, L., Halliday, G., Ravenscroft, G., Davis, M. R., Laing, N. G., Fellner, A., Kennerson, M., Kumar, K. R., and Deveson, I. W. (2021). Comprehensive genetic diagnosis of tandem repeat expansion disorders with programmable targeted nanopore sequencing. *bioRxiv* doi:10.1101/2021.09.27.21263187.

Stoiber, M. and Brown, J. (2017). BasecRAWller: Streaming Nanopore Basecalling Directly from Raw Signal. *BioRxiv* doi:10.1101/133058.

Tate, J. G., Bamford, S., Jubb, H. C., Sondka, Z., Beare, D. M., Bindal, N., Boutselakis, H., Cole, C. G., Creatore, C., Dawson, E., Fish, P., Harsha, B., Hathaway, C., Jupe, S. C., Kok, C. Y., Noble, K., Ponting, L., Ramshaw, C. C., Rye, C. E., Speedy, H. E., Stefancsik, R., Thompson, S. L., Wang, S., Ward, S., Campbell, P. J., and Forbes, S. A. (2018). COSMIC: the catalogue of somatic mutations in cancer. *Nucleic Acids Research* 47(D1): D941–D947. doi:10.1093/nar/gky1015.

Teng, H., Cao, M. D., Hall, M. B., Duarte, T., Wang, S., and Coin, L. J. M. (2018). Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. *GigaScience* 7(5). doi:10.1093/gigascience/giy037.

The Pandas Development Team (2021). pandas-dev/pandas: Pandas 1.3.0. doi:10.5281/zenodo.5060318.

Tyson, J. R., O'Neil, N. J., Jain, M., Olsen, H. E., Hieter, P., and Snutch, T. P. (2018). Minion-based long-read sequencing and assembly extends the caenorhabditis elegans reference genome. *Genome Research* 28(2): 266–274. ISSN 1088-9051 1549-5469. doi:10.1101/gr.221184.117.

Ulrich, J.-U., Lutfi, A., Rutzen, K., and Renard, B. Y. (2022). ReadBouncer: Precise and scalable adaptive sampling for nanopore sequencing. *bioRxiv* doi:10.1101/2022.02.01.478636.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., Skupski, M., Subramanian, G., Thomas, P. D., Zhang, J., Miklos, G. L. G., Nelson, C., Broder, S., Clark, A. G., Nadeau, J., McKusick, V. A., Zinder, N., Levine, A. J., Roberts, R. J., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I., Charlab, R., Chaturvedi, K., Deng, Z., Francesco, V. D., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A. E., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T. J., Higgins, M. E., Ji, R.-R., Ke, Z., Ketchum, K. A., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G. V., Milshina, N., Moore, H. M., Naik, A. K., Narayan, V. A., Neelam, B., Nusskern, D., Rusch, D. B., Salzberg, S., Shao, W., Shue, B., Sun, J., Wang, Z. Y., Wang, A., Wang, X., Wang, J., Wei, M.-H., Wides, R., Xiao, C., Yan, C., Yao, A., Ye, J., Zhan, M., Zhang, W., Zhang, H., Zhao, Q., Zheng, L., Zhong, F., Zhong, W., Zhu, S. C., Zhao, S., Gilbert, D., Baumhueter, S., Spier, G., Carter, C., Cravchik, A., Woodage, T., Ali, F., An, H., Awe, A., Baldwin, D., Baden, H., Barnstead, M., Barrow, I., Beeson, K., Busam, D., Carver, A., Center, A., Cheng, M. L., Curry, L., Danaher, S., Davenport, L., Desilets, R., Dietz, S., Dodson, K., Doup, L., Ferriera, S., Garg, N., Gluecksmann, A., Hart, B., Haynes, J., Haynes, C., Heiner, C., Hladun, S., Hostin, D., Houck, J., Howland, T., Ibegwam, C., Johnson, J., Kalush, F., Kline, L., Koduru, S., Love, A., Mann, F., May, D., McCawley, S., McIntosh, T., McMullen, I., Moy, M., Moy, L., Murphy, B., Nelson, K., Pfannkoch, C., Pratts, E., Puri, V., Qureshi, H., Reardon, M., Rodriguez, R., Rogers, Y.-H., Romblad, D., Ruhfel, B., Scott, R., Sitter, C., Smallwood, M., Stewart, E., Strong, R., Suh, E., Thomas, R., Tint, N. N., Tse, S., Vech, C., Wang, G., Wetter, J., Williams, S., Williams, M., Windsor, S., Winn-Deen, E., Wolfe, K., Zaveri, J., Zaveri, K., Abril, J. F., Guigó, R., Campbell, M. J., Sjolander, K. V., Karlak, B., Kejariwal, A., Mi, H., Lazareva, B., Hatton, T., Narechania, A., Diemer, K., Muruganujan, A., Guo, N., Sato, S., Bafna, V., Istrail, S., Lippert, R., Schwartz, R., Walenz, B., Yooseph, S., Allen, D., Basu, A., Baxendale, J., Blick, L., Caminha, M., Carnes-Stine, J., Caulk, P., Chiang, Y.-H., Coyne, M., Dahlke, C., Mays, A. D., Dombroski, M., Donnelly, M., Ely, D., Esparham, S., Fosler, C., Gire, H., Glanowski, S., Glasser, K., Glodek, A., Gorokhov, M., Graham, K., Gropman, B., Harris, M., Heil, J., Henderson, S., Hoover, J., Jennings, D., Jordan, C., Jordan, J., Kasha, J., Kagan, L., Kraft, C., Levitsky, A., Lewis, M., Liu, X., Lopez, J., Ma,

D., Majoros, W., McDaniel, J., Murphy, S., Newman, M., Nguyen, T., Nguyen, N., Nodell, M., Pan, S., Peck, J., Peterson, M., Rowe, W., Sanders, R., Scott, J., Simpson, M., Smith, T., Sprague, A., Stockwell, T., Turner, R., Venter, E., Wang, M., Wen, M., Wu, D., Wu, M., Xia, A., Zandieh, A., and Zhu, X. (2001). The sequence of the human genome. *Science* 291(5507): 1304–1351. doi:10.1126/science.1058040.

Votintseva, A. A., Bradley, P., Pankhurst, L., del Ojo Elias, C., Loose, M., Nilgiriwala, K., Chatterjee, A., Smith, E. G., Sanderson, N., Walker, T. M., Morgan, M. R., Wyllie, D. H., Walker, A. S., Peto, T. E. A., Crook, D. W., and Iqbal, Z. (2017). Same-day diagnostic and surveillance data for tuberculosis via whole-genome sequencing of direct respiratory samples. *Journal of Clinical Microbiology* 55(5): 1285–1298. doi:10.1128/jcm.02483-16.

Walter, N. G. and Engelke, D. R. (2002). Ribozymes: catalytic rnas that cut things, make things, and do odd and useful jobs. *Biologist (London, England)* 49(5): 199—203. ISSN 0006-3347.

Wanunu, M. (2012). Nanopores: A journey towards DNA sequencing. *Physics of Life Reviews* 9(2): 125–158. doi:10.1016/j.plrev.2012.05.010.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software* 6(60): 3021. doi:10.21105/joss.03021.

Watson, J. D. and Crick, F. H. C. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature* 171(4356): 737–738. doi:10.1038/171737a0.
URL: *https://doi.org/10.1038/171737a0*

Wescoe, Z. L., Schreiber, J., and Akeson, M. (2014). Nanopores discriminate among five c5-cytosine variants in dna. *Journal of the American Chemical Society* 136(47): 16,582–16,587. ISSN 0002-7863. doi:10.1021/ja508527b.

Wick, R. R., Judd, L. M., and Holt, K. E. (2019). Performance of neural network basecalling tools for oxford nanopore sequencing. *Genome Biology* 20(1). doi:10.1186/s13059-019-1727-y.

Wilfinger, W. W., Mackey, K., and Chomczynski, P. (1997). Effect of pH and ionic strength on the spectrophotometric assessment of nucleic acid purity. *BioTechniques* 22(3): 474–481. doi:10.2144/97223st01.

Wood, D. E., Lu, J., and Langmead, B. (2019). Improved metagenomic analysis with kraken 2. *Genome Biology* 20(1). doi:10.1186/s13059-019-1891-0.

Workman, R. E., Tang, A. D., Tang, P. S., Jain, M., Tyson, J. R., Razaghi, R., Zuzarte, P. C., Gilpatrick, T., Payne, A., Quick, J., Sadowski, N., Holmes, N., de Jesus, J. G., Jones, K. L., Soulette, C. M., Snutch, T. P., Loman, N., Paten, B., Loose, M., Simpson, J. T., Olsen, H. E., Brooks, A. N., Akeson, M., and Timp, W. (2019). Nanopore native RNA sequencing of a human poly(a) transcriptome. *Nature Methods* 16(12): 1297–1305. doi:10.1038/s41592-019-0617-2.

Zalokar, M. (1960). Sites of protein and ribonucleic acid synthesis in the cell. *Experimental Cell Research* 19(3): 559–576. doi:10.1016/0014-4827(60)90064-1.

Zhang, H., Li, H., Jain, C., Cheng, H., Au, K. F., Li, H., and Aluru, S. (2021). Real-time mapping of nanopore raw signals. *Bioinformatics* 37(Supplement_1): i477–i483.

Zhao, N., Cao, J., Xu, J., Liu, B., Liu, B., Chen, D., Xia, B., Chen, L., Zhang, W., Zhang, Y., Zhang, X., Duan, Z., Wang, K., Xie, F., Xiao, K., Yan, W., Xie, L., Zhou, H., and Wang, J. (2021). Targeting RNA with next- and third-generation sequencing improves pathogen identification in clinical samples. *Advanced Science* 8(23): 2102,593. doi:10.1002/advs.202102593.

Appendices

Appendix A

Raw Nanopore Data

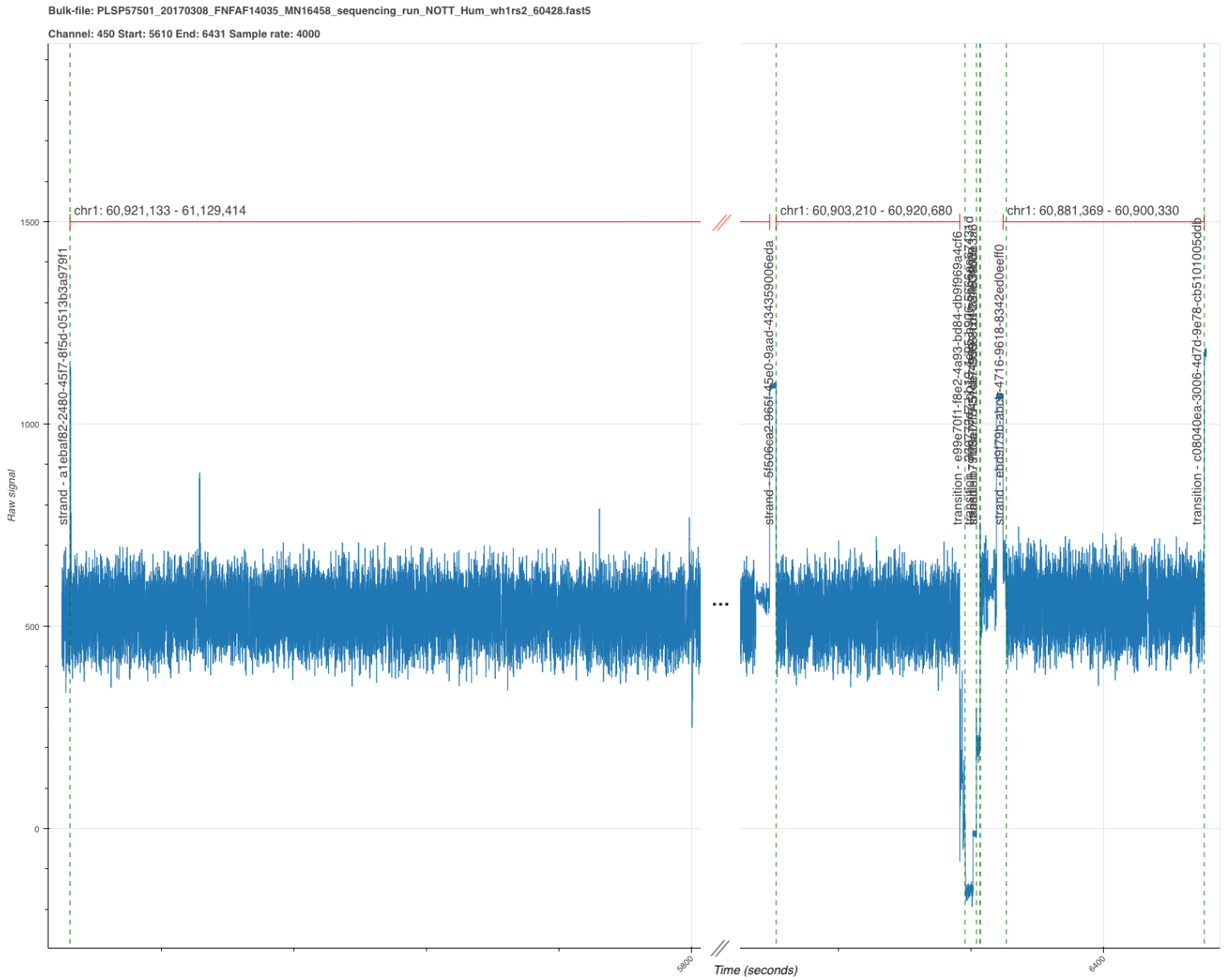


Figure A.1: Cropped view of Figure 3.4 showing the initial sequencing followed by a new "strand" classification then "transition". Each new "strand" classification is a segmented read as determined by MinKNOW.

Appendix B

Applications of readfish

Table B.1: COSMIC panel run summary statistics from NanoStat for experiment "ml_032" both the run before the nuclease flush and reload (Run 1) and after (Run 2).

	Run 1		
	Sequenced	Unblocked	Complete run
Active channels:	510.0	510.0	510.0
Mean read length:	6,290.8	516.7	795.7
Mean read quality:	9.7	11.2	11.1
Median read length:	3,728.0	433.0	439.0
Median read quality:	11.0	11.5	11.5
Number of reads:	224,821.0	4,428,557.0	4,653,378.0
Read length N50:	11,691.0	509.0	941.0
Total bases:	1,414,312,427.0	2,288,331,832.0	3,702,644,259.0
	Run 2		
Active channels:	476.0	479.0	479.0
Mean read length:	5,606.0	499.6	747.5
Mean read quality:	9.6	10.9	10.8
Median read length:	2,845.0	419.0	425.0
Median read quality:	10.6	11.2	11.2
Number of reads:	411,317.0	8,058,730.0	8,470,047.0
Read length N50:	10,891.0	496.0	820.0
Total bases:	2,305,826,485.0	4,025,766,070.0	6,331,592,555.0
	Complete Run		
Active channels:	511.0	511.0	511.0
Mean read length:	5,848.0	505.6	764.6
Mean read quality:	9.6	11.0	10.9
Median read length:	3,098.0	424.0	430.0
Median read quality:	10.7	11.3	11.3
Number of reads:	636,138.0	12,487,287.0	13,123,425.0
Read length N50:	11,191.0	501.0	855.0
STDEV read length:	6,730.2	334.0	1,902.2
Total bases:	3,720,138,912.0	6,314,097,902.0	10,034,236,814.0

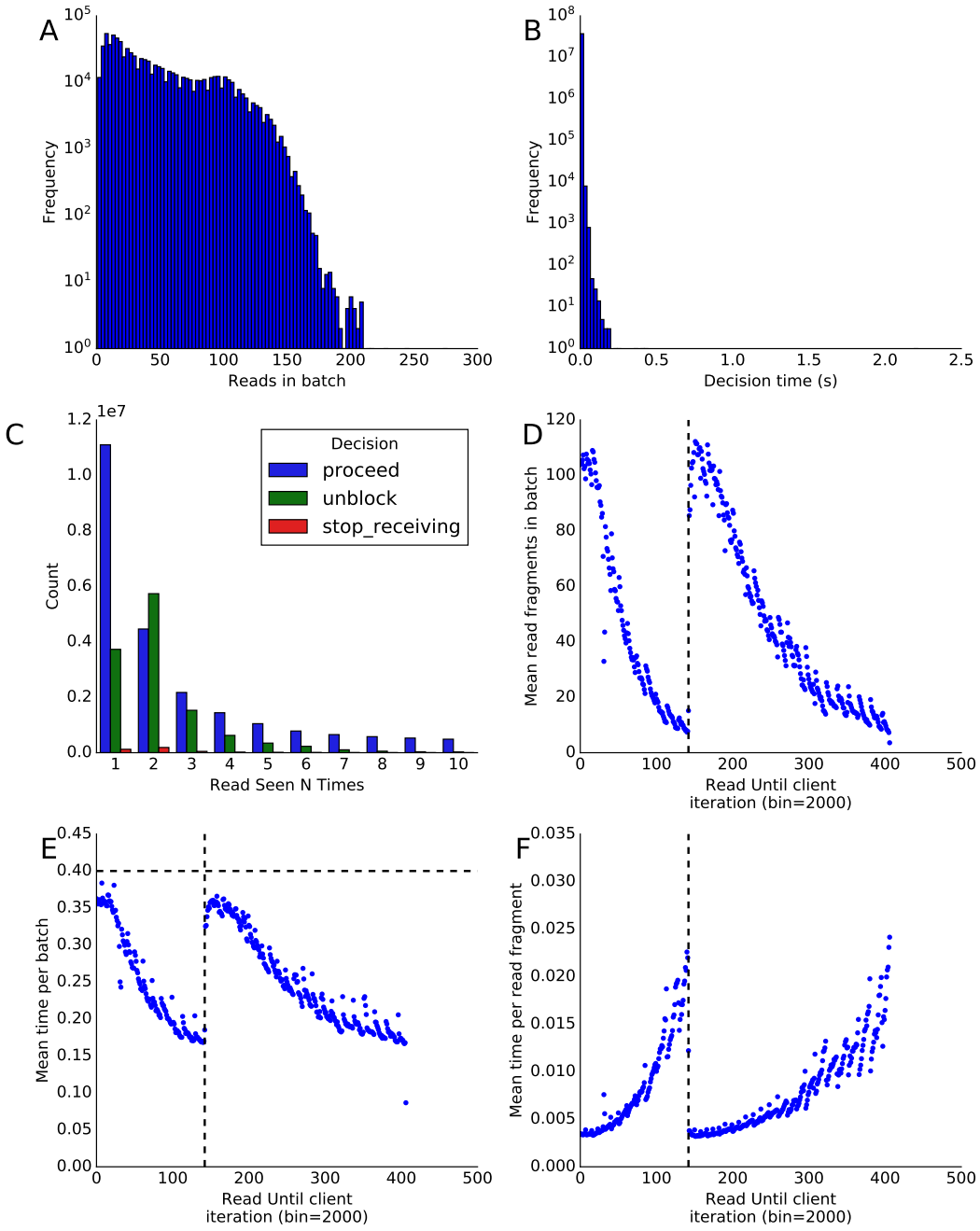


Figure B.1: COSMIC panel Run 1 (a) Histogram of read batch size throughout the selective sequencing program. (b) Histogram of decision times (time to choose unblock, stop receiving, or proceed from an alignment). (c) Counts of decision classifications for read fragments seen a given number of times. (d) Mean batch size, in bins of 2000, seen throughout the selective sequencing program. (e) Mean process time, in bins of 2000, for batches of read fragments throughout the run. (f) Mean decision time per read fragment, in bins of 2000, throughout the run. As the number of reads in a batch reduces, the overhead time of calling becomes more apparent.

Table B.2: GridION MK1 CPU

	Run 1		
	Sequenced	Unblocked	All
Active channels:	512.0	512.0	512.0
Mean read length:	4,119.5	700.5	899.0
Mean read quality:	9.7	11.0	10.9
Median read length:	1,496.0	691.0	694.0
Median read quality:	10.9	11.5	11.5
Number of reads:	465,897.0	7,556,623.0	8,022,520.0
Read length N50:	8,631.0	773.0	854.0
Total bases:	1,919,276,876.0	5,293,311,213.0	7,212,588,089.0
	Run 2		
Active channels:	492.0	492.0	494.0
Mean read length:	3,781.9	653.3	823.0
Mean read quality:	9.4	10.6	10.6
Median read length:	655.0	663.0	663.0
Median read quality:	10.4	11.2	11.1
Number of reads:	318,620.0	5,553,384.0	5,872,004.0
Read length N50:	8,621.0	737.0	799.0
Total bases:	1,204,990,366.0	3,627,816,134.0	4,832,806,500.0
	Run 3		
Active channels:	477.0	467.0	480.0
Mean read length:	3,294.0	632.0	812.0
Mean read quality:	8.8	10.1	10.0
Median read length:	575.0	627.0	625.0
Median read quality:	9.1	10.6	10.5
Number of reads:	341,022.0	4,702,331.0	5,043,353.0
Read length N50:	8,407.0	712.0	787.0
Total bases:	1,123,339,604.0	2,971,970,803.0	4,095,310,407.0
	Complete Run		
Active channels:	512.0	512.0	512.0
Mean read length:	3,773.8	667.7	852.3
Mean read quality:	9.4	10.6	10.6
Median read length:	729.0	665.0	665.0
Median read quality:	10.2	11.2	11.2
Number of reads:	1,125,539.0	17,812,338.0	18,937,877.0
Read length N50:	8,572.0	745.0	819.0
Total bases:	4,247,606,846.0	11,893,098,150.0	16,140,704,996.0

Table B.3: Linux GPU

	Complete Run		
	Sequenced	Unblocked	All
Active channels:	512.0	512.0	512.0
Mean read length:	4,792.7	486.8	711.5
Mean read quality:	9.2	10.0	9.9
Median read length:	3,981.0	402.0	407.0
Median read quality:	10.2	10.3	10.3
Number of reads:	491,653.0	8,931,842.0	9,423,495.0
Read length N50:	8,180.0	464.0	799.0
Total bases:	2,356,363,784.0	4,348,280,265.0	6,704,644,049.0

Table B.4: Linux CPU

	Run 1		
	Sequenced	Unblocked	All
Active channels:	503.0	503.0	503.0
Mean read length:	2,935.8	638.1	777.3
Mean read quality:	8.7	10.4	10.3
Median read length:	450.0	575.0	572.0
Median read quality:	9.4	10.9	10.9
Number of reads:	415,860.0	6,448,876.0	6,864,736.0
Read length N50:	8,203.0	724.0	799.0
Total bases:	1,220,873,524.0	4,114,773,085.0	5,335,646,609.0
	Run 2		
Active channels:	485.0	488.0	488.0
Mean read length:	2,584.7	614.1	740.8
Mean read quality:	8.5	10.4	10.3
Median read length:	391.0	572.0	565.0
Median read quality:	9.1	10.9	10.8
Number of reads:	517,357.0	7,528,276.0	8,045,633.0
Read length N50:	7,966.0	701.0	762.0
Total bases:	1,337,218,403.0	4,623,312,751.0	5,960,531,154.0
	Run 3		
Active channels:	465.0	469.0	470.0
Mean read length:	2,051.3	626.3	732.9
Mean read quality:	7.9	9.9	9.7
Median read length:	384.0	601.0	586.0
Median read quality:	7.8	10.2	10.1
Number of reads:	446,538.0	5,525,462.0	5,972,000.0
Read length N50:	7,631.0	713.0	764.0
Total bases:	915,995,052.0	3,460,688,401.0	4,376,683,453.0
	Complete Run		
Active channels:	503.0	504.0	504.0
Mean read length:	2,517.9	625.5	750.5
Mean read quality:	8.4	10.3	10.1
Median read length:	402.0	581.0	573.0
Median read quality:	8.7	10.7	10.7
Number of reads:	1,379,755.0	19,502,614.0	20,882,369.0
Read length N50:	7,966.0	711.0	774.0
Total bases:	3,474,086,979.0	12,198,774,237.0	15,672,861,216.0

Table B.5: MacOS run

	Run 1		
	Sequenced	Unblocked	All
Active channels:	509.0	508.0	509.0
Mean read length:	3,185.3	943.9	1,168.6
Mean read quality:	10.3	12.1	12.0
Median read length:	1,028.0	817.0	823.0
Median read quality:	10.8	12.5	12.4
Number of reads:	88,424.0	793,565.0	881,989.0
Read length N50:	9,740.0	945.0	1,121.0
Total bases:	281,654,228.0	749,023,928.0	1,030,678,156.0
	Run 2		
Active channels:	481.0	481.0	482.0
Mean read length:	4,264.3	882.7	1,124.0
Mean read quality:	10.8	12.1	12.0
Median read length:	1,290.0	778.0	785.0
Median read quality:	11.5	12.4	12.4
Number of reads:	65,642.0	854,358.0	920,000.0
Read length N50:	10,703.0	881.0	1,044.0
Total bases:	279,915,896.0	754,181,345.0	1,034,097,241.0
	Run 3		
Active channels:	463.0	466.0	470.0
Mean read length:	4,012.6	909.2	1,138.7
Mean read quality:	10.1	11.8	11.7
Median read length:	1,160.0	771.0	778.0
Median read quality:	10.9	12.2	12.1
Number of reads:	195,438.0	2,446,511.0	2,641,949.0
Read length N50:	10,663.0	897.0	1,123.0
Total bases:	784,210,007.0	2,224,303,806.0	3,008,513,813.0
	Run 4		
Active channels:	440.0	439.0	441.0
Mean read length:	2,426.0	925.6	1,102.2
Mean read quality:	9.5	11.8	11.5
Median read length:	826.0	810.0	810.0
Median read quality:	9.6	12.1	11.9
Number of reads:	34,846.0	261,154.0	296,000.0
Read length N50:	9,208.0	914.0	1,042.0
Total bases:	84,534,921.0	241,729,008.0	326,263,929.0
	Run 5		
Active channels:	439.0	441.0	444.0
Mean read length:	3,252.0	978.4	1,181.5
Mean read quality:	10.0	11.9	11.8
Median read length:	988.0	851.0	855.0
Median read quality:	10.5	12.3	12.2
Number of reads:	66,101.0	673,899.0	740,000.0
Read length N50:	10,314.0	972.0	1,129.0
Total bases:	214,961,204.0	659,339,653.0	874,300,857.0

Table B.5 continued

	Run 6		
	Sequenced	Unblocked	All
Active channels:	423.0	422.0	432.0
Mean read length:	3,724.2	863.3	1,078.7
Mean read quality:	10.0	11.7	11.6
Median read length:	1,035.0	721.0	728.0
Median read quality:	10.6	12.1	12.0
Number of reads:	213,365.0	2,621,139.0	2,834,504.0
Read length N50:	10,482.0	860.0	1,138.0
Total bases:	794,621,738.0	2,262,838,406.0	3,057,460,144.0
	Run 7		
Active channels:	397.0	394.0	400.0
Mean read length:	2,510.7	891.2	1,078.1
Mean read quality:	9.5	11.7	11.4
Median read length:	829.0	768.0	770.0
Median read quality:	9.7	12.0	11.8
Number of reads:	30,306.0	232,262.0	262,568.0
Read length N50:	9,308.0	878.0	1,040.0
Total bases:	76,088,740.0	206,982,496.0	283,071,236.0
	Run 8		
Active channels:	434.0	437.0	439.0
Mean read length:	3,564.9	855.5	1,071.7
Mean read quality:	10.1	11.8	11.7
Median read length:	979.0	711.0	717.0
Median read quality:	10.6	12.2	12.1
Number of reads:	231,852.0	2,674,351.0	2,906,203.0
Read length N50:	10,594.0	856.0	1,149.0
Total bases:	826,521,815.0	2,287,955,988.0	3,114,477,803.0
	Run 9		
Active channels:	415.0	409.0	418.0
Mean read length:	3,094.2	903.9	1,100.5
Mean read quality:	9.7	11.6	11.5
Median read length:	955.0	744.0	749.0
Median read quality:	10.1	12.0	11.9
Number of reads:	184,710.0	1,872,800.0	2,057,510.0
Read length N50:	9,936.0	921.0	1,210.0
Total bases:	571,528,135.0	1,692,808,317.0	2,264,336,452.0
	Complete Run		
Active channels:	511.0	508.0	511.0
Mean read length:	3,524.0	891.3	1,107.3
Mean read quality:	10.0	11.8	11.6
Median read length:	1,021.0	753.0	759.0
Median read quality:	10.6	12.2	12.1
Number of reads:	1,110,684.0	12,430,039.0	13,540,723.0
Read length N50:	10,387.0	894.0	1,129.0
Total bases:	3,914,036,684.0	11,079,162,947.0	14,993,199,631.0

Table B.6: Windows Subsystem Linux run

	Run 1		
	Sequenced	Unblocked	All
Active channels:	507.0	507.0	507.0
Mean read length:	3,658.9	985.0	1,240.7
Mean read quality:	10.9	12.7	12.6
Median read length:	1,084.0	855.0	863.0
Median read quality:	11.8	13.0	12.9
Number of reads:	396,023.0	3,745,681.0	4,141,704.0
Read length N50:	10,616.0	1,000.0	1,157.0
Total bases:	1,449,025,461.0	3,689,642,942.0	5,138,668,403.0
	Run 2		
Active channels:	490.0	489.0	492.0
Mean read length:	3,448.7	971.5	1,215.7
Mean read quality:	10.8	12.8	12.6
Median read length:	1,032.0	839.0	845.0
Median read quality:	11.7	13.0	12.9
Number of reads:	425,554.0	3,891,638.0	4,317,192.0
Read length N50:	10,204.0	980.0	1,145.0
Total bases:	1,467,619,590.0	3,780,742,828.0	5,248,362,418.0
	Run 3		
Active channels:	465.0	462.0	467.0
Mean read length:	2,592.9	965.0	1,165.4
Mean read quality:	10.7	12.6	12.3
Median read length:	880.0	822.0	825.0
Median read quality:	11.3	12.8	12.7
Number of reads:	541,215.0	3,855,175.0	4,396,390.0
Read length N50:	8,140.0	972.0	1,130.0
Total bases:	1,403,303,224.0	3,720,290,059.0	5,123,593,283.0
	Run 4		
Active channels:	449.0	445.0	450.0
Mean read length:	2,350.5	987.1	1,175.4
Mean read quality:	10.5	12.4	12.1
Median read length:	794.0	834.0	831.0
Median read quality:	11.0	12.6	12.4
Number of reads:	346,283.0	2,160,916.0	2,507,199.0
Read length N50:	7,719.0	995.0	1,168.0
Total bases:	813,949,140.0	2,133,061,980.0	2,947,011,120.0
	Complete Run		
Active channels:	507.0	507.0	507.0
Mean read length:	3,003.9	975.9	1,201.5
Mean read quality:	10.7	12.6	12.4
Median read length:	934.0	837.0	841.0
Median read quality:	11.4	12.9	12.8
Number of reads:	1,709,075.0	13,653,410.0	15,362,485.0
Read length N50:	9,372.0	986.0	1,148.0
Total bases:	5,133,897,415.0	13,323,737,809.0	18,457,635,224.0

Table B.7: Coverage of COSMIC targets when repeated using DeepNano-Blitz (CPU) on different platforms. With the exception of a single run (Linux GPU) 99% of targets in each run have at least 15× coverage, this is an effect of the yield of this run (Table 5.3).

	mean	std	1%	50%	99%
GridION MK1 CPU	30.02	5.14	15.60	31.18	38.39
GridION MK1 GPU	31.51	5.44	17.33	32.49	40.15
Linux CPU	27.98	4.96	15.58	29.08	36.70
Linux GPU	19.19	3.19	10.00	19.70	24.81
MacBook Pro CPU	29.24	5.21	16.83	30.29	36.30
Windows CPU	34.64	6.59	18.76	35.75	42.91

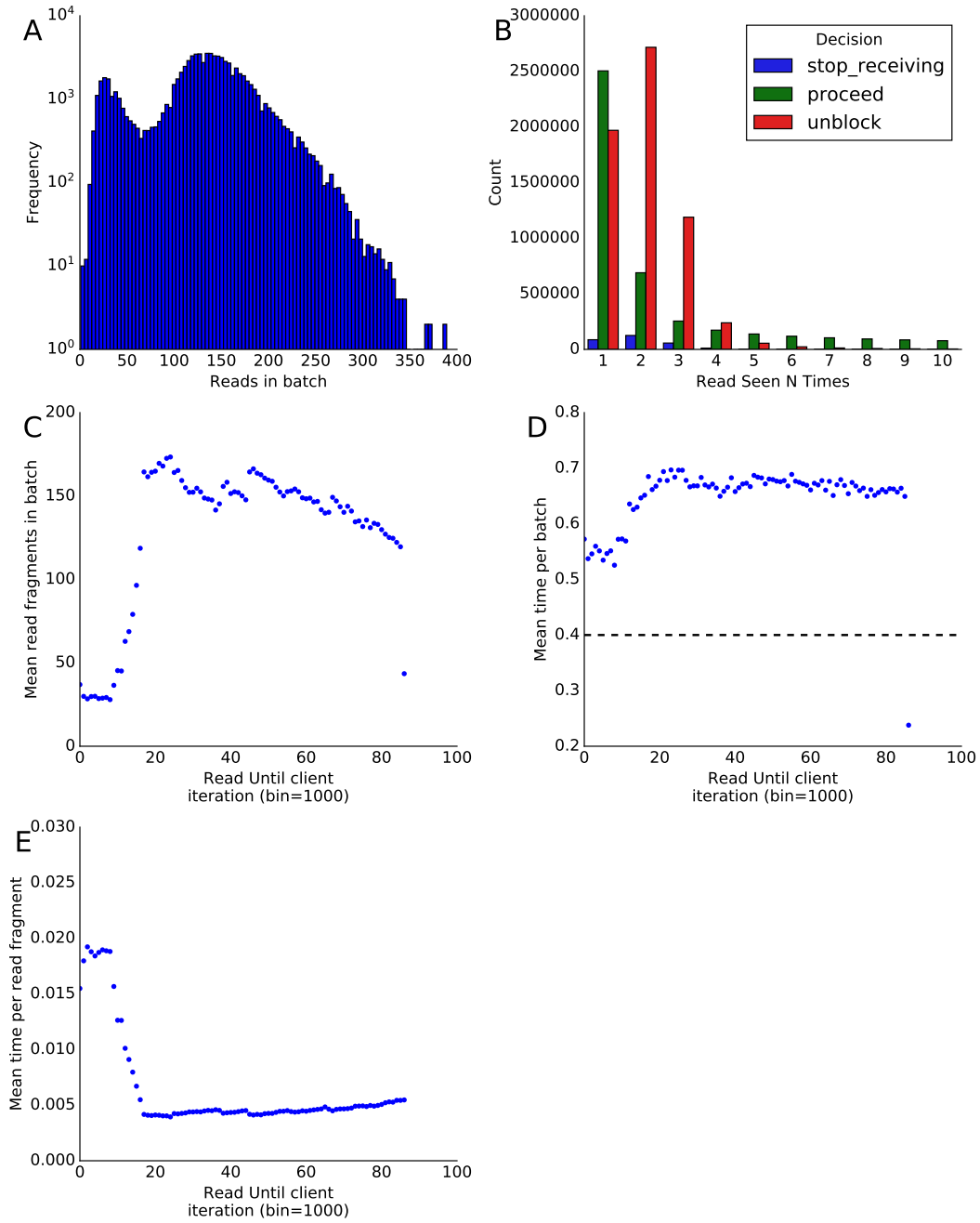


Figure B.2: iteralign is ml_007-zymo_gradual_reject_40x_hac

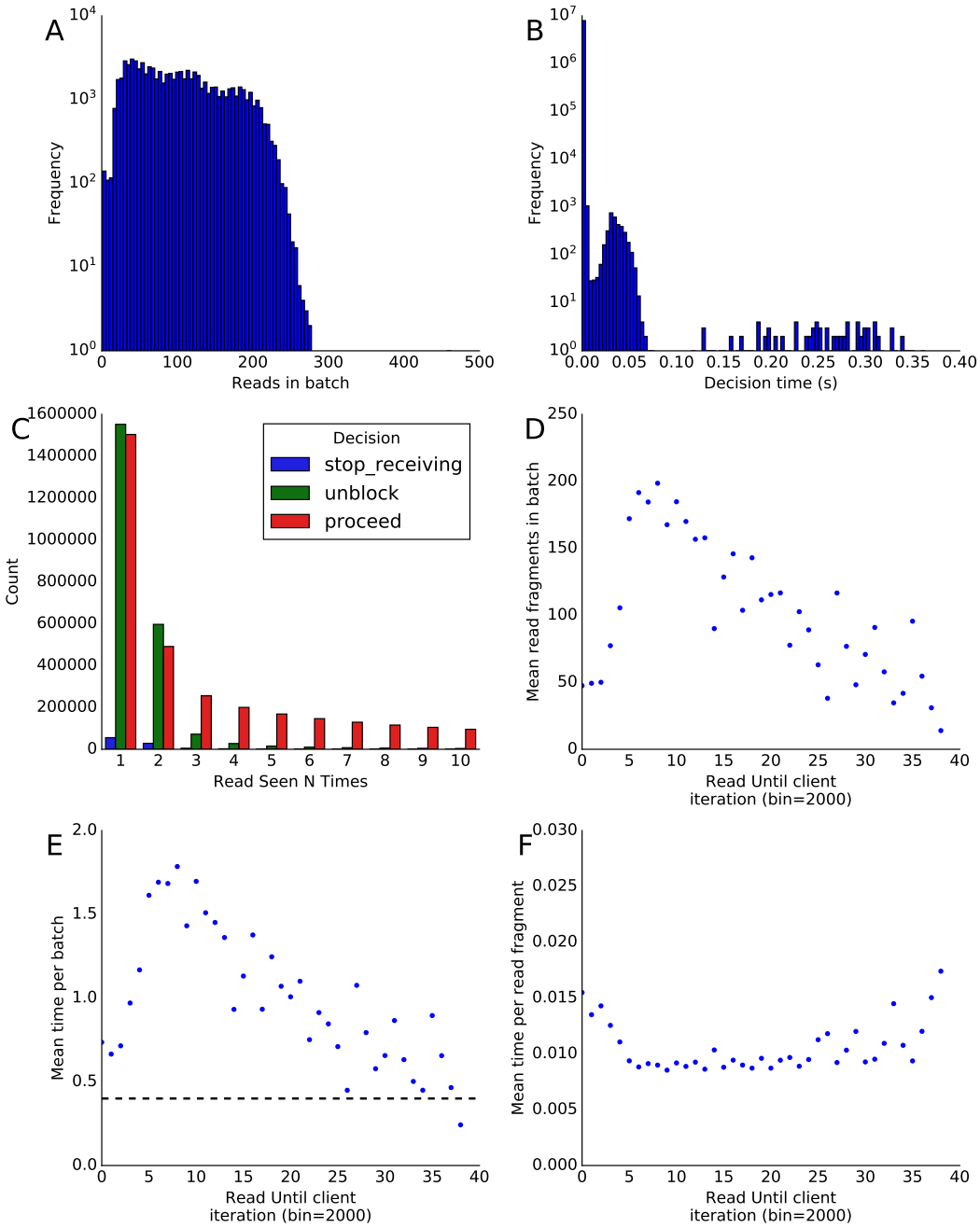


Figure B.3: readfish centrifuge run statistics

Appendix C

Submitted Papers

Bioinformatics, 35(13), 2019, 2193–2198
doi: 10.1093/bioinformatics/bty841
Advance Access Publication Date: 20 November 2018
Original Paper

OXFORD

Sequence analysis

BulkVis: a graphical viewer for Oxford nanopore bulk FAST5 files

Alexander Payne[1], Nadine Holmes[2], Vardhman Rakyan[3] and Matthew Loose[1,*]

[1]School of Life Sciences and [2]DeepSeq, School of Life Sciences, University of Nottingham, Nottingham NG7 2UH, UK and [3]The Blizard Institute, Barts and The London School of Medicine and Dentistry and Centre for Genomic Health, LSI, Queen Mary University of London, London, UK

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on June 1, 2018; revised on September 6, 2018; editorial decision on September 25, 2018; accepted on October 12, 2018

Abstract

Motivation: The Oxford Nanopore Technologies (ONT) MinION is used for sequencing a wide variety of sample types with diverse methods of sample extraction. Nanopore sequencers output FAST5 files containing signal data subsequently base called to FASTQ format. Optionally, ONT devices can collect data from all sequencing channels simultaneously in a bulk FAST5 file enabling inspection of signal in any channel at any point. We sought to visualize this signal to inspect challenging or difficult to sequence samples.

Results: The BulkVis tool can load a bulk FAST5 file and overlays MinKNOW (the software that controls ONT sequencers) classifications on the signal trace and can show mappings to a reference. Users can navigate to a channel and time or, given a FASTQ header from a read, jump to its specific position. BulkVis can export regions as Nanopore base caller compatible reads. Using BulkVis, we find long reads can be incorrectly divided by MinKNOW resulting in single DNA molecules being split into two or more reads. The longest seen to date is 2 272 580 bases in length and reported in eleven consecutive reads. We provide helper scripts that identify and reconstruct split reads given a sequencing summary file and alignment to a reference. We note that incorrect read splitting appears to vary according to input sample type and is more common in 'ultra-long' read preparations.

Availability and implementation: The software is available freely under an MIT license at https://github.com/LooseLab/bulkvis.

Contact: matt.loose@nottingham.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Oxford Nanopore Technologies (ONT) range of sequencing platforms (MinION, GridION and PromethION) utilize biological nanopores, embedded in synthetic membranes, to sequence individual single-stranded molecules of DNA (Jain *et al.*, 2015). As DNA passes through the nanopore it creates sequence specific disruptions in current flow (Ip *et al.*, 2015). The resultant reads are written to disk as soon as the DNA has translocated the pore; uniquely enabling rapid analysis of sequence data ideal for both field and clinical work (Euskirchen *et al.*, 2017; Quick *et al.*, 2016). The software controlling sequencing (MinKNOW) does this by monitoring the flow cell in real time to determine if the signal observed from each channel represents DNA. MinKNOW processes the continuous data stream from the sequencer into individual read FAST5 files containing raw signal data that are subsequently base called to reveal the sequence.The sequence of the DNA can even be analysed while the DNA is in the pore, enabling approaches such as 'Read Until' where specific molecules can be dynamically rejected according to user customisable parameters (Loose *et al.*, 2016).

Partitioning the real-time data stream into reads results in information loss about the current state before and after an individual

© The Author(s) 2018. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/article/35/13/2193/5193712 by guest on 26 January 2022

read. To better understand these events and view the effects of user intervention on sequencing when developing methods for read until or using difficult samples, we wished to visualize the entire data stream from the MinION device. ONT provide an optional bulk FAST5 file format to capture the entire data stream from every channel on the sequencing device (see http://bulkvis.readthedocs.io for guidance on how to collect a bulk FAST5 file). This file includes raw signal and metadata for every channel including MinKNOW classifications (see Supplementary Table S1). To visualize bulk FAST5 files, we developed BulkVis using the bokeh visualization package (https://bokeh.pydata.org). BulkVis annotates signal features based on the metadata and optionally mappings from a PAF file. These annotations provide a simple method to relate base called reads back to the channel and time in the data stream from which they originate and visualize their genomic location.

Whilst developing BulkVis, we observed examples of reads incorrectly segmented by MinKNOW leading to a reduction in the read lengths reported. This incorrect splitting of reads appears to correlate with read lengths such that ultra-long reads are more likely to be affected. In some cases there is no apparent reason for the read to have been split, but in many others we observe examples of reads that exhibit unusual signal patterns prior to the incorrect split.

2 Results

BulkVis scans a folder containing bulk FAST5 files at startup. An individual file is selected and specific channels plotted (Fig. 1). Basic metadata are displayed to the user. To navigate coordinates are input in the format `channel: start_time-end_time`. Alternatively pasting a FASTQ read header from a base called read will jump to its channel, time and raw signal from the bulk FAST5 file. Files can also be navigated by jumping to the next or previous instance of a specific annotation (Supplementary Table S1). Annotations are overlaid on the signal plot as vertical dashed lines, labelled with the type and associated ID if available (Fig. 1); mappings, generated by gen_bmf.py, are overlaid horizontally above the signal, with blue/red indicating forward/reverse mappings, respectively (Fig. 2). Raw signal data are proportionally smoothed to aid rapid visualization. BulkVis allows export of the signal section being viewed to a read FAST5 file compatible with Nanopore base callers. To avoid confusion with MinKNOW derived reads, BulkVis reads are custom named and include the channel number and the start and end index of the read segment recorded in samples. The read segment shown in Figure 1 results in a read FAST5 file named plsp57501_20170308_fnfaf14035_mn16458_sequencing_run_nott_hum_wh1rs2_60428_bulkvis-read_22448000-25724000_ch_450.FAST5. This region captures three single reads that when called as one read generates a 215 662 base sequence (Supplementary File Collection S1). The three individual reads base call with a combined length of 215 153 bases (Supplementary File Collection S1) and the single called read maps well to the original three (Supplementary Fig. S1a).

During library preparation, adapter sequences are added to DNA molecules such that every sequenced read should begin with an adapter sequence. MinKNOW classifies sequences in real time, usually labeling read starts with the annotation ‘adapter’. A channel without DNA in a pore will be labelled ‘pore’. Typically then adapter sequences should be detected (labelled ‘adapter’) followed by the

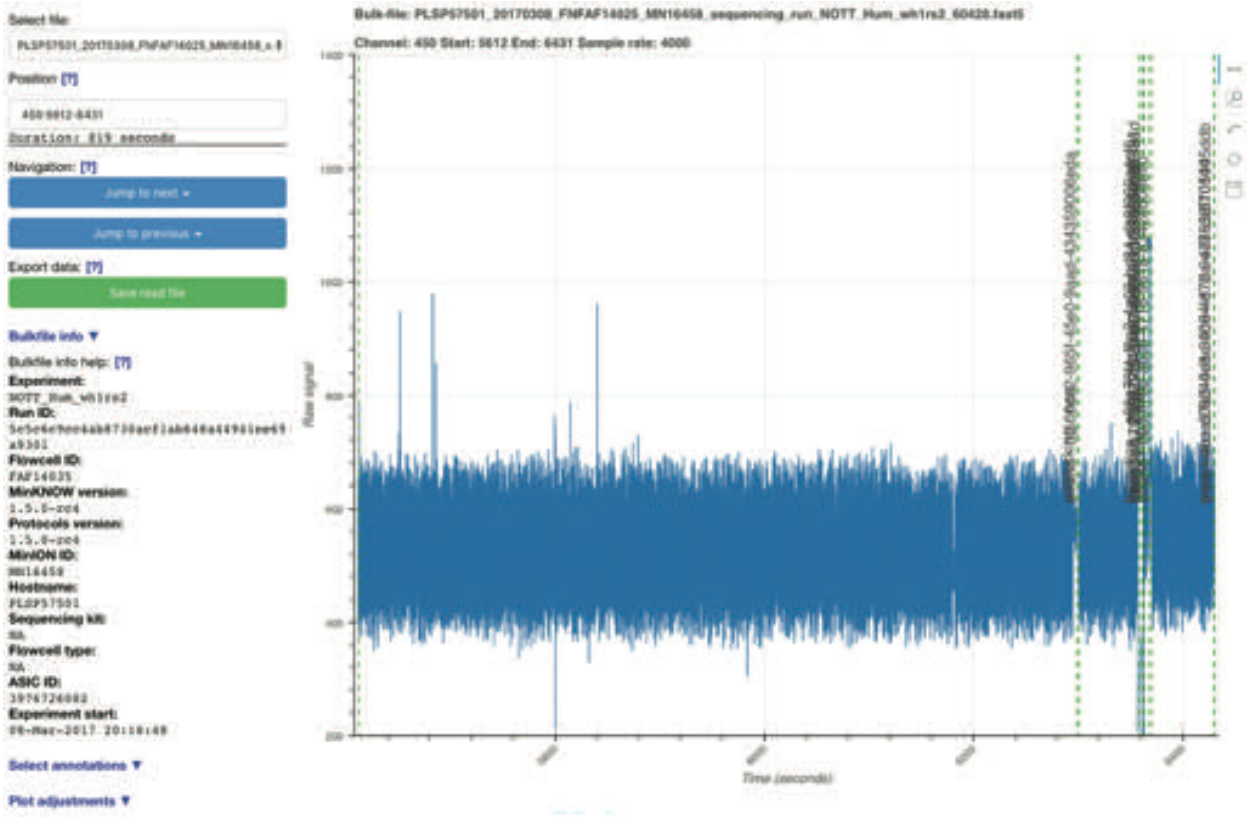


Fig. 1. Screenshot of the BulkVis application running. The vertical dashed lines indicate different annotations overlaid by MinKNOW on the signal trace in real time. The left panel provides configuration and navigation options for the selected bulk FAST5 file

Downloaded from https://academic.oup.com/bioinformatics/article/35/13/2193/5193712 by guest on 26 January 2022

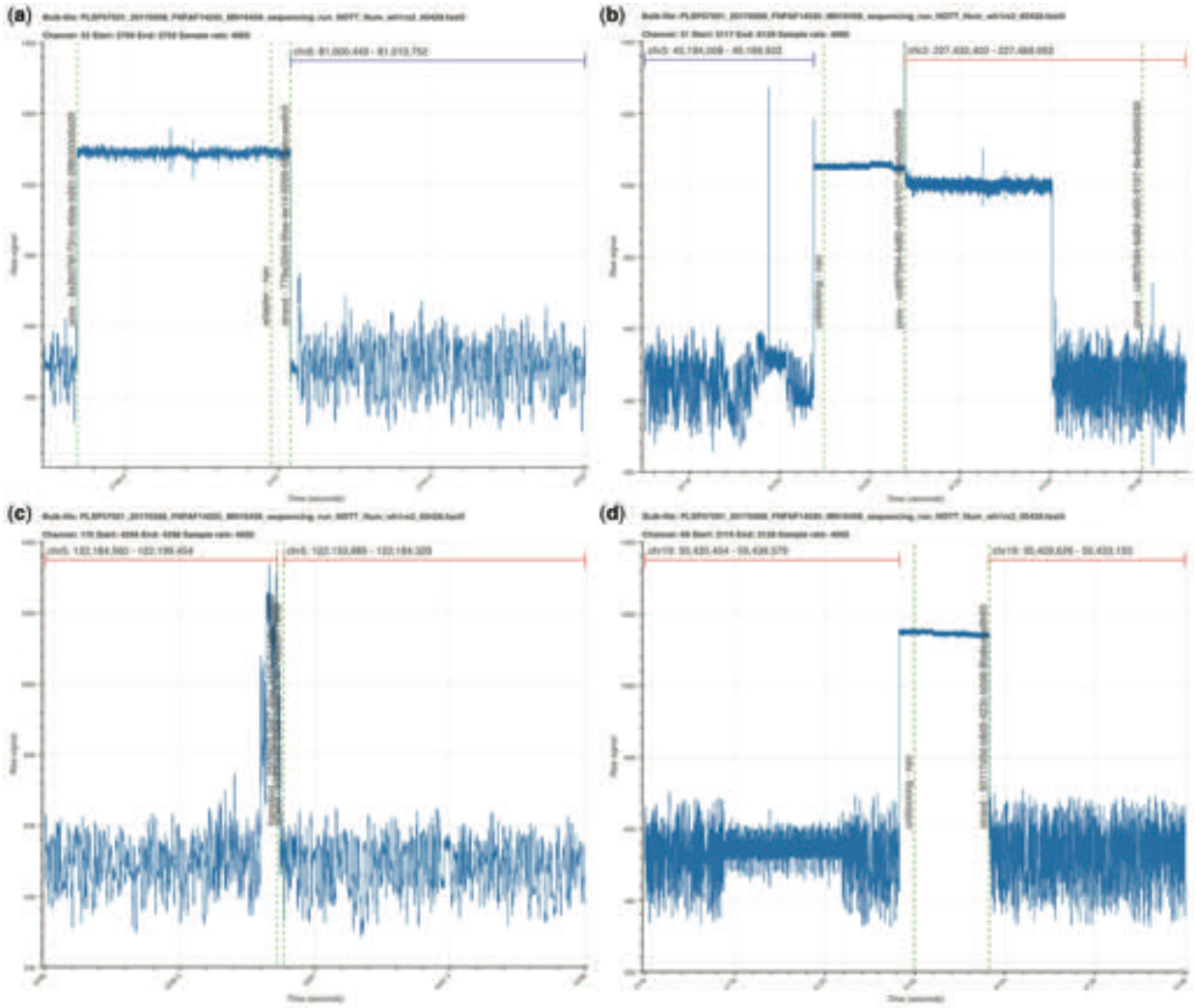


Fig. 2. Illustrative segments from a bulk FAST5 file visualized with BulkVis. (**a**) The start of a read mapping to chromosome 6. Open channel 'pore', followed by an 'adapter', and 'strand' as annotated by MinKNOW. (**b**) Read ending with an 'unblock' followed by 'pore' and then a new read. (**c**) Adjacent reads from a channel separated by unusual current patterns.These two reads are reported as distinct molecules by MinKNOW, they map consecutively to the reference. (**d**) Two adjacent reads separated by an 'unblock' signal. The unblock does not successfully remove the DNA. Instead the read continues to sequence again mapping adjacently to the reference

signal derived from the read itself ('strand') (Fig. 2a). BulkVis was developed in part to observe the effects of unblocking (the reversal of voltage across a specific channel to eject material from the pore) on DNA sequence in a nanopore. Unblocking is used in two ways; firstly the sequencer detects and removes blockages in the pore and, secondly, for the rejection of unwanted DNA in selective sequencing or 'Read Until' (Loose *et al.*, 2016). To observe the effect of an unblock (Fig. 2b) on a channel immediately after the read has been ejected users must analyse a bulk FAST5 file. Alternatively reads can be inspected in order from an individual channel. For the data presented here, unblocks have a fixed duration of 2 seconds after which the channel should return to its normal state. ONT have released an updated version of unblock, termed 'Progressive Unblock' that gradually increases the duration of the flick time (MinKNOW 2.0 Stuart Reid Pers Comm.).

During recent efforts sequencing the human genome on a MinION (Jain *et al.*, 2018), a protocol to sequence ultra-long DNA molecules was generated by Quick (2018). We used BulkVis to investigate the signal from MinKNOW during one of these runs (ASIC ID 3976726082, Supplementary Note S1). We observed reads without the expected 'pore', 'adapter', 'strand' sequence. We found 'strand' sequences separated by either 'above' and/or 'transition' (Fig. 2c) or even 'unblock' (Fig. 2d) signals without any evidence of 'pore' or 'adapter' sequences present. This was surprising given that every sequenced read should begin with an adapter. We therefore closely examined reads before and after these unusual read split events. By looking at read mappings prior and post the events shown in Figure 2c and d, we determined the two sequences were derived from adjacent positions on the same chromosome (Table 1). These reads, sequenced one after another, were most likely derived from single molecules. The alternative explanation is the chance sequencing of two independent molecules that map adjacently on the human reference, one after another, through the same pore.

Mapping all the reads (ASIC ID 3976726082) against the GRCh38 reference (Schneider *et al.*, 2017) and using read and channel numbers to sort by order through each channel we asked how many adjacent reads mapped to contiguous positions [whale_watch.py (Colloquially, Nanopore reads exceeding 1 Mb have been

Downloaded from https://academic.oup.com/bioinformatics/article/35/13/2193/5193712 by guest on 26 January 2022

Table 1. Mapping data for events shown in Figure 2c and d

	Read ID	Chan	Read	Length	Chr	Start	End
2C	7ed4aafb-d058-481c-ad60-903fd8327240	176	943	10 275	5	122 184 560	122 199 454
	83d0cea6-69ad-406b-87fb-7eaa2b178f68			43 145		122 133 985	122 184 329
2D	c13c1e73-f7e0-4ae2-8cda-729f3b4dcb79	68	758	5068	19	55 435 454	55 439 579
	50117d5d-b8d5-423c-b0d6-8fa8eaea9b65			25 596		55 409 626	55 433 153

Note: Reads mapped to GRCh38 (minimap2 -x map-ont). Combined read length (2C) is 56 284 bases, mapping to a span of 65 469 bases. Combined read length (2D) is 30 664 bases, mapping to a span of 29 953 bases. All reads here map in reverse orientation.

referred to as 'whales', with the species of whale determined by converting the length of a read in kb to a mass in kg, hence our script naming conventions.)]. About 2983 of 75 689 reads were incorrectly split with pairs of reads mapping adjacently to the reference. Stitching these reads together (using whale_merge.py) increased read length N50 from 98 876 to 103 925 bases. Mean read length of incorrectly split reads (55 190 bases) is higher than the entire dataset (23 717 bases). Re-examining previous ultra-long datasets revealed incorrect read splitting occurred 1-10% of the time (Supplementary Table S2). Incorrectly split reads had consistently higher mean read lengths than those which appear to be true single molecules. As such, these reads have significant effects on read N50 (up to 21 kb).

We generated additional ultra-long reads from the same reference human genomic DNA sample using the RAD004 transposase kit for ultra-long reads (Jain *et al.*, 2018; Quick, 2018). This revealed more incorrectly split reads with up to 30% of reads in one run affected and increases in read N50 of up to 40 kb (data not shown). Differences between runs include the input DNA, the sequencing kit, other unknown variables within the flowcells and MinKNOW software itself. Within this dataset we found a single read of 1 204 840 bases that maps to 1 325 742 bases on chromosome 5 (Fig. 3a). Remarkably, we found a set of eleven reads that, when merged, were 2 272 580 bases in length. This merged read maps to a single location in the human genome spanning 2 290 436 bases (Supplementary Table S3, Fig. 3b, Supplementary File Collection S2). Unfortunately, we did not collect a bulk FAST5 file for this run. The next longest 'fused' read caught in a bulk FAST5 file was 1 385 925 bases in length, derived from nine individual reads (Supplementary Table S4, Fig. 3c, Supplementary Fig. S2). Using BulkVis we created a single read FAST5 file from the signal covering all these reads and base called it using albacore resulting in a read that maps in its entirety to a single location in the genome.

Investigating further revealed changes in normal current flow that cause real time MinKNOW read detection to split the read. Occasionally, these events trigger unblock activity, after which the read continues to sequence from the same point in the reference (in one instance this unblock loop lasted >40 minutes, then continued to sequence the same molecule, Supplementary Fig. S3). The most complex fused read observed to date consists of 38 individual reads mapping contiguously to the genome (Fig. 3d), Supplementary Fig. S4, Supplementary File Collection S2]. The plot seen in Figure 1 (Supplementary Fig. S1B) also represents a 'fused read'. When called as a single read, the base called sequence maps contiguously to chromosome 1 from 60 882 202 to 61 129 414 bases (spanning 247 212 bases).

Analysis of a representative bulk FAST5 file identifies annotation states correlating with the starts and ends of incorrectly split reads (Fig. 4). These are either 'above' or 'transition' classifications occuring at the change from one read to the next. At lower frequency unblocks can split reads. The 'above' or 'transition' signals can be seen in the signal traces (Fig. 2). We asked if interference from surrounding channels might cause this but grouping signals from surrounding channels failed to reveal any clear pattern (not shown).

Clearly, correcting split reads should result in more contiguous assemblies. To test this, we ran our whale_merge.py analysis across the entire data set generated by the Nanopore Human Genome consortium (Jain *et al.*, 2018). This dataset consists of 16.1 million reads with an N50 of 13 kb. Running whale_watch.py across this entire dataset identifies almost 100 000 incorrectly split reads. To demonstrate the impact of split reads on assembly we identified all reads mapping to chromosome 20 and used minimap2/miniasm to assemble reads before and after correction (Li, 2016, 2018). Prior to correction, the assembly length was 52.5 Mb with an N50 of 3 699 497 bases. After correction, the assembly length increased to 55 Mb with an N50 of 4 673 412 bases, an N50 increase of just under 1 Mb.

3 Discussion

BulkVis enables visualization of bulk FAST5 files collected from Nanopore sequencers. Whilst developing BulkVis, we identified ultra-long reads can be incorrectly split by MinKNOW. This disproportionately affects ultra-long read preparations. We note that the method used for ultra-long reads is outside the normal operating conditions for nanopore sequencing (Quick, 2018). Similarly, the number of ultra-long datasets analysed in this way is limited. However, for those wishing to maximize read length the fact that adjacent reads from a single pore may represent a single molecule of DNA is significant. We have no formal explanation for why this occurs, but speculate that potential causes include DNA damage or contaminants physically linked to the DNA causing spikes in the signal. We cannot exclude the possibility that some observed split reads are caused by single strand breaks.

Additionally we note some instances where reversal of the voltage does not successfully reject a read. This effect is apparently rare and typically occurs within long reads. For applications such as selective sequencing (Loose *et al.*, 2016), reads will be rejected early in the sequencing process. We expect this will be more efficient than reads rejected midway through their length, aligning with our previous observations on 'read until' (Loose *et al.*, 2016). Whilst it is possible to determine the length of a read that is not rejected from a pore, it is impossible to measure the true length of reads that are successfully rejected. When running read until, reads that do not successfully unblock can be identified using the whale_watch.py script as they will appear as fused reads.

We provide helper scripts identifying candidate incorrectly split reads. These scripts are limited as they rely on suitable reference genomes to map against. It is possible to recognize candidate reads by close analysis of bulk FAST5 files although we anticipate MinKNOW itself can be further optimized to avoid incorrectly split

Downloaded from https://academic.oup.com/bioinformatics/article/35/13/2193/5193712 by guest on 26 January 2022

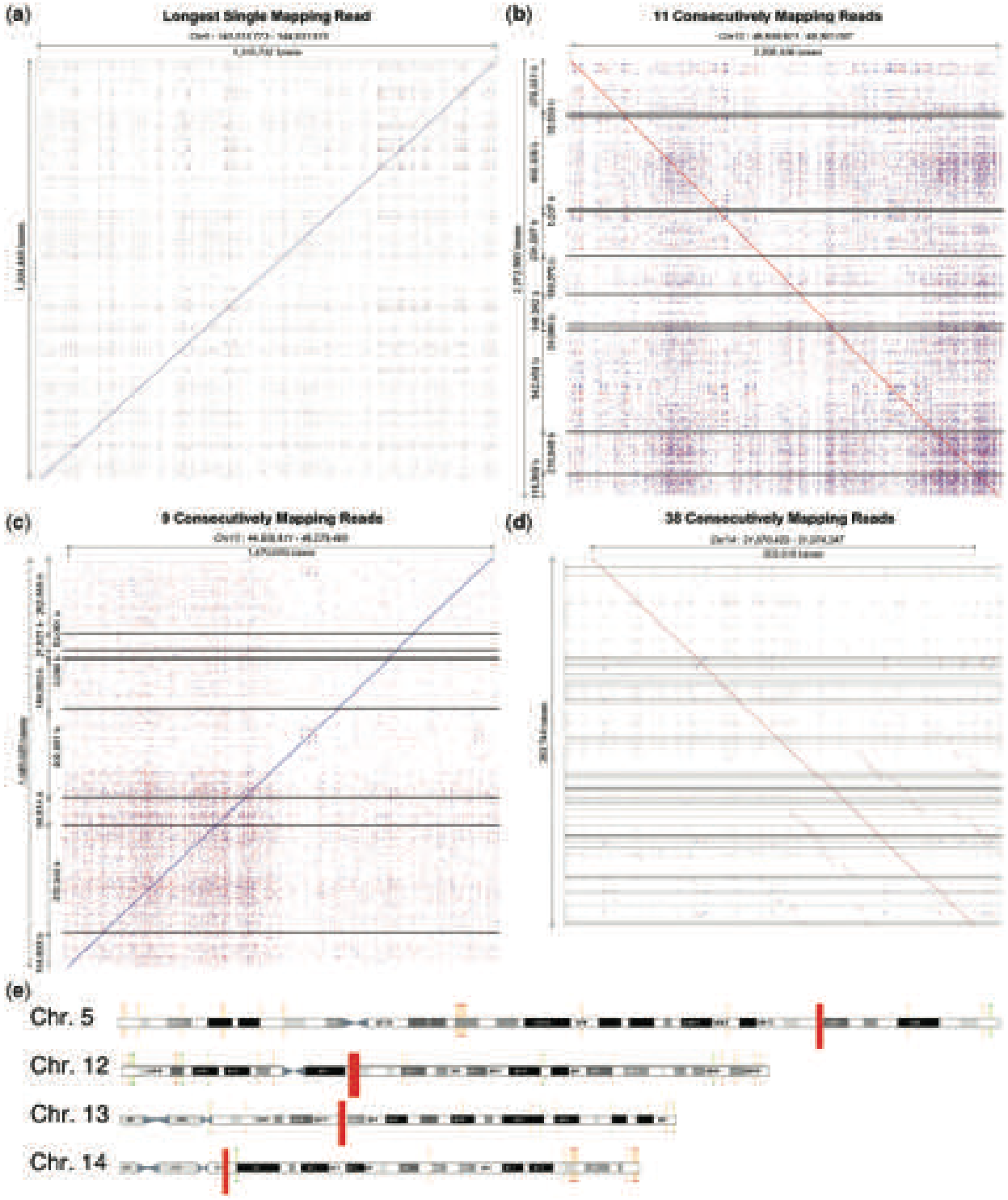


Fig. 3. Read mappings. (**a**) Longest single read. (**b**) Longest fused read (>2 Mb), sequenced in 11 reads. (**c**) Longest fused read sequenced with a bulk FAST5 file. (**d**) Fused read comprising 38 individual sequences. (a–d) Reads mapped and visualized with last (-m 1) and last-dotplot (Kiełbasa *et al.*, 2011). Horizontal lines indicate breaks between individual reads. (**e**) Illustration of reads, shown as red rectangles, from A to D mapped against GRCh38 in ENSEMBL

reads. These optimizations highlight the tension between under splitting reads, leading to chimeras (White *et al.*, 2017) versus over splitting resulting in artificially shortened reads. For general use, over splitting is clearly preferential to chimeras. However for *de novo* assembly and maximizing long reads users should be aware that decisions made by MinKNOW may not be correct. In future identifying candidate incorrectly split reads from the absence of adapter sequences might be of benefit.

Whilst we see no requirement for routine collection of bulk FAST5 files, those interested in *de novo* assembly may benefit from

Downloaded from https://academic.oup.com/bioinformatics/article/35/13/2193/5193712 by guest on 26 January 2022

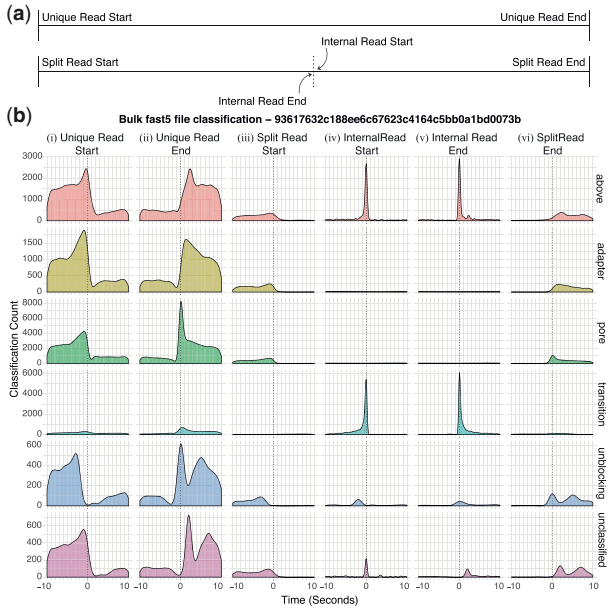


Fig. 4. MinKNOW Classifications. Here we show selected classifications (see Supplementary Table S1 for classification definitions) captured from an entire bulk FAST5 file. (**a**) Shows the labels used for reads. Unique Read Starts and Split Read Starts are genuine new molecules being sequenced. Unique Read Ends and Split Read Ends are the real end of a read. Internal Read End and Start refers to just those incorrectly split reads. (**b**) Shows the density of each selected MinKNOW classification in a 10 second window before and after each of these read labels

these files. BulkVis is provided for the visual inspection of challenging or difficult to sequence samples or where the user wishes to investigate specific events during a run. In these instances analysis of a bulk FAST5 file may provide some visual indication of the underlying issues. We note that we have seen evidence of incorrect read splitting by MinKNOW across all current versions of MinKNOW and all Nanopore platforms including MinION, GridION and PromethION.

4 Materials and methods

4.1 Sequencing

Sequencing using high molecular weight DNA extracted and prepared as previously described (Jain *et al.*, 2018; Quick, 2018). RAD002 datasets are as described in Jain *et al.* (2018). RAD004 sequencing was performed using MinKNOW version 1.11.5. Standard MinKNOW running scripts were used with manual restarting to maximize the number of sequencing channels.

4.2 BulkVis installation and operation

BulkVis and companion scripts are available on github (https://www.github.com/LooseLab/bulkvis). Scripts make use of the python modules: NumPy (Oliphant, 2015), Pandas (McKinney, 2010), bokeh (https://bokeh.pydata.org) and h5py (Collette, 2013). Full instructions and documentation are provided at http://bulkvis.readthedocs.io.

Acknowledgements

The authors thank DeepSeq at Nottingham, Nick Loman, Josh Quick, Jared Simpson and John Tyson for helpful discussions and advice as well as Graham Hall (Oxford Nanopore Technologies) for insights into unblocking behaviour.

Funding

This work was supported by the Wellcome Trust [grant number 204843/Z/16/Z]; and the Biotechnology and Biological Sciences Research Council [grant number BB/N017099/1, BB/M020061/1, BB/M008770/1, 1949454].

Conflict of Interest: ML was a member of the MinION access program and has received free flow cells and sequencing reagents in the past. ML has received reimbursement for travel, accommodation and conference fees to speak at events organized by Oxford Nanopore Technologies.

References

Collette,A. (2013) Python and HDF5. O'Reilly Media, Incorporated.

Euskirchen,P. *et al.* (2017) Same-day genomic and epigenomic diagnosis of brain tumors using real-time nanopore sequencing. *Acta Neuropathol.*, **134**, 691–703.

Ip,C.L. *et al.* (2015) MinION Analysis and Reference Consortium: Phase 1 data release and analysis [version 1; referees: 2 approved]. *F1000Res.*, **4**, 1075.

Jain,M. *et al.* (2015) Improved data analysis for the MinION nanopore sequencer. *Nat. Methods*, **12**, 351–356.

Jain,M. *et al.* (2018) Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.*, **36**, 338–345.

Kiełbasa,S.M. *et al.* (2011) Adaptive seeds tame genomic sequence comparison. *Genome Res.*, **21**, 487–493.

Li,H. (2016) Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, **32**, 2103–2110.

Li,H. (2018) Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**, 3094–3100.

Loose,M. *et al.* (2016) Real-time selective sequencing using nanopore technology. *Nat. Methods*, **13**, 751–754.

McKinney,W. (2010). Data structures for statistical computing in python. In: van der Walt, S. and Millman, J. (eds.) *Proceedings of the 9th Python in Science Conference*, pp. 51–56.

Oliphant,T.E. (2015). *Guide to NumPy*. 2nd edn. CreateSpace Independent Publishing Platform, USA.

Quick,J. (2018) *Ultra-Long Read Sequencing Protocol for RAD004 v3 (Protocols.io.mrxc57n)*.

Quick,J. *et al.* (2016) Real-time, portable genome sequencing for ebola surveillance. *Nature*, **530**, 228–232.

Schneider,V.A. *et al.* (2017) Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Res.*, **27**, 849–864.

White,R. *et al.* (2017) Investigation of chimeric reads using the MinION. *F1000Res.*, **6**, 631.

Downloaded from https://academic.oup.com/bioinformatics/article/35/13/2193/5193712 by guest on 26 January 2022

nature methods

ARTICLES

https://doi.org/10.1038/s41592-019-0617-2

Corrected: Author Correction

Nanopore native RNA sequencing of a human poly(A) transcriptome

Rachael E. Workman[1,9], Alison D. Tang[2,3,9], Paul S. Tang[4,9], Miten Jain[2,3,9], John R. Tyson[5,9], Roham Razaghi[1,9], Philip C. Zuzarte[4], Timothy Gilpatrick[1], Alexander Payne[6], Joshua Quick[7], Norah Sadowski[1], Nadine Holmes[6], Jaqueline Goes de Jesus[7], Karen L. Jones[5], Cameron M. Soulette[2,3], Terrance P. Snutch[5], Nicholas Loman[7], Benedict Paten[2,3], Matthew Loose[6], Jared T. Simpson[4,8], Hugh E. Olsen[2,3,10], Angela N. Brooks[2,3,10], Mark Akeson[2,3,10]* and Winston Timp[1,10]*

High-throughput complementary DNA sequencing technologies have advanced our understanding of transcriptome complexity and regulation. However, these methods lose information contained in biological RNA because the copied reads are often short and modifications are not retained. We address these limitations using a native poly(A) RNA sequencing strategy developed by Oxford Nanopore Technologies. Our study generated 9.9 million aligned sequence reads for the human cell line GM12878, using thirty MinION flow cells at six institutions. These native RNA reads had a median length of 771 bases, and a maximum aligned length of over 21,000 bases. Mitochondrial poly(A) reads provided an internal measure of read-length quality. We combined these long nanopore reads with higher accuracy short-reads and annotated GM12878 promoter regions to identify 33,984 plausible RNA isoforms. We describe strategies for assessing 3′ poly(A) tail length, base modifications and transcript haplotypes.

Sequencing-by-synthesis (SBS) strategies have dominated RNA sequencing since the early 1990s[1]. They involve generation of cDNA templates by reverse transcription (RT)[2,3] coupled with PCR amplification[4]. Nanopore RNA strand sequencing has emerged as an alternative single-molecule strategy[5–7]. It differs from SBS-based platforms in that native RNA nucleotides, rather than copied DNA nucleotides, are identified as they thread through and touch a nanoscale sensor. Nanopore RNA strand sequencing shares the core features of nanopore DNA sequencing; that is, a processive helicase motor regulates movement of a bound polynucleotide driven through a protein pore by an applied voltage. As the polynucleotide advances through the nanopore in single-nucleotide steps, ionic current impedance reports on the structure and dynamics of nucleotides in or proximal to the channel as a function of time. This continuous ionic current series is converted into nucleotide sequence using an ONT neural network algorithm trained with known RNA molecules.

Here we describe sequencing and analysis of a human poly(A) transcriptome from the GM12878 cell line using the Oxford Nanopore (ONT) platform. We demonstrate that long native RNA reads allow for discovery and characterization of polyA RNA molecules that are difficult to observe using short read cDNA methods[8,9]. Data and resources are posted online at https://github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md.

Results

RNA preparation, nanopore sequencing and computational pipeline. The protocol we used to isolate and sequence native poly(A) RNA from a human B lymphocyte cell line (GM12878) is summarized in Fig. 1a and detailed in Methods. A typical ionic current trace during *TP53* mRNA translocation through a nanopore reveals key features (Fig. 1b). The ionic current readout for each poly(A) RNA strand was basecalled using Albacore version 2.1.0 (ONT).

We also performed nanopore cDNA sequencing using the identical GM12878 RNA sample and analysis pipeline, but with modified parameters that are appropriate for cDNA sequencing (Methods). Both the RNA and cDNA data were archived and used for downstream analyses (Fig. 1c).

Native poly(A) RNA sequencing statistics. Six laboratories performed five nanopore sequencing runs each (Supplementary Table 1). These 30 runs produced 13.0 million poly(A) RNA strand reads, of which 10.3 million passed quality filters (PHRED > 7). Throughput varied between 50,000 and 831,000 pass reads per flow cell, with a read N50 length of 1,334 bases, and a median length of 771 bases. Of these, 9.9 million aligned using minimap2 (ref. [10]) to the GRCh38 human genome reference. The 360,000 unaligned pass reads had a median read length of 211 bases.

We next aligned the RNA pass reads to the GENCODE v27 transcriptome reference using minimap2 (ref. [10]). The aligned reads ranged in length from 85 nucleotides (nt) (a fragment of an mRNA encoding Ribosomal Protein RPL39), to 21 kb (a messenger RNA encoding spectrin repeat containing nuclear envelope protein 2 (*SYNE2*)). A comprehensive list of the genes and isoforms can be found on GitHub and in Supplementary Tables 2 and 3, respectively.

MarginStats (version 0.1)[11] was employed to calculate percent identity and the number of matches, mismatches and indels per

[1]Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD, USA. [2]Department of Biomolecular Engineering, University of California, Santa Cruz, Santa Cruz, CA, USA. [3]UCSC Genomics Institute, University of California, Santa Cruz, USA. [4]Ontario Institute for Cancer Research, Toronto, Ontario, Canada. [5]Michael Smith Laboratories and Djavad Mowafaghian Centre for Brain Health, University of British Columbia, Vancouver, British Columbia, Canada. [6]DeepSeq, School of Life Sciences, University of Nottingham, Nottingham, UK. [7]University of Birmingham, Birmingham, UK. [8]Department of Computer Science, University of Toronto, Toronto, Ontario, Canada. [9]These authors contributed equally: R. E. Workman, A. D. Tang, P. S. Tang, M. Jain, J. R. Tyson, R. Razaghi. [10]These authors jointly supervised this work: H. E. Olsen, A. N. Brooks, M. Akeson, W. Timp. *e-mail: makeson@soe.ucsc.edu; wtimp@jhu.edu

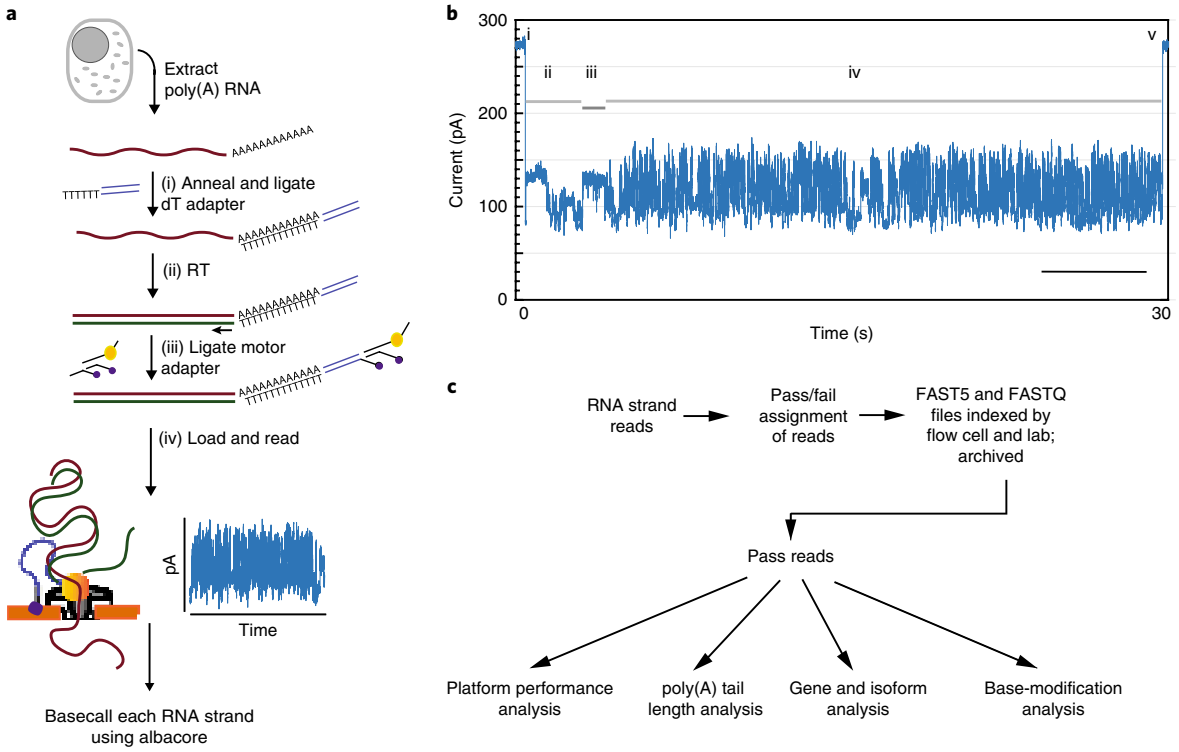


Fig. 1 | Nanopore native poly(A) RNA sequencing pipeline. a, RNA is isolated from cells followed by poly(A) selection using poly(dT) beads. Poly(A) RNA is then prepared for nanopore sequencing. **b**, A representative ionic current trace for a 2.3 kb *TP53* transcript. Ionic current components: (i) strand capture; (ii) ONT adapter translocation; (iii) poly(A) RNA tail translocation; (iv) mRNA translocation; and (v) exit of the strand into the trans compartment. Scale bar, 5 s. **c**, Processing of the RNA strand reads in silico, followed by data analysis.

aligned read in this population (Supplementary Table 4). Median identity was 86 ± 0.86% (Fig. 2a). The mismatch, insertion and deletion errors were 2.4%, 4.3% and 4.4% respectively. The basecaller seldom confused G-for-C or C-for-G (0.38% and 0.47% errors, respectively); C-to-U and U-to-C errors were substantially higher (3.62% and 2.23%, respectively) (Fig. 2b). We compared the observed read length with the expected transcript length as defined by GENCODE v27, and found general agreement (Fig. 2c). The discrete clusters below the diagonal represent incorrect assignments to GENCODE isoforms, and the diffuse shading represents fragmented RNA (see the text concerning RNA truncation).

For nanopore cDNA data, we observed a median identity of 85%, which is comparable to recent published nanopore DNA results[12]. The substitution error patterns for cDNA data were similar to those for native RNA data (data not shown).

***k*-mer coverage.** Previous analyses indicated that some nucleotide subsequences (*k*-mers) are over- or under-represented in nanopore-based DNA sequence reads[11,12]. We assessed nanopore RNA and cDNA 5-mer coverage using reads aligned to GENCODE v27 isoforms. Only reads that covered 90% or more of a given reference sequence were chosen; this selected 2.9 million of the total 10.3 million RNA reads. Of the 15.1 million pass cDNA reads, 3.9 million pass cDNA reads were selected. These reads included all 1,024 possible 5-mers (see Supplementary Fig. 1a,b for normalized native RNA and cDNA counts, respectively).

The 5-mers that were under-represented in native RNA and over-represented in cDNA are shown in Supplementary Tables 5 and 6, respectively. Similar to previous studies[11,12], the largest deviation from expectation occurred for homopolymer-rich *k*-mers.

Nanopore sequencing performance assessed using mitochondrially encoded RNA. We reasoned that mitochondrial (MT) poly(A) transcripts could be used to benchmark nanopore sequencing performance because they are abundant in all human cells, are single exon, and vary substantially in length (349–2,379 nt). Approximately 10% (950,879) of reads aligned to the mitochondrial genome (Fig. 3a and public UCSC track (http://genome.ucsc.edu/s/miten/nvRNA_f_r). As expected, most of these poly(A) transcripts corresponded to mitochondrial ribosomal RNA or to mitochondrial mRNA. Overall, the nanopore RNA reads recapitulated known features of the human MT-transcriptome (Supplementary Figs. 2 and 3). We also observed poly(A) RNA strands that are difficult to observe by conventional means (Supplementary Figs. 4 and 5).

Mitochondrial RNA (MT-RNA) read length analysis was revealing. 5,000 reads aligned to mitochondrially encoded cytochrome c oxidase II (*MT-CO2*) or to mitochondrially encoded NADH:ubiquinone oxidoreductase core subunit 4L (*MT-ND4L*) and *MT-ND4* genes combined (Fig. 3b). For each transcript, a dominant band corresponded closely to the expected length (732 nt and 1,673 nt for *MT-CO2* and *MT-ND4L/ND4*, respectively). However, for each of these, a population of truncated reads was randomly distributed between the dominant band and about 300 nt in length. When we quantified the fraction of truncated reads as a function of nominal transcript length for ten MT-mRNA of the heavy strand (Methods), we found a strong linear anti-correlation in most cases (Fig. 3c). The single outlier was *MT-ND5*, which is the mitochondrial transcript with a 568 nt 3′ untranslated region (UTR).

These MT-poly(A) RNA truncations could occur at any of several non-biological steps during the sequencing process, or they could arise from regulated enzymatic degradation in the mitochondrion[13]. Here we considered three possible non-biological causes that were specific to the nanopore platform.

One systematic cause of read truncations occurred because the enzyme that controls translocation through the pore is 10–15 nt from the nanopore sensor. When the enzyme releases the last nt at

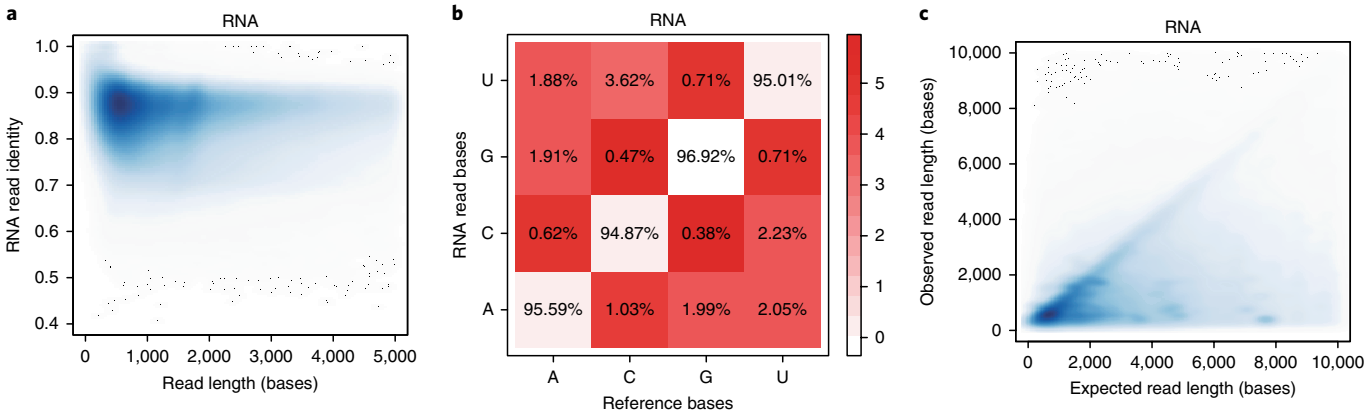


Fig. 2 | Performance statistics for nanopore native RNA sequencing. a, Alignment identity versus read length for native RNA reads. **b**, Substitution matrix for native RNA reads. The *x* axis is the known base identity for the GENCODE v27 transcriptome at positions that aligned to nanopore reads. The *y* axis is base identity at the same position for nanopore reads. The values within boxes are the percentage of times nanopore basecalls corresponded to correct (diagonal) or incorrect (red-shaded) calls according to the reference. The color intensity in the boxes represents the negative natural log probability of basecall matches or mismatches (see color key at right). **c**, Observed versus expected read length for ~9.7 million native RNA reads. The discrete clusters below the diagonal represent incorrect assignments to GENCODE isoforms, and the diffuse shading represents fragmented RNA.

the 5′ end, the strand is rapidly driven through the pore which prevents reading the terminal 10–15 nt. This phenomenon was evident by close inspection of read coverage at the 5′ end of MT-mRNA transcripts (http://genome.ucsc.edu/s/miten/nvRNA_f_r), and is expected for all direct RNA reads in the present ONT protocol.

Another possible cause was ionic current signal artifacts associated with enzyme stalls during RNA translocation, or with extraneous voltage spikes (Supplementary Fig. 6a). Similar artifacts have been shown to disrupt strand reads during MinION sequencing of DNA[14]. Systematic analysis of 2,729 *MT-CO1* reads within bulk FAST5 files from Lab 1 identified 527 reads that started or ended abnormally (Methods). By including ionic current segments that were identified before or after many of these truncations, we reconstructed 300 reads with longer alignments to *MT-CO1* (Supplementary Fig. 6 and Supplementary Table 7). This phenomenon was length dependent (Fig. 3d), ranging from 4.2% of reads with rescued segments for ND3 (346 nt nominal length) to 17.6% for ND5 (2,379 nt nominal length).

A third possible cause was strand breaks during nanopore sequencing runs. We analyzed *MT-CO1* read-length distribution for each of the six laboratories as a function of time on ONT flow cells. We found that the read frequency at all lengths declined steadily over 36 h as expected, however the full-length fraction declined by only 5% (Supplementary Fig. 7). This analysis also revealed that RNA from Lab 6 had degraded prior to the sequencing run. Therefore, isoform-level analyses (see below) focused on 8.17 million aligned poly(A) RNA reads from Labs 1–5.

Isoform detection and analysis. Long nanopore reads could improve resolution of RNA exon–exon connectivity, allowing for discovery of unannotated RNA isoforms. However, these reads averaged 14% per-read basecall errors, confounding precise determination of splice sites. Also, biological RNA processing and in vitro 5′-end truncations (see above) can make it difficult to define transcription start sites (TSS).

To overcome these limitations we employed FLAIR (full-length alternative isoform analysis of RNA, see Methods). We first replaced any nanopore-based splice sites bearing apparent sequencing errors with splice sites supported by GENCODE v27 annotations or by Illumina GM12878 cDNA data (Supplementary Fig. 8)[15,16]. Second, to overcome TSS uncertainty caused by truncated RNA reads, we considered only reads with 5′ ends proximal to promoter regions (defined by ENCODE promoter chromatin states for GM12878 (refs. [17–19])). Third, we used FLAIR to group reads into isoforms according to chains of splice junctions.

We compiled two FLAIR isoform sets (Supplementary Table 8) using different supporting read criteria (see Methods and Supplementary Fig. 9): (1) a FLAIR-sensitive set that included isoforms with three or more uniquely mapped reads (see https://github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md#analyses). This large set could be useful for isoform discovery, at the risk of false positives; (2) a FLAIR-stringent set that was compiled by filtering set (1) for isoforms having 3 or more supporting reads that spanned ≥80% of the isoform with ≥25 nt coverage into the first and last exon.

We screened for unannotated isoforms within the FLAIR-stringent dataset. Of the 33,984 isoforms representing 10,793 genes (Supplementary Table 9), 52.6% had a splice junction chain that was unannotated in GENCODE (13.0% of total assigned reads) (Fig. 4a). We observed that non-coding genes had more complex splicing patterns per gene than did coding genes (Fig. 4b), in agreement with prior studies demonstrating increased alternative splicing in non-coding exons[20,21].

As a conservative alternative to FLAIR, we compiled two GENCODE-based isoform sets using a lower coverage threshold because GENCODE is curated (Supplementary Table 8): (1) a GENCODE-sensitive set that included isoforms with 1 or more reads that mapped uniquely to GENCODE v27; (2) a GENCODE-stringent set that was compiled by filtering set (1) for isoforms having 1 or more supporting reads that spanned ≥80% of the isoform with ≥25 nt coverage into the first and last exon.

To estimate the sequencing depth required to completely characterize the GM12878 transcriptome, we plotted the number of isoforms detected in the GENCODE-sensitive and FLAIR-stringent isoform sets versus the number of subsampled reads in 10% increments. We then fitted a hyperbolic function to the data (Fig. 4c, Supplementary Fig. 10 and Supplementary Table 10). It is evident that the curves did not saturate and that additional reads would be required to capture a complete GM12878 transcriptome.

Assignment of transcripts to parental alleles. Allele-specific expression (ASE) is the preferential transcription of RNA from the paternal or maternal copy of a gene. Although the importance of this phenomenon has been characterized[22], the consequences are

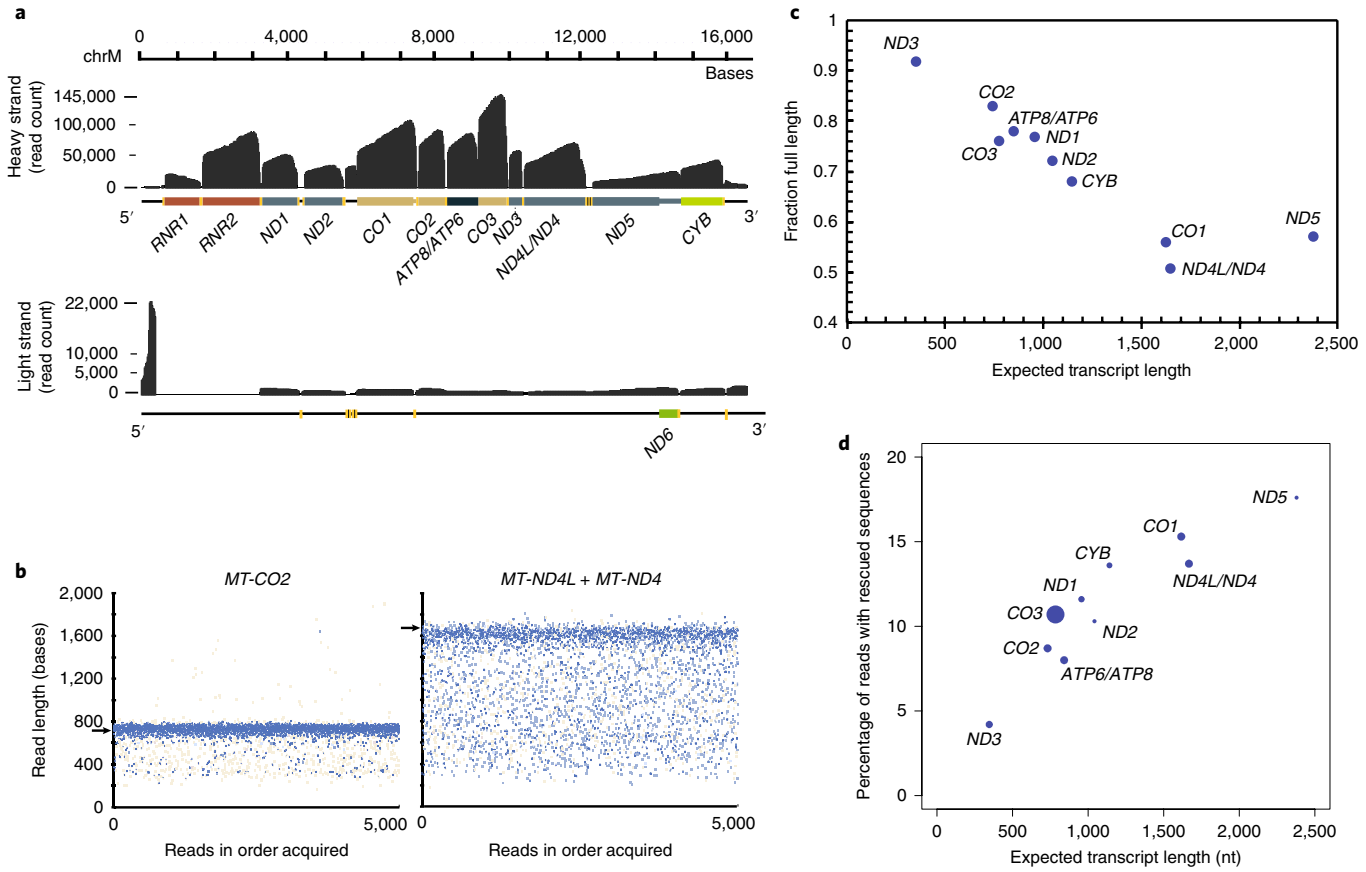


Fig. 3 | Mitochondrially encoded poly(A) RNA transcripts. a, Read coverage of the H strand (top) and the L strand (bottom). Dark gray is base coverage along the MT genome. Labeled colored bars represent protein-coding genes including known UTRs, or ribosomal RNA (*RNR1, RNR2*). Text denotes specific genes without the MT prefix. Yellow bars represent transfer RNA genes. **b**, Distribution of nanopore read lengths for *MT-CO2* and *MT-ND4L + MT-ND4* transcripts. Each point represents 1 of approximately 5,000 reads in the order acquired from a single Lab 1 MinION experiment. Horizontal arrows are expected transcript read lengths. **c**, Relationship between expected transcript read length and fraction of nanopore poly(A) RNA reads that were full length. Each point is for a protein-coding transcript on the H strand. Labels are for mitochondrial genes without the MT prefix. See Methods for definition of 'Full Length'. **d**, Percent of artificially truncated strand reads where sequence was recovered from the ionic current signal. Points are for protein coding transcripts as in **c**. Dot sizes indicate relative number of reads.

not fully understood. This is partly owing to technical limitations of haplotype identification using short read sequencing technologies.

We reasoned that the long nanopore RNA reads would be easier to assign to the parental allele of origin due to the greater chance of encountering a heterozygous SNP. Reads with at least two heterozygous SNPs were assigned to the parental allele of origin using HapCUT2 (ref. [23]). To discover the most possible genes, we used the FLAIR-sensitive dataset. In it, we found 3,751 genes with at least 10 haplotype informative reads. Of these genes, 3,707 were from autosomal chromosomes and 44 were from the X chromosome (Supplementary Table 11). Among autosomal genes, 228 (6.1%) showed significant ASE (binomial test, $P<0.001$), and among X chromosome genes, 23 (95.7%) showed significant ASE (binomial test, $P<0.001$). X chromosome expression was biased, with 22 of 23 allele-specific X-linked genes originating from the maternal allele, consistent with previous results for this cell line[24]. The sole paternally expressed X-linked locus encoded the long non-coding RNA XIST (Supplementary Fig. 11), which is transcribed from the inactive X-chromosome and recruits epigenetic silencing machinery for X-inactivation in females[25]. The remaining genes were expressed equally from both parental alleles.

We combined these allele-specific reads with isoforms from the FLAIR-sensitive set to mine for allele specificity (Methods). We identified five genes with one isoform expressed from one allele and another isoform expressed from the other allele (binomial test, $P<0.001$, Supplementary Table 12). One of these genes, interferon induced with helicase C domain 1 (*IFIH1*), had a paternal isoform with exon 8 retained, while the maternal isoform did not retain exon 8 (Fig. 4d and Supplementary Fig. 12). The closest SNV used in allele-assignment was 886 nt away from the alternative splicing event in this transcript. This would be undetectable using short read sequencing.

3′ poly(A) analysis. Transcript poly(A) tails are thought to have a role in post-transcriptional regulation, including mRNA stability and translational efficiency[26]. However, these homopolymers can be several hundred nucleotides long making them difficult to measure using short-read SBS data[27,28].

We measured poly(A) tail lengths directly using a low-variance ionic current signal associated with the 3′ end of each poly(A) strand (Fig. 1b, iii). We developed a computational method ('nanopolish-polya', https://github.com/jts/nanopolish) to segment this signal and estimate how many ionic current samples were drawn from the poly(A) tail region. By correcting for the rate at which the RNA molecule passes through the pore, nanopolish-polya estimates the length of the poly(A) tail. Algorithmic details can be found in Supplementary Note 1.

To test this method, we obtained six MinION-derived poly(A) RNA control datasets generated by ONT (ENA accession

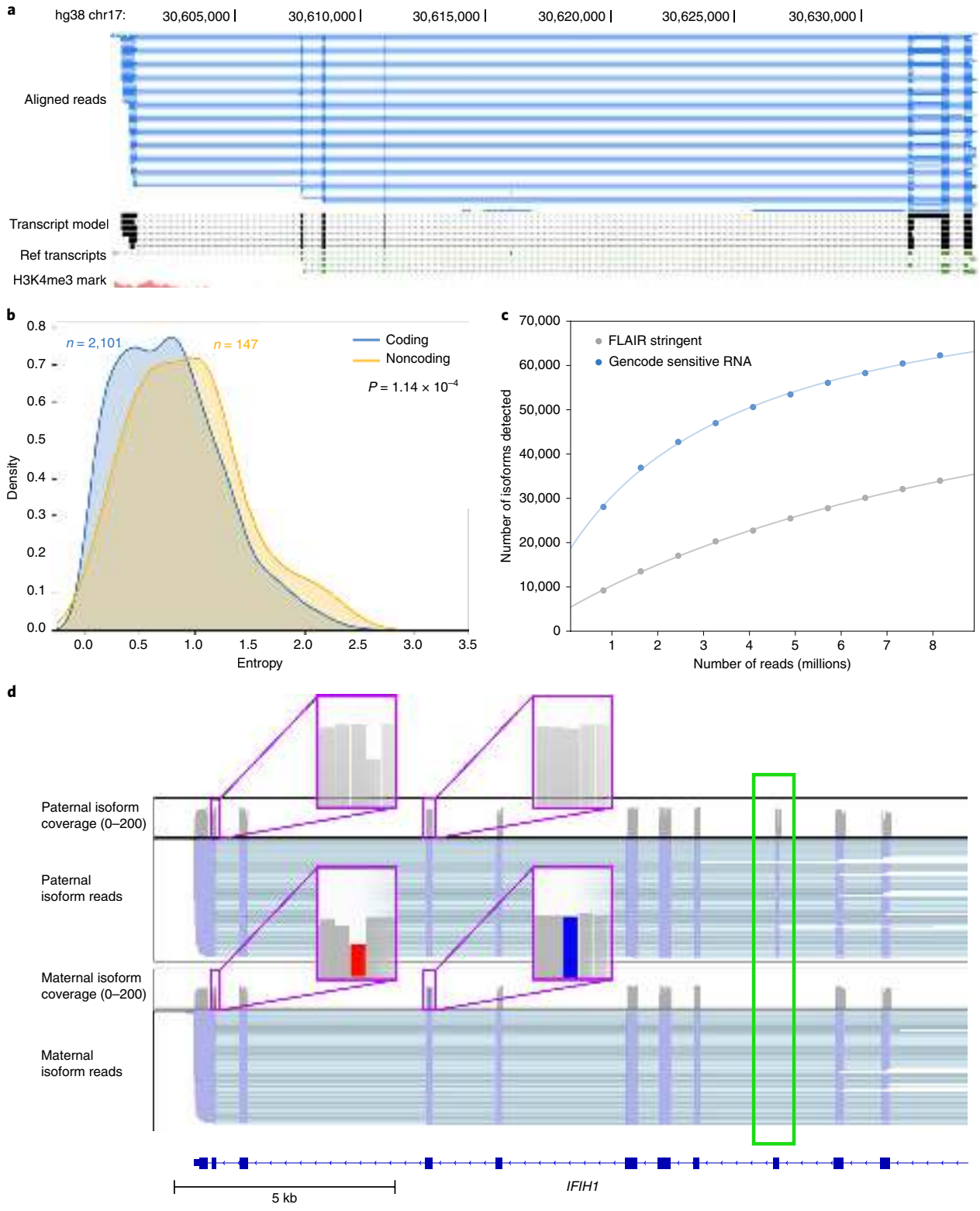


Fig. 4 | Isoform-level analysis of GM12878 native poly(A) RNA sequence reads. a, Genome browser view of unannotated isoforms that aligned to SMURF2P1-LRRC37BP1. The tracks are: a subset of the aligned native RNA reads (blue); the FLAIR-defined isoforms (black); SMURF2P1-LRRC37BP1 annotated isoforms from GENCODE v27 comprehensive set (green); transcription regulatory histone methylation marks (red). **b**, Shannon entropy of isoform expression for coding versus noncoding genes detected by FLAIR. Only genes with at least 50 reads and more than two isoforms were used. The *P* value was calculated using a Mann–Whitney U test. **c**, Saturation plot showing the number of isoforms discovered (*y* axis) versus the number of native RNA reads (*x* axis). **d**, IGV view of allele-specific isoforms for *IFIH1*. Purple boxes (insets) indicate the location of SNPs used to assign allele specificity (gray reference; red and blue SNPs). The alternatively spliced exon is indicated by a green box. The numbers in brackets indicate coverage (number of reads).

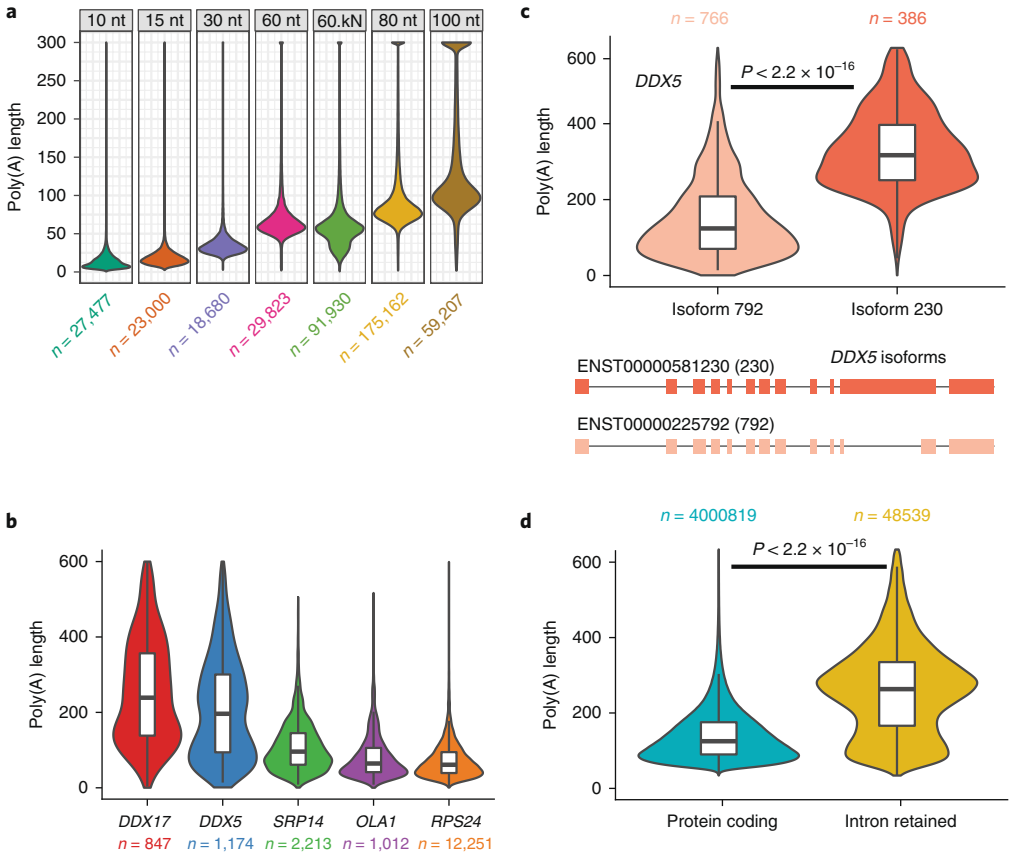


Fig. 5 | Testing and implementation of the poly(A) tail length estimator nanopolish-polya. a, Estimate of poly(A) lengths for a synthetic enolase control transcript bearing 3′ poly(A) tails of 10, 15, 30, 60, 80 or 100 nt. 60 nt-kN contained a 10-nt random sequence inserted between the enolase sequence and the 3′ poly(A) 60-mer. **b**, Violin and box plots showing poly(A) tail-length distributions for genes with the longest (*DDX5*, *DDX17*), median (*SRP14*) and shortest (*RPS24*, *OLA1*) values from a ranked list of 1,043 genes. **c**, Distribution of poly(A) tail lengths (top) and gene models (bottom) for two isoforms of *DDX5*. **d**, Distribution of poly(A) tail lengths for representative intron-retaining and intron-free transcripts identified using the GENCODE-Sensitive isoform set. Kruskal–Wallis *P* values are denoted. Each box plot shows the maximum and minimum values of the data (top and bottom lines), the third and first quartiles (edges of upper and lower boxes respectively) and the median (center line).

PRJEB28423). These datasets consisted of ionic current traces for synthetic *S. cerevisiae* enolase transcripts appended with 3′ poly(A) tails of 10, 15, 30, 60, 80 or 100 nt. A second version of the 60-nt poly(A) tailed construct (60 nt-kN) contained a 10-nt randomer between the enolase sequence and the 3′ poly(A) (Fig. 5a, Supplementary Table 13 and Supplementary Note 1).

We applied this poly(A) length estimator to the complete GM12878 native poly(A) RNA sequence dataset. Overall, the poly(A) length distribution centered at ~50 nt, with mitochondrial transcripts averaging 52 nt and almost no poly(A) tail lengths greater than 100 nt. This is consistent with results for mitochondrial poly(A) RNA from other human cell lines[29]. Conversely, nuclear transcripts showed a broader length distribution, with a peak at 58 nt, a mean of 112 nt, and a large number of poly(A) tails greater than 200 nt.

Next, we measured poly(A) tail length differences between genes with at least 500 reads and ranked 1,043 genes by median values (Fig. 5b and Supplementary Table 14). For some genes, for example the RNA-binding protein DEAD-box helicase 5 (DDX5), multiple peaks were observed (Fig. 5b), suggesting the presence of isoform-specific poly(A) tail-length sub-populations. To explore this, we analyzed genes in the GENCODE-sensitive dataset, and found 215 genes that had isoforms with significantly different poly(A) lengths (Supplementary Fig. 13).

When we compared two GENCODE isoforms of DDX5, we noted that an intron-retaining isoform (ENST00000581230, ‘230’) had a median poly(A) tail length of 327 nt, compared with the protein-coding isoform (ENST00000225792, ‘792’), which had a median poly(A) tail length of 125 nt (Fig. 5c). This difference motivated us to explore the relationship between poly(A) tail length and RNA intron-retention. We classified each isoform in GENCODE-sensitive as either protein-coding or intron-retaining. The subset of transcripts with retained introns tended to have longer poly(A) tails (median 232 nt) than did transcripts without introns (median 91 nt) (*t*-test *P* value $< 2.2 \times 10^{-16}$, Fig. 5d).

Modification detection. Nanopore sequencing has been used to identify base modifications in DNA[30,31] and RNA[5,7]. N6-methyladenine (m6A) is the most common internal modification on mRNA[32], and has been implicated in many facets of RNA metabolism[33]. m6A dysregulation has been linked to human diseases, including obesity and cancer[34]. Because m6A modifications are enriched in 3′ UTRs, with two-thirds of these containing miRNA sites[35], the impact of this modification appears to be largely regulatory, as opposed to altering protein-coding sequence.

We focused our studies on the GGACU binding motif of methyltransferase 3 (METTL3), a subunit of the m6A methyltransferase complex[36]. As an example, we compared the raw current signal at a putative m6A site (chr19:3976327) in eukaryotic elongation factor 2 (*EEF2*) with the signal for an in vitro transcribed copy (Methods). This comparison revealed an ionic current change attributable to m6A (Fig. 6a). To validate this result, we used synthetic oligomers

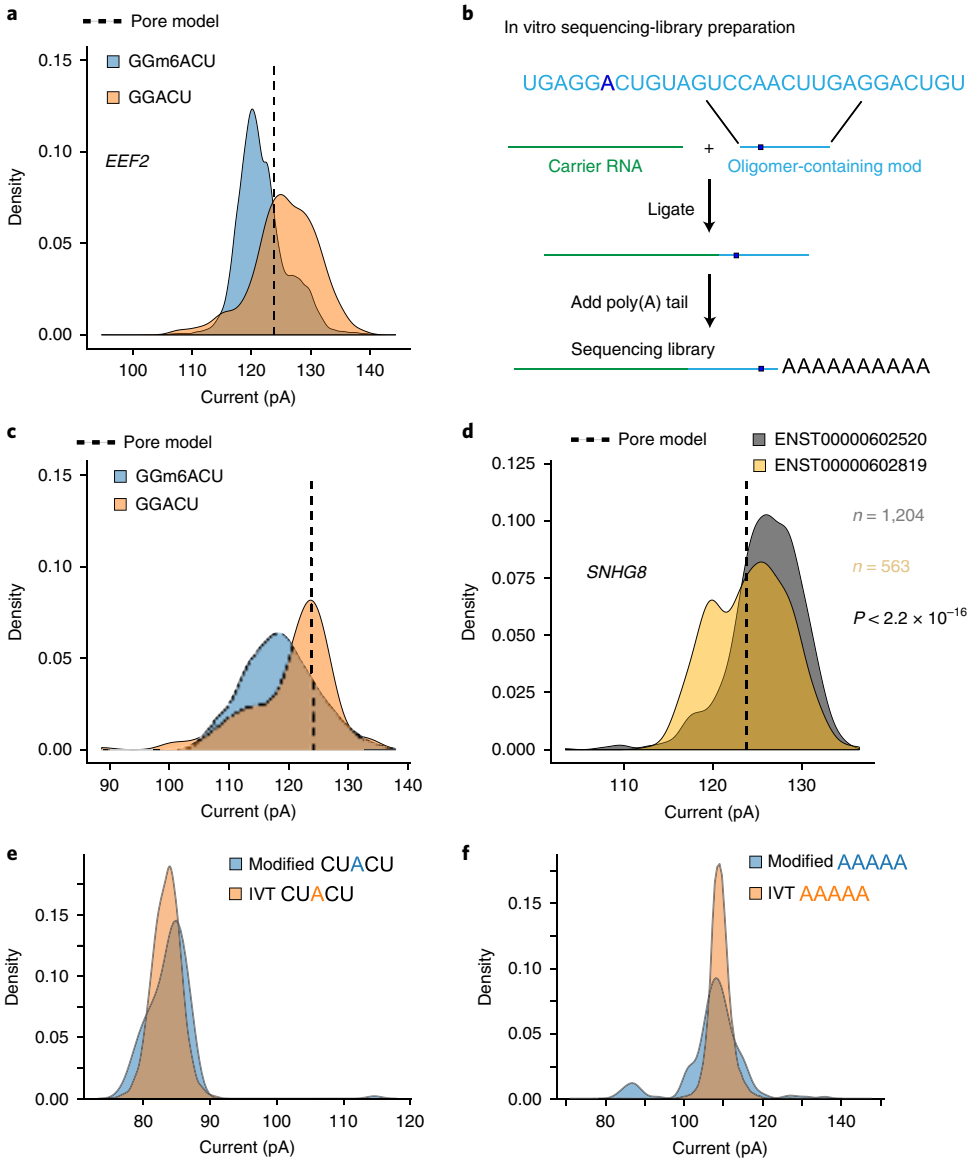


Fig. 6 | Nanopore detection of m6A and inosine base modifications. a, Comparing current signal from m6A-modified and unmodified GGACU motifs in the native RNA dataset for *EEF2* and in vitro transcribed dataset. Pore model (indicated by a dashed line) is defined as the mean current amplitude (pA) for the canonical GGACU 5-mer in the ONT model. **b**, Schematic for the oligomer-ligation. A synthetic RNA oligomer (Trilink Biotechnologies) containing canonical and modified m6A bearing GGACU 5-mer was ligated to a carrier RNA. This was followed by in vitro polyadenylation. **c**, Comparison of ionic current signals for m6A-modified and canonical GGACU motifs. The data were acquired using the assay described in **b**. **d**, Ionic current distributions for GGACU motifs within *SNHG8* gene isoforms (see gene models in Supplemental Fig. 7). **e**, Ionic current distributions for putative inosine-bearing CUACU 5-mer in the 3′-UTR region of the *AHR* gene. Blue is native RNA and orange is in vitro transcribed (IVT) RNA. **f**, Ionic current distributions for putative inosine-bearing AAAAA 5-mer in the 3′-UTR region of the *AHR* gene. Blue is native RNA and orange is IVT RNA.

that were identical except for the presence or absence of m6A within the GGACU motif (Fig. 6b). This revealed a clear current difference (Fig. 6c) consistent with the *EEF2* result.

To determine if m6A modifications differed between isoforms of the same gene, we screened GENCODE-sensitive isoforms for ionic current changes at the GGACU motif. We found 86 genes (198 isoforms) for which the median current levels at a single GGACU were significantly different between gene isoforms (Kruskal–Wallis, Student's *t*-test, and Kolmogorov–Smirnov statistical testing with Bonferroni multiple-testing correction). An example is illustrated for the *SNHG8* gene (Fig. 6d, isoform models in Supplementary Fig. 14).

Another post-transcriptional modification, A-to-I RNA editing[37], plays a role in splicing and regulating innate immunity[38,39]. NGS detects A-to-I editing as an A-to-G nucleotide variant in cDNA sequences.

Previous nanopore experiments documented the presence of systematic base miscalls in regions of *E. coli* 16S rRNA bearing modified RNA bases[7]. We found systematic base miscalls at putative inosine bearing positions in the GM12878 aryl hydrocarbon receptor (*AHR*) data (Supplementary Fig. 15). To cross-validate, we compared our cDNA sequence data relative to the GM12878 reference and found that putative inosines were detected as an A-to-G base change as expected (that is a single inosine for the CUACU 5-mer, and multiple inosines for the AAAAA 5-mer).

The ionic current distribution for the putative single inosine 5-mer (CUACU) was modestly different from the canonical 5-mer (Fig. 6e). The ionic current distribution for the inosine containing

AAAAA 5-mer was more complex, possibly reflecting the presence of multiple inosines (Fig. 6f).

Discussion

Nanopore RNA sequencing has two useful features. (1) The sequence composition of each strand is read as it existed in the cell. This permits direct detection of post-transcriptional modifications including nucleotide alterations and polyadenylation. (2) Reads can be continuous over many thousands of nucleotides providing splice-variant and haplotype phasing. Although each of these features is useful in itself, the combination is unique and likely to provide new insights into RNA biology. The two principal drawbacks of the present ONT nanopore RNA sequencing platform is the relatively high error rate (compared to Illumina cDNA sequencing), and uncertainty about the 5′ end of the transcript.

We were concerned about read fragmentation due to RNA degradation during sequencing. However, we found minimal (~5%) reduction in the full-length fraction of a 1.6 kb mRNA (*MT-CO1*) over 36 h. Preliminary analysis indicated that read truncations were more often caused by electronic signal noise due to current spikes of unknown origin. We showed that meaningful biological signals can be recovered from bulk Fast5 files around these truncations, suggesting that future improvements to the MinKNOW read segmentation pipeline are needed.

When combined with more accurate short Illumina reads, long nanopore reads allowed for end-to-end documentation of RNA transcripts bearing numerous splice junctions, which would not be possible using either platform alone. We documented a high proportion (52.6%) of unannotated isoforms, similar to other long-read transcriptome sequencing studies (for example, 35.6% and 49%)[40,41]. While many of these unannotated isoforms are low abundance and their protein coding potentials are unknown, it is important to catalog them because subtle splicing changes can impact function[42,43]. We also note that the number of detected isoforms did not saturate using the nanopore poly(A) RNA dataset, indicating that greater sequence depth will be necessary to give a comprehensive picture of the GM12878 poly(A) transcriptome.

A variety of techniques have been used to examine allele-specific expression (ASE)[15,24]. However, identification of ASE is limited using short read platforms because heterozygous variants are rare within any given window of a few hundred nucleotides. Nanopore sequencing has the advantage of long reads, albeit limited by errors. We attempted to mitigate the effects of these errors by requiring multiple heterozygous variants and a stringent false-discovery rate (FDR) during ASE analysis. Therefore, the number of genes that we report as demonstrating ASE (167) is likely an underestimation. We report nearly exclusive use of the maternal X-chromosome, with the only paternal transcripts originating from the *XIST* locus, consistent with previous findings[24]. We have shown that nanopore sequencing enables allele-specific isoform studies, especially in cases where the splicing variation does not have a heterozygous variant within range of conventional short read sequencing.

Polyadenylation of RNA 3′ ends regulates RNA stability and translation efficiency by modulating RNA-protein binding and RNA structure[26]. However, transcriptome-wide poly(A) analysis has been difficult because of basecalling and dephasing errors[28]. Recently implemented modifications to the Illumina strategy address these limitations[27,28]; but cannot resolve distal relationships, such as between splicing and poly(A) length. Nanopore poly(A) tail length estimation using nanopolish-polya offers the advantages of both direct length assessment and maintenance of information about isoform and modification status per transcript. Our preliminary studies revealed differences in poly(A) length distribution between mitochondrial and nuclear genes, between different nuclear genes, and between different isoforms of the same gene. We note in particular an increase in poly(A) tail length for some intron-retaining isoforms. This is consistent with previous work showing that hyper-adenylation targets intron-retaining nuclear transcripts for degradation through recognition by a poly(A)-binding protein (PABPN1)[44]. Additionally, deadenylation of cytoplasmic transcripts is a core part of the RNA-degradation pathway[45], suggesting that time-course experiments investigating RNA decay kinetics[46] could be possible with this technology.

We have demonstrated detection of N6-methyladenosine and inosine modifications in human poly(A) RNA. This validates prior work which showed modification-dependent ionic current shifts associated with m6A (*S. cerevisiae*)[5]. Differences in m6A modification level proved to be discernible at the isoform level for human *SNHG8* mRNA (Fig. 6d), documenting splicing variation and modification changes simultaneously.

Although other methods exist for high-throughput analysis of RNA modifications[47], they often require enrichment, which limits quantification, and they are usually short-read based. The latter precludes analysis of long-distance interactions between modifications, and between modifications and other RNA features such as splicing and poly(A) tail length. The capacity to detect these long-range interactions is likely to be important given recent work suggesting links between RNA modifications, splicing regulation and RNA transport and lifetime[48,49]. We argue that nanopore native RNA sequencing could deliver this long-range information for entire transcriptomes. However, this will require algorithms trained on large, cross-validated datasets as has been accomplished for cytosine and adenine methylation in genomic DNA[30,31].

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information, details of author contributions and competing interests, and statements of data and code availability are available at https://doi.org/10.1038/s41592-019-0617-2.

Received: 28 December 2018; Accepted: 19 September 2019;
Published online: 18 November 2019

References

1. Adams, M. D. Complementary DNA sequencing: expressed sequenced tags and human genome project. *Science* **252**, 1651–1656 (1991).
2. Temin, H. M. & Mizutani, S. RNA-dependent DNA polymerase in virions of Rous sarcoma virus. *Nature* **226**, 1211–1213 (1970).
3. Baltimore, D. Viral RNA-dependent DNA polymerase: RNA-dependent DNA polymerase in virions of RNA tumour viruses. *Nature* **226**, 1209 (1970).
4. Saiki, R. K. et al. Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science* **239**, 487–491 (1988).
5. Garalde, D. R. et al. Highly parallel direct RNA sequencing on an array of nanopores. *Nat. Methods* **15**, 201–206 (2018).
6. Jenjaroenpun, P. et al. Complete genomic and transcriptional landscape analysis using third-generation sequencing: a case study of Saccharomyces cerevisiae CEN.PK113-7D. *Nucleic Acids Res.* **46**, e38 (2018).
7. Smith, A. M., Jain, M., Mulroney, L., Garalde, D. R. & Akeson, M. Reading canonical and modified nucleobases in 16S ribosomal RNA using nanopore native RNA sequencing. *PLoS One* **14**, e0216709 (2019).
8. Steijger, T. et al. Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods* **10**, 1177–1184 (2013).
9. Venturini, L., Caim, S., Kaithakottil, G. G., Mapleson, D. L. & Swarbreck, D. Leveraging multiple transcriptome assembly methods for improved gene structure annotation. *Gigascience* **7**, giy093 (2018).
10. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
11. Jain, M. et al. Improved data analysis for the MinION nanopore sequencer. *Nat. Methods* **12**, 351–356 (2015).
12. Jain, M. et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338 (2018).
13. Szczesny, R. J. et al. RNA degradation in yeast and human mitochondria. *Biochim. Biophys. Acta* **1819**, 1027–1034 (2012).
14. Payne, A., Holmes, N., Rakyan, V. & Loose, M. BulkVis: a graphical viewer for Oxford nanopore bulk FAST5 files. *Bioinformatics* **35**, 2193–2198 (2018).

15. Tilgner, H., Grubert, F., Sharon, D. & Snyder, M. P. Defining a personal, allele-specific, and single-molecule long-read transcriptome. *Proc. Natl Acad. Sci. USA* **111**, 9869–9874 (2014).
16. Cho, H. et al. High-resolution transcriptome analysis with long-read RNA sequencing. *PLoS ONE* **9**, e108095 (2014).
17. Bernstein, B. E. et al. Genomic maps and comparative analysis of histone modifications in human and mouse. *Cell* **120**, 169–181 (2005).
18. Ernst, J. & Kellis, M. Discovery and characterization of chromatin states for systematic annotation of the human genome. *Nat. Biotechnol.* **28**, 817–825 (2010).
19. Ernst, J. et al. Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature* **473**, 43–49 (2011).
20. Deveson, I. W. et al. Universal alternative splicing of noncoding exons. *Cell Syst.* **6**, 245–255 (2018).
21. Gonzàlez-Porta, M., Frankish, A., Rung, J., Harrow, J. & Brazma, A. Transcriptome analysis of human tissues and cell lines reveals one dominant transcript per gene. *Genome Biol.* **14**, R70 (2013).
22. Baralle, F. E. & Giudice, J. Alternative splicing as a regulator of development and tissue identity. *Nat. Rev. Mol. Cell Biol.* **18**, 437–451 (2017).
23. Edge, P., Bafna, V. & Bansal, V. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. *Genome Res.* **27**, 801–812 (2017).
24. Rozowsky, J. et al. AlleleSeq: analysis of allele-specific expression and binding in a network framework. *Mol. Syst. Biol.* **7**, 522 (2011).
25. Brown, C. J. et al. A gene from the region of the human X inactivation centre is expressed exclusively from the inactive X chromosome. *Nature* **349**, 38 (1991).
26. Eckmann, C. R., Rammelt, C. & Wahle, E. Control of poly(A) tail length. *Wiley Interdiscip. Rev. RNA* **2**, 348–361 (2011).
27. Subtelny, A. O., Eichhorn, S. W., Chen, G. R., Sive, H. & Bartel, D. P. Poly(A)-tail profiling reveals an embryonic switch in translational control. *Nature* **508**, 66–71 (2014).
28. Chang, H., Lim, J., Ha, M. & Kim, V. N. TAIL-seq: genome-wide determination of poly(A) tail length and 3' end modifications. *Mol. Cell* **53**, 1044–1052 (2014).
29. Temperley, R. J., Wydro, M., Lightowlers, R. N. & Chrzanowska-Lightowlers, Z. M. Human mitochondrial mRNAs—like members of all families, similar but different. *Biochim. Biophys. Acta Bioenerg.* **1797**, 1081–1085 (2010).
30. Simpson, J. T. et al. Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods* **14**, 407–410 (2017).
31. Rand, A. C. et al. Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods* **14**, 411–413 (2017).
32. Liu, N. & Pan, T. N6-methyladenosine–encoded epitranscriptomics. *Nat. Struct. Mol. Biol.* **23**, 98–102 (2016).
33. Dai, D., Wang, H., Zhu, L., Jin, H. & Wang, X. N6-methyladenosine links RNA metabolism to cancer progression. *Cell Death Dis.* **9**, 124 (2018).
34. Sibbritt, T., Patel, H. R. & Preiss, T. Mapping and significance of the mRNA methylome. *Wiley Interdiscip. Rev. RNA* **4**, 397–422 (2013).
35. Meyer, K. D. et al. Comprehensive analysis of mRNA methylation reveals enrichment in 3' UTRs and near stop codons. *Cell* **149**, 1635–1646 (2012).
36. Roost, C. et al. Structure and thermodynamics of N6-methyladenosine in RNA: a spring-loaded base modification. *J. Am. Chem. Soc.* **137**, 2107–2115 (2015).
37. Licht, K., Kapoor, U., Mayrhofer, E. & Jantsch, M. F. Adenosine to Inosine editing frequency controlled by splicing efficiency. *Nucleic Acids Res.* **44**, 6398–6408 (2016).
38. Nishikura, K. Functions and regulation of RNA editing by ADAR deaminases. *Annu. Rev. Biochem.* **79**, 321–349 (2010).
39. Tajaddod, M., Jantsch, M. F. & Licht, K. The dynamic epitranscriptome: A to I editing modulates genetic information. *Chromosoma* **125**, 51–63 (2016).
40. Tardaguila, M. et al. SQANTI: extensive characterization of long-read transcript sequences for quality control in full-length transcriptome identification and quantification. *Genome Res.* **28**, 396–411 (2018).
41. Anvar, S. Y. et al. Full-length mRNA sequencing uncovers a widespread coupling between transcription initiation and mRNA processing. *Genome Biol.* **19**, 46 (2018).
42. Wang, L. et al. Transcriptomic characterization of SF3B1 mutation reveals its pleiotropic effects in chronic lymphocytic leukemia. *Cancer Cell* **30**, 750–763 (2016).
43. Bradley, R. K., Merkin, J., Lambert, N. J. & Burge, C. B. Alternative splicing of RNA triplets is often regulated and accelerates proteome evolution. *PLoS Biol.* **10**, e1001229 (2012).
44. Bresson, S. M., Hunter, O. V., Hunter, A. C. & Conrad, N. K. Canonical Poly(A) polymerase activity promotes the decay of a wide variety of mammalian nuclear RNAs. *PLoS Genet.* **11**, e1005610 (2015).
45. Yi, H. et al. PABP cooperates with the CCR4-NOT complex to promote mRNA deadenylation and block precocious decay. *Mol. Cell* **70**, 1081–1088 (2018).
46. Parker, R. & Song, H. The enzymes and control of eukaryotic mRNA turnover. *Nat. Struct. Mol. Biol.* **11**, 121–127 (2004).
47. Li, X., Xiong, X. & Yi, C. Epitranscriptome sequencing technologies: decoding RNA modifications. *Nat. Methods* **14**, 23–31 (2016).
48. Roundtree, I. A., Evans, M. E., Pan, T. & He, C. Dynamic RNA modifications in gene expression regulation. *Cell* **169**, 1187–1200 (2017).
49. Lee, M., Kim, B. & Kim, V. N. Emerging roles of RNA modification: m(6)A and U-tail. *Cell* **158**, 980–987 (2014).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2019

Methods

Unless otherwise noted, kit-based protocols described below followed the manufacturer's instructions.

GM12878 cell tissue culture. GM12878 cells (passage 4) were received from the Coriell Institute and cultured in RPMI medium (Invitrogen cat no. 21870076) supplemented with 15% non heat-inactivated FBS (Lifetech cat no. 12483020) and 2 mM L-Glutamax (Lifetech cat no. 35050061). Cells were grown to a density of 1×10^6 per ml before subsequent dilution of 1/3 every ~3 d and expanded to 9 × T75 flasks (45 ml of medium in each). Cells were centrifuged for 10 min at 100 × g (4 °C), washed in 1/10th volume of PBS (pH 7.4) and combined for homogeneity. The cells were then evenly split between 8 × 15 ml tubes and pelleted at 100*g* for 10 min at 4 °C. The cell pellets were then snap frozen in liquid nitrogen and immediately stored at −80 °C before shipping on dry ice. Two tubes of 5×10^7 frozen GM12878 cell pellets from passage 10 from a single passage, cultured at the University of British Columbia (UBC), were distributed and used at UBC, Ontario Institute of Cancer Research (OICR), Johns Hopkins University (JHU) and University of California Santa Cruz (UCSC). Two tubes of cells from passage 11 were distributed to University of Nottingham from UBC, and an independently cultured passage of GM12878 was used at University of Birmingham.

Total RNA isolation. The following protocol was used by each of the six institutions. Four ml of TRI-Reagent (Invitrogen AM9738) was added to a frozen pellet of 5×10^7 GM12878 cells and vortexed immediately. This sample was incubated at room temperature for 5 min. Four hundred µl BCP (1-Bromo-3-chloro-propane) or 200 µl $CHCl_3$ (Chloroform) was added per ml of sample, vortexed, incubated at room temperature for 5 min, vortexed again and centrifuged for 10 min at 12,000*g* (4 °C). The aqueous phase was pooled in a LoBind Eppendorf tube and combined with an equal volume of isopropanol. The tube was mixed, incubated at room temperature for 15 min, and centrifuged for 15 min at 12,000*g* (4 °C). The supernatant was removed, the RNA pellet was washed with 750 µl 80% ethanol and then centrifuged for 5 min at 12,000*g* (4 °C). The supernatant was removed. The pellet was air-dried for 10 min, resuspended in nuclease-free water (100 µl final volume), quantified and either stored at −80 °C or processed further by poly(A) purification.

Poly(A) RNA isolation. One hundred µg aliquots of total RNA were diluted in 100 µl of nuclease-free water and poly(A) selected using NEXTflex Poly(A) Beads (BIOO Scientific cat. no. NOVA-512980). Resulting poly(A) RNA was eluted in nuclease-free water and stored at −80 °C.

MinION native RNA sequencing of GM12878 poly(A) RNA. Biological poly(A) RNA (500–775 ng) and a synthetic control (Lexogen SIRV Set 3, 5 ng) were prepared for nanopore direct RNA sequencing generally following the ONT SQK-RNA001 kit protocol, including the optional reverse transcription step recommended by ONT. One difference from the standard ONT protocol was in the use of Superscript IV (Thermo Fisher) for reverse transcription. RNA sequencing on the MinION and GridION platforms was performed using ONT R9.4 flow cells and the standard MinKNOW (version 1.7.14) protocol script (NC_48 h_sequencing_FLO-MIN106_SQK-RNA001) recommended by ONT, with one exception — we restarted the sequencing runs at several time points to improve active pore counts and throughput during the first 24 h.

cDNA synthesis. First-strand cDNA synthesis was performed using Superscript IV (Thermo Fisher) and 100 ng of poly(A) purified RNA. Reverse transcription and strand-switching primers were provided by ONT in the SQK-PCS108 kit. After reverse transcription, PCR was performed using LongAmp Taq Master Mix (NEB) under the following conditions: 95 °C for 30 s, 11–15 cycles (95 °C for 15 s, 62 °C for 15 s, 65 °C for 15 min), 65 °C for 15 min, hold at 4 °C. The 15 cycle PCR was performed when using the SQK-PCS108 kit and 11 cycle PCR was performed when using the SQK-LSK308 kit. PCR products were purified using 0.8X AMPure XP beads.

MinION sequencing of GM12878 cDNA. cDNA sequencing libraries were prepared using 1 µg of cDNA following the standard ONT protocol for SQK-PCS108 (1D sequencing) or SQK-LSK308 (1D^2 sequencing) with one exception. That is, we used 0.8X aAMPure XP beads for cleanup. We used standard ONT MinKNOW scripts for MinION sequencing with one exception. That is, we restarted the sequencing runs at several time points to improve active pore counts and throughput during the first 24 h.

Acquiring continuous data for nanopore sequencing runs and resegmenting reads. For a subset of runs, 'bulk FAST5 files' containing continuous raw current traces and read decisions made by MinKNOW were recorded for more detailed analysis. This can be enabled in MinKNOW by looking at 'Additional options' under 'Output' when configuring a run to start in MinKNOW. Options were set to capture raw signal data and the read table. Events were not captured to reduce file size[14]. Bulk FAST5 files were investigated using BulkVis[14] and scripts available on GitHub (https://github.com/nanopore-wgs-consortium/NA12878/tree/master/nanopore-human-transcriptome/scripts/bulk_signal_read_correction). To identify reads with abnormal start or ends the read classifications made by MinKNOW in the 2 s before and after each read start or end respectively. Read starts should include 'pore', 'good_single', 'inrange' or 'unblocking' classifications[14]. Read ends should also end with these categories. Reads which did not start or end with these classifications were considered as potentially abnormal. Additional signal before and after the read was extracted from the bulk FAST5 file and a new synthetic read created for base calling (using Albacore version 2.1.3). For abnormal read starts, signal up to the start of the previous read was prepended. For abnormal read ends, signal up to the start of the following read was appended. Base calling is disrupted by signal incorrectly classified as open pore. Therefore these incorrect signal chunks were replaced with signal matching the mean for each read to generate a corrected read. These reads were recalled and mapped against the candidate targets using minimap2 with standard ONT parameters. This method can result in incorrectly concatenated reads, and so mapping to the target was used to filter out such sequences. The difference in target coverage for each read was used to indicate recovery of sequence data as summarized in Supplementary Fig. 7 and Supplementary Table 7. All corrected read files, basecalls, mapping files and scripts used to generate them are available on GitHub (link cited above).

Length analysis of mitochondrial protein-coding transcripts. In this analysis, we limited the test population for each gene to reads that aligned to a 50 nt sequence at the 3′ prime end of its ORF, except for *MT-ND5* where alignment was to a 50 nt sequence at the end of its 568 nt 3′ UTR. Full length was defined as extending to at least within 25 nt of the genes expected 5′ terminus. This limit was chosen because the processive enzyme that regulates RNA translocation is distal from the CsgG nanopore limiting aperture and necessarily falls off before the 5′ end is read. The sharpest coverage drop-off is typically at 10 nt from the 5′ transcript end; we chose the 25 nt limit to ensure that all likely full-length reads were captured in the count.

In vitro transcription. cDNA synthesis was performed according to ONT instructions (SQK-PCS108 kit) by combining Superscript IV (Thermo Fisher), RT and ONT strand switching primers, and 100 ng of poly(A) purified RNA. Next, an 11-cycle PCR reaction was performed using the ONT SQK-LSK308 kit but with a modified version of the primer that included a T7 promoter as recommended by NEB (catalog number E2040S). The PCR reaction was run under the following conditions: 95 °C for 30 s, 11 cycles (95 °C for 15 s, 62 °C for 15 s, 65 °C for 15 min), 65 °C for 15 min, hold at 4 °C.

PCR products were purified using 0.8X AMPure XP beads. Next, in vitro transcription was performed using the NEB HiScribe T7 High Yield RNA Synthesis Kit following NEB instructions. The IVT product was poly(A) tailed using the same kit. The resulting IVT RNA was purified using LiCl precipitation and then adapted for RNA sequencing on the MinION the using SQK-RNA001 kit.

Oligomer ligation. The oligomer containing the N6-methyladenosine modification was obtained as a lyophilized pellet from Trilink BioTechnologies and resuspended to 20 µM using TE buffer (Quality Biological catalog no. 351-011-721). The firefly luciferase (FLuc) transcript used as the carrier molecule was produced by in vitro transcription using the HiScribe ARCA mRNA Kit (with tailing) (NEB catalog no. E2060) and supplied protocol with the following exception: after DNase treatment, the reaction was terminated and the RNA purified using 1X Agencourt RNAClean XP beads (Beckman Coulter A63987). The oligomer was then treated with T4 polynucleotide kinase (PNK) (NEB catalog no. M0201) to phosphorylate the 5′ end for ligation. After phosphorylation, the oligomer was purified using the Oligo Clean & Concentrator kit (Zymo Research catalog no. D4060). The phosphorylated oligomer and FLuc transcript were quantified, combined in equimolar amounts, and ligated using T4 RNA Ligase 1 (NEB catalog no. M0204). The reaction mixture was incubated at 16 °C overnight. After incubation, the RNA was purified using RNAClean XP beads. The ligated product was poly(A) tailed using *E. coli* Poly(A) Polymerase (NEB HiScribe ARCA mRNA Kit) according to the supplier's instructions. After A-tailing, the RNA was purified using RNAClean XP beads. The isolated RNA was poly(A) selected using NEXTflex Poly(A) beads. The resulting poly(A) RNA was eluted in nuclease-free water and immediately prepared for sequencing using Oxford Nanopore's direct RNA sequencing kit (SQK-RNA001) and protocol.

Basecalling, alignments and percent identity calculations. We used the ONT Albacore workflow (version 2.1.0) for basecalling direct RNA and cDNA data. A strand read with an average sequence quality of 7 or higher (Q7) was classified as pass (default setting for Albacore (version 2.1.0)). We used minimap2 version 2.1 (ref. [10]) (recommended parameters that is -ax splice -uf -k14 for alignments to the human genome and -ax map-ont for alignments to the human transcriptome) to align the nanopore RNA and cDNA reads to the GRCh38 human genome reference (https://ftp-trace.ncbi.nih.gov/1000genomes/ftp/technical/reference/GRCh38_reference_genome/) and to the GENCODE v27 transcriptome reference (https://www.gencodegenes.org/releases/current.html). Minimap2 was chosen because it aligns nanopore reads to exons while spanning across introns. We used marginStats (version 0.1)[11] to calculate alignment identities and errors for pass RNA strand reads and pass 1D cDNA strand reads. Substitutions were calculated using custom scripts available within marginAlign (version 0.1)[11].

***k*-mer analysis.** We assessed nanopore RNA and cDNA 5-mer coverage using GENCODE isoforms. The read sequences were filtered by length and only reads covering 90% or more of the respective reference sequence were chosen. We calculated expected 5-mer counts from the set of reference sequences and observed 5-mer counts from the set of read sequences. For plotting purposes, we normalized the read and reference counts to coverage per megabase. The scripts are available within marginAlign[11].

Isoform detection and characterization. To define isoforms from the sets of native RNA and cDNA reads, we used FLAIR v1.4, a version of FLAIR[50] with additional considerations for native RNA nanopore data. For our analysis, we first removed reads generated by lab 6, because a disproportionate number of those molecules appeared to be truncated prior to addition to the nanopore flow cell. We also removed 71,276 aligned reads with deletions greater than 100 bases caused by minimap2 version 2.1. We then selected reads that had TSSs within promoter regions that were computationally derived from ENCODE ChIP-seq data[18,19]. Using FLAIR-correct, we corrected primary genomic alignments for pass reads based on splice junction evidence from GENCODE v27 annotations and Illumina short-read sequencing of GM12878. This step also removes reads containing non-canonical splice junctions not present in the annotation or short-read data. The filtered and corrected reads were then processed by FLAIR-collapse which generates a first-pass isoform set by grouping reads on their splice junctions chains. Next, pass reads were realigned to the first-pass isoform set, retaining alignments with MAPQ > 0. Isoforms with fewer than three supporting reads or those which were subsets of a longer isoform were filtered out to compile the FLAIR-sensitive isoform set. A FLAIR-stringent isoform set was also compiled by filtering the FLAIR-sensitive set for isoforms which had 3 supporting reads that spanned ≥80% of the isoform and a minimum of 25 nt into the first and last exons. Unannotated isoforms were defined as those with a unique splice junction chain not found in GENCODE v27. Isoforms were considered intron-retaining if they contained an exon which completely spanned another isoform's splice junction. Isoforms with unannotated exons were defined as those with at least one exon that did not overlap any existing annotated exons in GENCODE v27. Genes that did not contain an annotated start codon were considered non-coding genes.

Defining promoter regions in GM12878 for isoform filtering. Promoter chromatin states for GM12878 were downloaded from the UCSC Genome Browser in BED format from the hg18 genome reference. Chromatin states were derived from an HMM based on ENCODE ChIP-seq data of nine factors[18,19]. The liftover tool[51] was used to convert hg18 coordinates to hg38. The active, weak and poised promoter states were used.

Haplotype assignment and allele-specific analysis. We obtained genotype information for GM12878 from existing phased Illumina platinum genome data generated by deep sequencing of the cell donors' familial trio[52]. The bcftools package was used to filter for only variants that are heterozygous in GM12878. Starting with aligned reads, we used the extractHAIRS utility of the haplotype-sensitive assembler HapCUT2 (ref. [23]) to identify reads with allele-informative variants. For allelic assignment, we required a read to contain at least 2 variants, and required that greater than 75% of identified variants agreed on the parental allele of origin—this stringent threshold was selected to reduce the chances of incorrect assignment from nanopore sequencing errors. Through this approach, each read was annotated as maternal, paternal or unassigned. To identify genes that demonstrated a very strong bias for a single allele, we performed a binomial test of all reads assigned to a parental allele, with an FDR of 0.001. We also visually inspected numerous genes displaying genes demonstrating allele-specificity using IGV, to increase our confidence in proper mapping of the reads and evaluate the presence of variants.

We further integrated this haplotype-specific analysis with our isoform pipeline to explore for the presence of allele-specific isoforms. If reads for a specific isoform originated from a single parental allele (binomial test; false discovery rate, 0.001), the isoform was assigned as allele specific. We then filtered for any genes which contained both maternal and paternal allele-specific isoforms, and visually inspected these isoforms using IGV to compare location of variants and splicing events.

Poly(A) tail length analysis. Supplementary Note 1 describes use of nanopolish-polya version 0.10.2 (https://github.com/jts/nanopolish) to estimate polyadenylated tail lengths of nanopore native RNA sequence reads. We used the Kruskal–Wallis test as implemented in Python to determine statistically significant changes between isoforms; code is available at https://github.com/nanopore-wgs-consortium/NA12878/tree/master/nanopore-human-transcriptome/scripts.

Modification detection and analysis. We focused our initial efforts on m6A modification in genes previously identified as enriched in modifications from m6A immunoprecipitation sequencing data on human cell lines[36,53]. We aligned native RNA reads and IVT RNA reads to candidate genes and then extracted ionic current information (mean current and standard deviation in pA) for specific 5-mers using nanopolish eventalign (version 0.10.2). We compared ionic current kernel density estimates (KDE) for GGACU within the 3′ UTR of the *EEF2* gene in native RNA with the KDE for its canonical IVT RNA counterpart. The extent and directionality of current shifts observed by m6A modification within the GGACU motif were orthogonally investigated using an in-vitro oligomer ligation assay, as described above. We compared KDEs for the modified and unmodified GGACU motifs within the synthetic oligomer. Statistical testing (Kruskal–Wallis, Student's *t*-test, Kolmogorov–Smirnov and Bonferroni correction) was implemented in Python with code available at https://github.com/nanopore-wgs-consortium/NA12878/tree/master/nanopore-human-transcriptome/scripts.

For detecting A-to-I editing, we focused on the 3′-UTR region of the human aryl hydrocarbon receptor (*AHR*) gene. Using the UCSC Genome Browser, we identified systematic G base variant calls in *AHR* cDNA data (probable inosine substitutions in RNA). We then tested for systematic base miscalls at the corresponding positions in native RNA data. Next, we used nanopolish eventalign (version 0.10.2) to extract ionic current information for two putative inosine-containing 5-mers (CUACU and AAAAA), and for their respective IVT-derived canonical 5-mers from chromosome 7. Ionic current distributions for CUACU and AAAAA 5-mers between the biological and IVT data were compared using kernel density estimates.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Sequence data including raw signal files (FAST5), event-level data (FAST5), base-calls (FASTQ) and alignments (BAM) are available as an Amazon Web Services Open Data set, for download from https://github.com/nanopore-wgs-consortium/NA12878. The scripts used for various analyses are also available from the same GitHub under nanopore-human-transcriptome/scripts.

Code availability

General scripts available at: https://github.com/nanopore-wgs-consortium/NA12878/tree/master/nanopore-human-transcriptome/scripts. Poly(A) caller ('nanopolish-polya', https://github.com/jts/nanopolish) and isoform analysis code for FLAIR (https://github.com/BrooksLabUCSC/flair).

References

50. Tang, A. D. et al. Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns. Preprint at *bioRxiv* https://doi.org/10.1101/410183 (2018).
51. Hinrichs, A. S. et al. The UCSC genome browser database: update 2006. *Nucleic Acids Res.* **34**, D590–D598 (2006).
52. Eberle, M. A. et al. A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27**, 157–164 (2016).
53. Molinie, B. et al. m6A-LAIC-seq reveals the census and complexity of the m6A epitranscriptome. *Nat. Methods* **13**, 692 (2016).

Acknowledgements

The authors are grateful for support from the following individuals. L. Snell, B. Sipos and D. Turner (ONT) provided materials and advice relevant to the 3′ poly(A) standards used to test nanopolish-polya. D. Garalde (ONT) provided early advice on use of the MinION for RNA sequencing. N. Conrad gave insight into the correlation of intron retention and poly(A) tail length. M. Diekhans reviewed the isoform analysis. Z. M. Chrzanowska-Lightowlers, T. Suzuki and S. Okada commented on early drafts of the manuscript. A. Beggs, L. Tee and T. Nieto (University of Birmingham, UK) provided cell cultures used in the Birmingham sequencing runs. The project was supported by the following grants: NIH HG010053 (A.N.B., B.P. and M.A.), NIH 5T32HG008345 (A.D.T.), NIH HG010538 (W.T.), NIH U54HG007990 (B.P.), U01 HL137183-02 (B.P.), Oxford Nanopore Research Grant SC20130149 (M.A.), National Institutes of Health Research Surgical Reconstruction and Microbiology Research Centre (J.Q.), Medical Research Council CLIMB Fellowship (N.L.), Wellcome Trust 204843/Z/16/Z (M.L.), BBSRC BB/N017099/1 and BB/M020061/1 (M.L.), the Canada Research Chair in Biotechnology and Genomics-Neurobiology (T.P.S.), the Canadian Institutes of Health Research (no. 10677; T.P.S.), the Canadian Epigenetics, Environment and Health Research Consortium (T.P.S.), the Koerner Foundation (T.P.S.), Genome Canada (OGI-136, J.T.S.), and the Ontario Institute for Cancer Research through funds provided by the Government of Ontario (J.T.S.), Pew Charitable Trust (A.N.B.).

Author contributions

M.A., W.T., H.E.O., M.J. and J.R.T. conceived the study. M.A., A.N.B. and W.T. coordinated the collaboration. R.E.W., N.S., N.H., J.Q., P.C.Z., H.E.O., M.J., J.R.T. and T.G. acquired data. R.E.W., A.D.T., N.S., T.G., M.L., A.P., N.L., R.R., A.N.B., P.S.T., J.T.S., B.P., H.E.O., J.R.T., W.T., M.A. and M.J. analyzed and interpreted data. Specifically, R.E.W. performed a first pass analysis and data indexing; T.G. and R.R. performed the allele-specific analysis; R.E.W. and R.R. performed the m6A modification analysis; P.S.T. and J.T.S. designed and implemented the poly(A) tail length estimation software; A.D.T. and A.N.B. performed transcript isoform analysis; P.S.T., W.T., R.R. and N.S. performed the polyA tail analysis; M.J. and H.E.O. performed the A-to-I base modification analysis;

J.T., M.J., N.L. and H.E.O. performed sequencer performance analysis; and M.A., M.J., H.E.O., M.L. and A.P. performed mitochondrial gene expression analysis. The following were principally responsible for text and figures by topic: RNA preparation, nanopore sequencing, and computational pipeline (M.J., H.E.O., J.R.T., M.A.); native poly(A) RNA sequencing statistics (M.J., H.E.O., J.R.T., M.A.); FLAIR-based isoform detection and analysis (A.D.T., C.M.S., A.N.B.); assignment of transcripts to parental alleles using nanopore reads (T.G., R.R., W.T.); mitochondrially-encoded transcripts (M.A., H.E.O., M.J., M.L., A.P.); *k*-mer coverage (H.E.O., M.J.); 3′ poly(A) analysis (P.S.T., J.T.S., W.T., R.R., T.G.); m6A analysis (R.E.W., W.T., R.R., N.S.); A-to-I conversion (M.J., H.E.O.). Manuscript revisions and edits (R.E.W., A.D.T., P.S.T., M.J., J.R.T., P.C.Z., T.G., R.R., N.S., T.P.S., N.L., B.P., M.L., J.T.P., H.E.O., A.N.B., M.A., W.T.). K.L.J. and J.G.d.J. replicated and distributed GM12878 cells.

Competing interests

M.A. holds options in Oxford Nanopore Technologies (ONT). M.A. is a paid consultant to ONT. R.E.W., W.T., T.G., J.R.T., J.Q., N.J.L., J.T.S., N.S., A.N.B., M.A., H.E.O., M.J. and M.L. received reimbursement for travel, accommodation and conference fees to speak at events organised by ONT. N.L. has received an honorarium to speak at an ONT company meeting. W.T. has two patents (8,748,091 and 8,394,584) licensed to ONT. M.A. is an inventor on 11 UC patents licensed to ONT (6,267,872, 6,465,193, 6,746,594, 6,936,433, 7,060,50, 8,500,982, 8,679,747, 9,481,908, 9,797,013, 10,059,988, and 10,081,835). J.T.S., M.L. and M.A. received research funding from ONT.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0617-2.

Correspondence and requests for materials should be addressed to M.A. or W.T.

Peer review information Nicole Rusk was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

ARTICLES
https://doi.org/10.1038/s41587-020-00746-x
nature biotechnology

Check for updates

Readfish enables targeted nanopore sequencing of gigabase-sized genomes

Alexander Payne, Nadine Holmes, Thomas Clarke, Rory Munro, Bisrat J. Debebe and Matthew Loose ✉

Nanopore sequencers can be used to selectively sequence certain DNA molecules in a pool by reversing the voltage across individual nanopores to reject specific sequences, enabling enrichment and depletion to address biological questions. Previously, we achieved this using dynamic time warping to map the signal to a reference genome, but the method required substantial computational resources and did not scale to gigabase-sized references. Here we overcome this limitation by using graphical processing unit (GPU) base-calling. We show enrichment of specific chromosomes from the human genome and of low-abundance organisms in mixed populations without a priori knowledge of sample composition. Finally, we enrich targeted panels comprising 25,600 exons from 10,000 human genes and 717 genes implicated in cancer, identifying *PML–RARA* fusions in the NB4 cell line in <15 h sequencing. These methods can be used to efficiently screen any target panel of genes without specialized sample preparation using any computer and a suitable GPU. Our toolkit, readfish, is available at https://www.github.com/looselab/readfish.

Selective sequencing, or 'Read Until', refers to the ability of a nanopore sequencer to reject individual molecules while they are being sequenced. This requires the rapid classification of current signal from the first part of the read to determine whether the molecule should be sequenced or removed and replaced with a new molecule. We first demonstrated this using dynamic time warping (DTW) to compare the signal with a simulated current trace derived from a reference sequence[1]. Although DTW enabled a small set of use cases, it required substantial computational resources, preventing its generalized use[2]. Another recent method using raw signal, UNCALLED[3], has a lighter computational footprint than previous signal-based methods, but is limited in search space and still requires considerable computational resources. An alternative approach, which uses direct base-calling of signal chunks[4], demonstrated benefit compared with sequencing without Read Until as it filtered out unwanted reads, but did not provide any enrichment and again required considerable computational resources.

Our goal was to work with nucleotide sequences rather than raw signals to exploit existing tools, utilize reasonable computational resources and show enrichment of targets. To do this, we used Oxford Nanopore Technologies (ONT) base-calling software. ONT have developed a number of base-callers for nanopore sequence data, initially utilizing hidden Markov models and available through the metrichor cloud service[5]. They replaced these with neural network models running on central processing units and then GPUs. For real-time base-calling, ONT provide a range of computational platforms with integrated GPUs (minIT, Mk1C, GridION and PromethION). These devices enable real-time base-calling sufficient to keep pace with flow cells generating data. Most recently, these base-callers acquired a server–client configuration, such that raw signal can be passed to the server and a nucleotide sequence returned. Using this, we show that GPU base-calling can be used to deliver a real-time stream of nucleotide data from flow cells sequencing with up to 512 channels simultaneously. At the same time, the GPU can base-call completed reads, and optimized tools such as minimap2 (ref. [6]) can therefore be used to map reads as they are generated, enabling dynamic updating of both the targets and the reference genome as results change.

As our method does not use raw signal comparison, we do not have to convert reference genomes into signal space as in DTW or other signal methods[1,3]. We are constrained by access to a sufficiently powerful GPU. The results presented here mainly utilize the ONT GridION MK1, which includes an NVIDIA GV100 GPU, but we also use an NVIDIA 1080, showing that this approach works on any device capable of real-time base-calling. We apply this approach to a range of model problems. First, we select specific human chromosomes, illustrating that gigabase references are not a constraint. Second, we enrich low-abundance genomes from a mixed population and find that we reduce the time required to answer a biological question (time-to-answer) and improve the ability to assemble low-copy genomes. Adaptive sampling is the process by which the software changes what is being sequenced in response to what has been seen during an experiment. To illustrate this, we use centrifuge to identify the most abundant species present in a metagenomic sample, monitor depth of coverage for each in real time and enrich for the least abundant genomes without a priori knowledge of content[7]. This method is necessarily limited by the composition of the reference database and also requires network access to retrieve references once identified. Finally, we enrich panels of human genes, including 25,600 target regions corresponding to ~10,000 genes and 717 genes from the Catalogue of Somatic Mutations in Cancer (COSMIC) panel[8]. We demonstrate how Read Until can be used to capture information on key targets without the need for custom library preparation and show that we can identify a known translocation in the NB4 cell line in <15 h (ref. [9]).

We provide a configurable toolkit, readfish, enabling targeted sequencing of gigabase-sized genomes. This includes depletion of host sequences as well as example methods giving the minimum coverage depth for specific sequences in a population. Configuration of these tools is relatively straightforward and requires no additional computing resources as long as a sufficiently powerful GPU capable of base-calling multiple flow cells in real time is available.

DeepSeq, School of Life Sciences, Queens Medical Centre, University of Nottingham, Nottingham, UK. ✉e-mail: matt.loose@nottingham.ac.uk

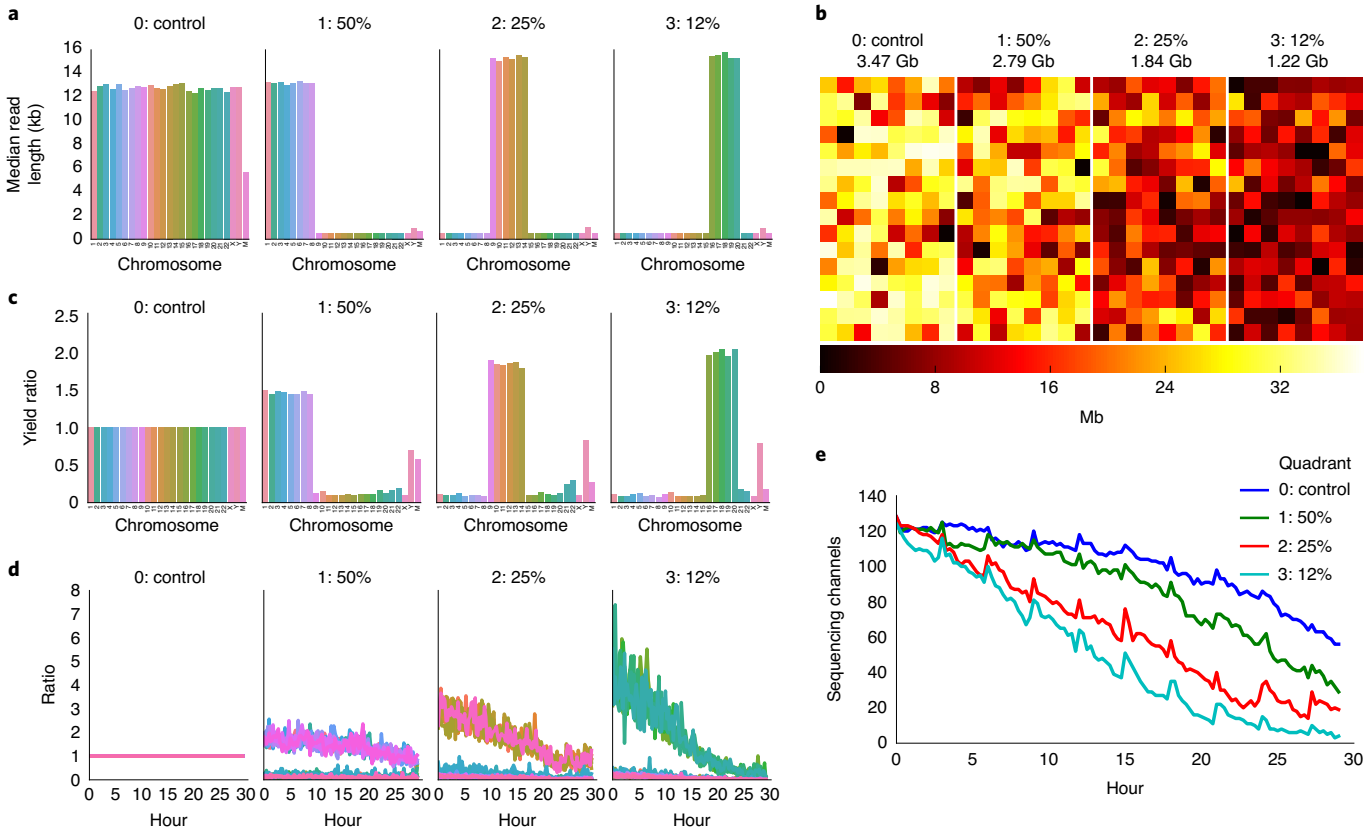


Fig. 1 | Human-genome-scale selective sequencing. a, Median read lengths for reads sequenced from GM12878 and mapped against hg38 excluding alternate chromosome representations. The four panels each represent a quadrant of the flow cell. In the control, all reads are sequenced; in the second, third and fourth quadrants, reads mapping to chromosomes 1–8, 9–14 and 16–20, respectively, are sequenced. The combined length of each of these target sets equates to approximately 1/2, 1/4 and 1/8 of the human genome, respectively. **b**, A heatmap of throughput per channel in each quadrant from the flow cell illustrating reduced yield as the proportion of reads rejected is increased. **c**, The yield ratio for each chromosome in each condition normalized against the yield observed for each chromosome in the control quadrant. **d**, The yield of on-target reads calculated in a rolling window over the course of the sequencing run showing the loss of enrichment potential. **e**, A plot of the number of channels contributing sequence data over the course of the sequencing run. Channels are lost at a greater rate when more reads are rejected.

Results

Methods overview. Selective sequencing requires bidirectional communication with a nanopore sequencer through the Read Until application programming interface (API; https://github.com/nanoporetech/read_until_api). The API provides a stream of raw current samples from every sequencing pore on the flow cell and allows the user to respond in real time, either rejecting a read from a specific pore or allowing a read to finish naturally. Previous API implementations served any signal seen as a potential read and so required the processing of many signals that were not genuine reads, causing analysis challenges[4]. The current API discriminates true DNA signal from background more efficiently and is configured to provide only signals identified as DNA, reducing the analysis burden. We reasoned that the signal served by the API should be compatible with the Guppy base-caller and so capture short signal sequences and process them in base space.

Supplementary Fig. 1a illustrates the workflow for base-calling reads as they are being sequenced. Briefly, data chunks of signal are served from the Read Until API. Chunks default to 1-s duration but can be configured by the user. We found that 0.4-s chunk durations (~180 bases; see Methods) balanced the need for small chunks with API performance (Supplementary Table 1 and Supplementary Fig. 2). The data chunk (up to 512 reads from a MinION flow cell) is converted to a Guppy-compatible format and base-called using pyguppyclient (https://github.com/nanoporetech/pyguppyclient). Base-called data are then mapped to a reference with minimap2 (ref. [6]). Reads may uniquely map, map to multiple locations or may not map at all. In response, the user can choose to reject a read (unblock), acquire more data for that read (proceed) or stop receiving data for the remainder of that read (stop receiving).

Read Until performance. To test the performance of our real-time base-calling approach on enrichment and depletion, we sequenced the well-studied NA12878 reference cell line[10]. The flow cell was configured to operate in quadrants each sequencing: a control (all reads accepted), chromosomes 1–8 (50% of reads accepted), chromosomes 9–14 (25% of reads accepted) and finally chromosomes 16–20 (12.5% of reads accepted). Reads are base-called and mapped to the reference regardless of quadrant. Median read lengths per chromosome in each quadrant indicate those sequenced or rejected (Fig. 1a). Selectively sequenced reads have a median read length of ~15 kilobases (kb). Rejected reads have a median length of ~500 bases, equating to ~1.1 s of sequencing time at 450 bases per second, although median data collected were closer to 1.5 s. Reads are base-called, mapped and the unblock action sent and actioned within ~1 s of the read starting. This run generated 9.32 Gb of aligned sequence data, unevenly distributed across the quadrants: 3.47 Gb in the control, 2.79 Gb at 50% acceptance, 1.84 at 25% acceptance and only 1.22 Gb at 12% (Fig. 1b and Supplementary Table 2). For each quadrant, the optimal enrichment is twofold, fourfold and eightfold,

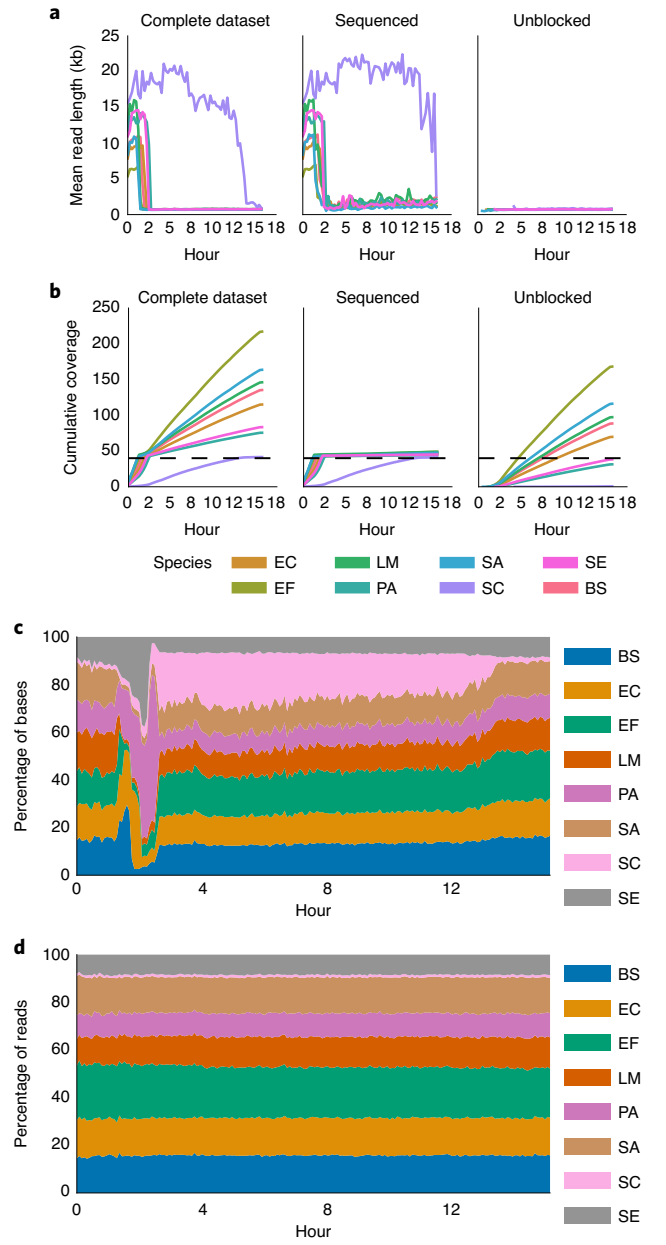


Fig. 2 | Adaptive sequencing enriching for the least abundant genome and ensuring uniform 40× coverage. a, Mean read lengths for reads sequenced from the ZymoBIOMICS mock metagenomic community mapped against the provided references (ZymoBIOMICS). Read lengths are reported for the whole run, the deliberately sequenced reads and those that were actively unblocked. **b**, Cumulative coverage of each ZymoBIOMICS genome during the sequencing run. The total coverage still accumulated as unblocked reads, though short, still map. Sequencing was automatically terminated once each sample reached 40×. **c**, A stacked area graph illustrating how the proportion of bases mapping to each species changes over time. **d**, By contrast, the proportion of reads mapping to each species over time does not change significantly. Species and composition: BS, *Bacillus subtilis* (14%); EF, *Enterococcus faecalis* (14%); EC, *Escherichia coli* (14%); LM, *Listeria monocytogenes* (14%); PA, *Pseudomonas aeruginosa* (14%); SC, *Saccharomyces cerevisiae* (2%); SE, *Salmonella enterica* (14%); SA, *Staphylococcus aureus* (14%).

but we see lower enrichments by the end of the experiment, presumably due to reduced yield (Fig. 1c). We observe enrichment of target sequences in all cases compared with control. Relative enrichment is closer to the theoretical maximum at the beginning of the sequencing run (Fig. 1d). Analysis of available channels contributing to data generation shows that sequencing capacity is lost faster as more reads are rejected (Fig. 1e). For this experiment, we did not nuclease flush the flow cell, but anticipate improvements in both the yield and enrichment if we did. We were able to call all batches within our 0.4-s window (Supplementary Fig. 3e).

A common goal in sequencing library preparation is to remove host DNA to enrich for a metagenomic subpopulation[11,12]. Selective sequencing may be beneficial in conjunction with library preparation methods. We considered metagenomics applications as a similar class of problem. Nicholls et al. generated a reference dataset using the ZymoBIOMICS Microbial Community Standards[13]. They were able to generate sufficient data to assemble several of the bacteria into single contigs (without binning). Notably, eukaryotic genomes that were present at lower abundance (2%) did not generate high-contiguity assemblies. This is not surprising as the coverage depth for *Saccharomyces cerevisiae* was 17× and that for *Cryptococcus neoformans* was 10× when sequencing on a single GridION flow cell[13]. Enriching for these low-abundance components is conceptually similar to depleting host material from a sample. In our experiments, we utilize the ZymoBIOMICS high-molecular-weight DNA standard (D6322). This sample will a priori improve assemblies owing to the longer read lengths and further differs from Nicholls et al. as it excludes *C. neoformans*.

To determine whether selective sequencing could improve the relative coverage of low-abundance material, we developed a simple pipeline (readfish align) to drive our selective sequencing decisions (Supplementary Fig. 1b). This pipeline aligns completed reads against a reference as they are written to disk, and then calculates the coverage depth. Once an individual species reaches the desired coverage depth, new reads mapping to that species are rejected. We simultaneously base-call both the real-time stream from Read Until and completed reads. Finally, we implemented Run Until to stop the run once all targets had reached sufficient coverage. These experiments used a community-specific reference file. Mean read lengths for target genomes reduce as they are added to the rejection list and the mean read length becomes dominated by short, rejected reads (Fig. 2a). Plotting coverage over time for reads not rejected by Read Until shows a decrease in coverage accumulation for completed genomes (that is, those at the desired coverage level) with an increase in sequencing potential for the least abundant sample, *S. cerevisiae* (Fig. 2b). The proportion of bases mapping to each genome reveals the shift in sequencing capacity to *S. cerevisiae* (Fig. 2c). Relative abundance can still be determined when running Read Until as the proportion of reads mapping to each genome does not change (Fig. 2d). The run automatically stops once each genome reaches 40×, taking ~16 h and 4.4 Gb of sequence data (Supplementary Fig. 4).

This sample should be 2% *S. cerevisiae* by bases, typically yielding ~88 Mb or 7× coverage of sequence data. Using selective sequencing, we see 40× coverage, naively a 5.7-fold increase in on-target data. However, a flow cell not implementing selective sequencing would have a higher yield, so real-world enrichment is lower. Nicholls et al. report 16 Gb on a similar sample generated in 48 h, which would result in ~25× coverage of *S. cerevisiae*, bringing enrichment closer to 1.6× (ref. [13]). Theoretically, enrichment of a 2% subset should be greater, but there is a cost to rejecting an individual read. Even so, we could enrich the least abundant element compared with that expected from the sample composition in multiple experiments ($n = 3$). Thus, we accelerate time-to-answer for a particular coverage depth (16 h versus 48 h). This approach assumes knowledge of the sample a priori and so is of limited practical relevance. By integrating a metagenomics classifier into our pipeline (readfish centrifuge), we avoid this requirement[7]. As strains are identified within the sample, they can be dynamically tracked and added to a rejection list, illustrating the principle of adaptive sequencing.

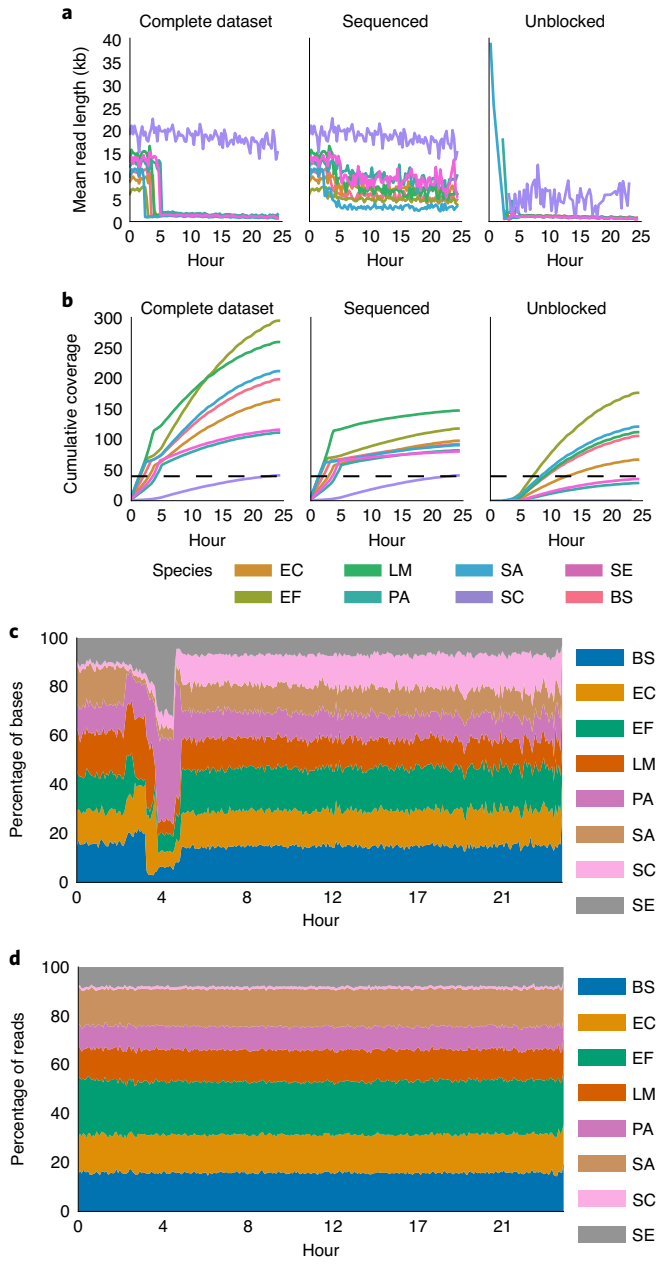


Fig. 3 | Adaptive sequencing enriching for the least abundant genome with centrifuge read classification and ensuring uniform 50× coverage. **a**, Mean read lengths for reads sequenced from the ZymoBIOMICS mock metagenomic community mapped against the provided references. Read lengths are reported for the whole run, the deliberately sequenced reads and those that were actively unblocked. **b**, Cumulative coverage of each ZymoBIOMICS genome during the sequencing run. The total coverage still accumulated as unblocked reads, though short, still map. Sequencing was automatically terminated once each sample reached 50×. The small overshoot in sequenced read coverage is likely caused by the centrifuge step lagging as reads are not instantly written to disk. **c**, A stacked area graph illustrating how the proportion of bases mapping to each species changes over time. **d**, By contrast, the proportion of reads mapping to each species over time does not change significantly. The species and composition are as in Fig. 2.

Using this approach, we generated 5.995 Gb of sequence data and identified all bacterial genomes in the sample; although we observed enrichment, the flow cell became completely blocked before reaching the target coverage (Fig. 3, Supplementary Table 2 and Supplementary Figs. 5 and 6). Six gigabases of sequence should result in ~10× coverage; here we obtained 41× coverage (Fig. 3b). In this case, we considered the entire duration of a read as a candidate for Read Until; consequently, some reads are rejected later into the read. This results in a wider range of mean rejected read lengths, particularly for *S. cerevisiae* (Fig. 3a). This experiment was completed within 24 h, illustrating the benefits in terms of time-to-answer. As expected, improved coverage depth results in almost complete assemblies using MetaFlye compared to that achieved by Nicholls et al. (Supplementary Fig. 7), in part a consequence of improved read lengths here[13,14]. Subsequent nuclease flushing of the flow cell would increase effective throughput, but this was not our goal.

Methods for target panel enrichment include PCR amplification, bait capture methods and CRISPR–Cas9 approaches[15–18]. These methods are reliable and cost effective at scale, but have development, instrument and consumable costs. Unlike methods that capture native DNA[17], PCR-based methods cannot capture methylation information without additional processing. Such panels cannot be altered easily.

Selective sequencing provides an alternative, and so we identified 19,296 target genes annotated as protein-coding with transcript name IDs (see Methods) from the human genome (GRCh38), excluding those on X and Y and ignoring alternate chromosome representations[19]. We extracted exon coordinates, extended 3 kb either side and collapsed overlapping targets. We enriched for targets found on odd-numbered chromosomes, rejecting all reads from outside these targets. This results in a total search space of 176 Mb (~5%) containing 25,600 targets covering ~10,000 genes (Fig. 4a). A single GridION flow cell with 1,660 pores gave 6.1 Gb of sequence data in 24 h. After nuclease flushing, loading additional library and 24 h more sequencing gave 5.573 Gb (total yield: 11.675 Gb, N50 (the read length such that reads of this length or greater sum to at least half the total bases): 9 kb; Supplementary Table 2). Exon targets had a median coverage of 17.23× (mean 17.39×) with 75% >14.15× and 25% >20.42×. On 'control' even-numbered chromosomes, the median coverage was 0.98× (mean 1.2×). Detailed coverage plots of targets on odd-numbered (Fig. 4c,d) and even-numbered (Fig. 4e,f) chromosomes correlate with the target regions. Controlling for these experiments is complicated by flow cell variability. We make comparisons with theoretical yields of 10, 20 and 30 Gb, resulting in approximately 3–10× coverage. Our effective enrichment is from 2.7× to 5.4×, consistent with our earlier observations. Nuclease flushing assists enrichment and flow cell efficiency (Supplementary Fig. 8).

Our exon panel contains 371 genes from COSMIC with a median coverage of 13.7× (Fig. 4b)[8]. Figure 4c,d shows the coverage for *BRCA1*, *PML* and surrounding targets. Although it is preferable to include introns, here we excluded intronic sequences to reduce the total search space (although not required). To further explore this and illustrate the flexibility of our approach, we targeted the entire COSMIC panel (717 genes) excluding genes with no given genomic coordinates (Supplementary File 1). Including flanking 5-kb sequences, our search space was 89.9 Mb (~2.7% of the genome). Using a flow cell with 1,724 pores, we generated 3.7 Gb within 24 h. Nuclease flush and reload generated a further 6.03 Gb, giving a total of 9.73 Gb, with a read N50 of 940 bases (Fig. 5, Supplementary Fig. 9 and Supplementary Table 2). Deliberately rejected reads had an N50 of 515 bases; sequenced reads had an N50 of 11,564 bases. Gene targets had a median coverage of 32.2× (mean 30.7×; Fig. 5a and Supplementary File 1), with 75% of genes >28× and 25% of genes >35×. Figure 5c–f shows the coverage for *BRCA1*, *PML*, *WIF1* and *HOXC11*/*C13*. The specificity of selective sequencing is clear, particularly where neighboring genes in the *HOXC* cluster are not sequenced. A second run, utilizing three flushes, one every 24 h, generated a total of 17.87 Gb with a read N50 of 793 bases

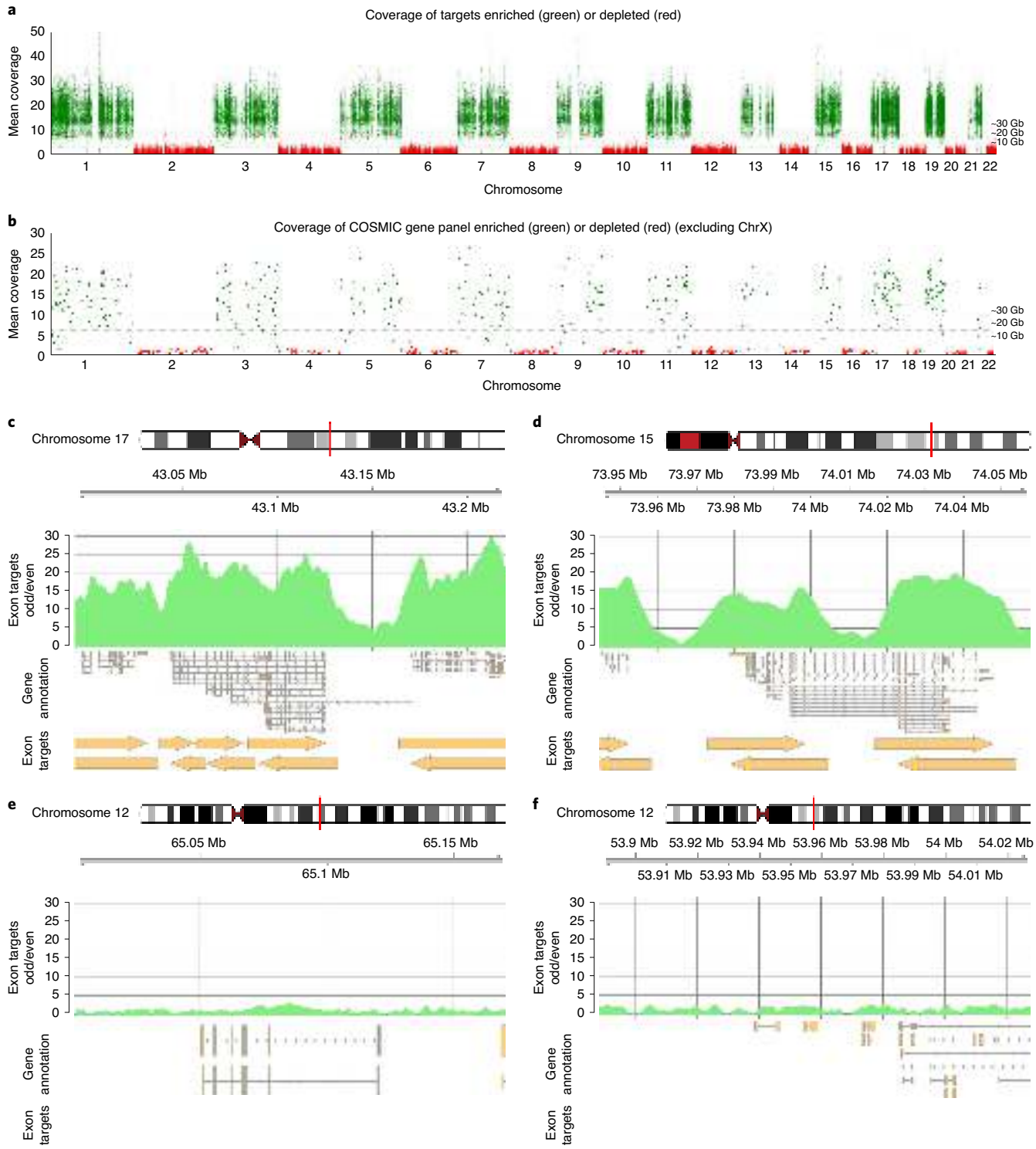


Fig. 4 | Half-exome panel targeted sequencing. a, The mean coverage across each exon target in the genome ordered by chromosome. Exons on odd-numbered chromosomes are enriched (green) and those on even-numbered chromosomes are depleted (red). **b**, The mean coverage across each exon for genes within the COSMIC panels. In **a** and **b**, the horizontal lines represent the mean expected coverage for flow cells yielding ~10, ~20 or ~30 Gb of data in a single run. The mean coverage was calculated by mosdepth[23]. **c**–**f**, Coverage plots for the highlighted genes including *BRCA1* (**c**), *PML* (**d**), *WIF1* (**e**), and *HOXC13* and *HOXC11* (**f**). The targets in **c** and **d** are enriched as they are found on chromosomes 17 and 15 while those in **e** and **f** are depleted as the genes are on chromosome 12. Exon target regions are indicated by arrows. In this experiment, different targets were used for the Watson and Crick strands as illustrated by the offsets. Note the absence of target regions in **e** and **f**.

(Supplementary Fig. 10 and Supplementary Table 2). Gene targets had a median coverage of 42.3× (mean 40.5×; Fig. 5b), with 75% of genes >38× and 25% of genes >44×. To test the performance of readfish on non-ONT hardware, we ran the same experiment using an NVIDIA GeForce GTX 1080 GPU using the fast model of the base-caller. This run generated only 6.7 Gb of data with a read N50

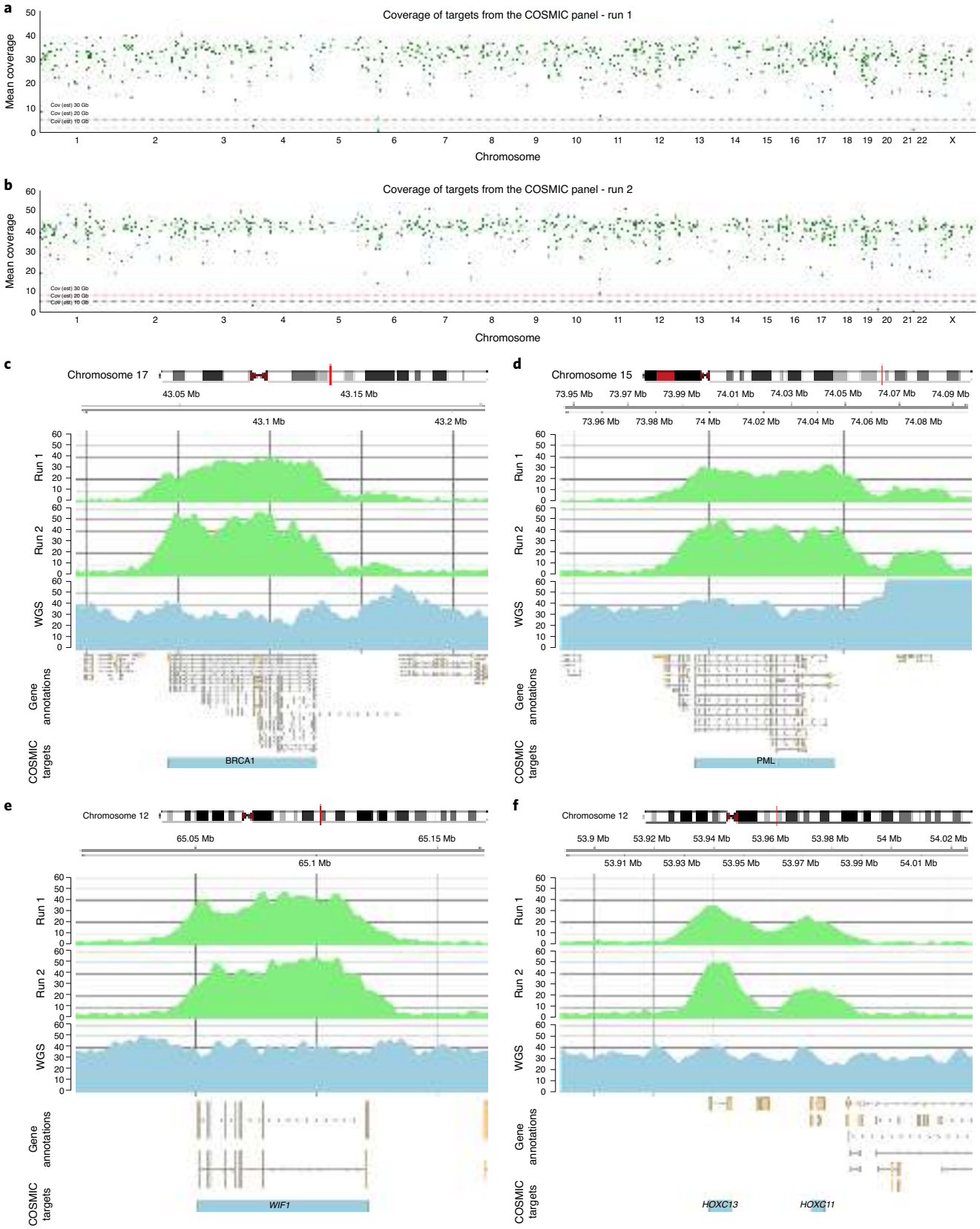


Fig. 5 | COSMIC panel targeted sequencing. a,**b**, The mean coverage across the selected COSMIC gene regions ordered by chromosome for two independent sequencing runs of NA12878. The horizontal lines represent the mean expected coverage for flow cells yielding ~10, ~20 or ~30 Gb of data in a single run. The mean coverage was calculated by mosdepth[23]. **c**–**f**, Coverage plots from each run (light green) for the highlighted genes including *BRCA1* (**c**), *PML* (**d**), *WIF1* (**e**), and *HOXC13* and *HOXC11* (**f**). For comparison, the coverage in the same regions for a 35× whole-genome-sequenced (WGS) nanopore run is shown in blue. COSMIC target regions are indicated by blue bars and include intronic sequences.

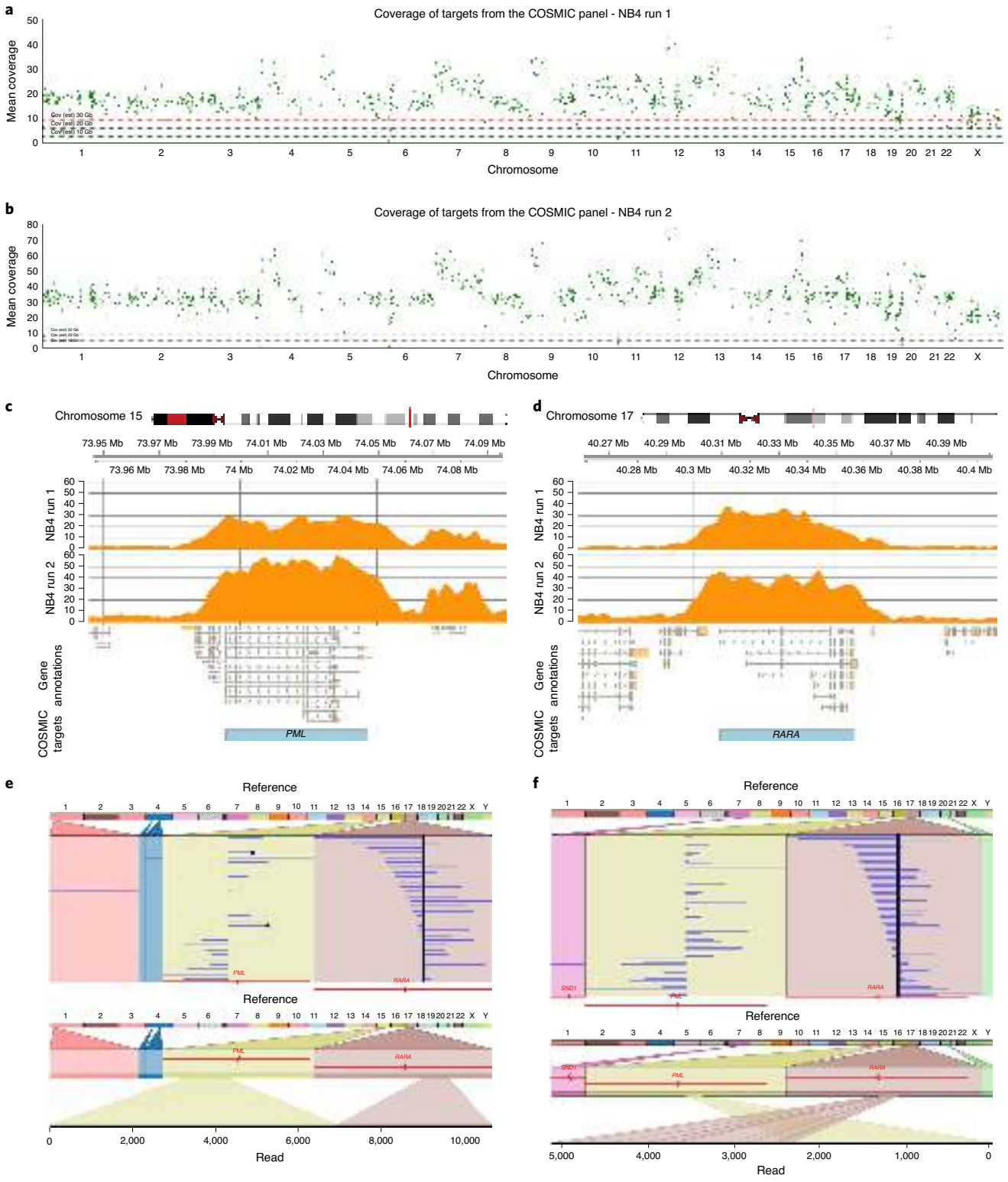


Fig. 6 | COSMIC panel targeted sequencing of NB4. a,**b**, The mean coverage across each of the COSMIC target regions ordered by chromosome for two independent sequencing runs of the NB4 cell line. The horizontal dashed line indicates the expected coverage from a flow cell yielding ~10, ~20 or ~30 Gb of sequence data in a single run. **c**,**d**, Coverage plots for each NB4 sequencing run shown in orange for *PML* (**c**) and *RARA* (**d**). **e**,**f**, Reads mapping to chromosomes 15 and 17 derived from the NB4 cell line runs 1 and 2 respectively, indicating the fusion between *PML* and *RARA*. Mappings of example individual reads are shown. Break points were identified using SVIM; visualizations were performed using Ribbon[20].

of 799 bases (Supplementary Fig. 11 and Supplementary Table 2). The median coverage of genes was 19.6× (mean 19.1×), with 75% of genes >17.78× and 25% of genes >20.99×.

The difference in the yield between these runs is largely due to flow cell variation, particularly for the third run, which showed unusual flow cell activity (Supplementary Fig. 12). However, normalizing

the enrichment to the total yield of each flow cell shows a similar performance in each experiment for a selection of target genes including *PML*, *WIF1*, *HOXC11/C13*, *RARA* and *BRCA1* (Supplementary Figs. 13–17). This suggests that any steps taken to maximize the yield, such as flushing, will result in enhanced enrichment. As with any native nanopore sequence data, these data can be used to assess structural variants and nucleotide variation. As shown in Supplementary Table 3, these data show recall and precision equivalent to, or better than, reference nanopore whole-genome data at a similar coverage without targeting[10]. Structural variants within the targeted regions can be detected with high recall (Supplementary Table 4). Crucially, between 5 and 10 typical flow cells would be required to generate equivalent coverage without Read Until.

To test screening for structural variants, we used the NB4 acute promyelocytic leukemia cell line[9]. Using the same COSMIC panel, we identified the translocation using a flow cell with only 1,196 pores, generating 4.5 Gb of sequence data in under 15 h (Supplementary Fig. 18). The median coverage of targets was 11.46× (mean 11.78×; Fig. 6a,c,d), with 75% of genes >9.5× and 25% of genes >13.4×. Analysis with SVIM looking for break-point ends, ignoring in/dels, identified two candidates passing default filtering (see Methods)[20]. The break point can also be detected with Sniffles (data not shown)[21]. Of these candidates, one captured the known break point supported by six reads. A further 24 h of sequencing (~3 Gb) resulted in a median coverage of 17.37× (mean 18×) and 9 reads supporting the variant (Fig. 6e and Supplementary Table 5). No complex rearrangements were reported in NA12878 using the same COSMIC panel (Supplementary Table 5). A subsequent repeat of this experiment (Supplementary Fig. 19), with flushing every 24 h, generated 15.9 Gb of sequence data. The median coverage of targets was 34× (mean 35.5×; Fig. 6b), with 75% of genes >30×, 25% of genes >38× and 23 reads supporting the break point (Fig. 6f and Supplementary Table 5).

Discussion

The idea of selectively sequencing (Read Until) individual molecules using only computational methods is a unique capability of nanopore sequencing[1]. Here we exploit ONT tools to provide a true real-time stream of sequence data as nucleotide bases and provide a toolkit to design and control selective sequencing experiments called readfish. This approach removes the need for complex signal mapping algorithms but does require a sufficiently fast base-caller. Previous work illustrated that this method was feasible, but required extensive additional computation and did not show significant enrichment over throughput achieved without running Read Until[4]. Here we demonstrate real enrichment over that expected from a similar control flow cell. We also show that standard techniques for enhancing the flow cell yield such as nuclease flushing and loading additional library are similarly beneficial for Read Until experiments. Although not extensively exploited here, nuclease flushing and reuse of flow cells do increase the yield and enrichment, and we have taken to flushing Read Until experiments every 24 h.

We find that increased rejection of reads on a flow cell negatively impacts the sequencing yield and so observed enrichment. The main benefit of selective sequencing in metagenomics and host depletion is to improve time-to-answer. For samples that sequence well (that is, do not tend to block the flow cell), additional enrichment benefits may be observed. Notably, running selective sequencing does not disrupt the proportion of reads by count that map to a specific reference. Thus, for metagenomics, it is still possible to assess the relative abundance while focusing sequencing length on specific subsets of reads. Future methods proposed by ONT to address blocking, such as onboard nucleases, might increase the throughput in future.

The key benefit of our approach is that we utilize only computational resources available in the GridION Mk1. As we use current commercially provided base-callers, we can utilize new algorithms and pores as they are developed. Thus, although not yet tested, we could use this method on RNA if sufficiently long reads require depletion. Similarly. we could use methylation-aware base-callers to sequence regions of DNA starting from either high- or low-methylation regions. As we obtain sequence, rather than signal, we greatly simplify the construction of pipelines for downstream analysis of reads. Although we focus on results for the GridION Mk1, we show that this method can be used with any MinION configuration provided there is sufficient available GPU to base-call a sequencing run in real time (Supplementary Note 1). As we show here, it is possible to utilize the fast base-calling model and obtain effective enrichment using a single NVIDIA GeForce GTX 1080 GPU. Other users have reported success with the high-accuracy model on systems configured with NVIDIA 2080 GPUs (J. Tyson, personal communication). In cost terms, any platform capable of real-time base-calling will be compatible with our approach. In principle, this method should scale to the PromethION platform.

We demonstrate that selective sequencing of arbitrary targeted regions of the human genome results in actionable coverage and can identify single-nucleotide variants and structural variants in the COSMIC panel. For structural variant analysis, DNA extraction, library preparation, sequencing and analysis could be completed within 24 h. When sequencing a subset of a large genome, large numbers of off-target reads are sampled while detecting those of interest and the precise parameters of optimal target size and coverage have yet to be defined. Consequently, library preparation methods enriching for regions of interest will result in a higher coverage than Read Until. However, the design of such panels is relatively costly and inflexible once developed. Methods relying on amplification result in the loss of methylation data, which can be found using the methods presented here.

In readfish selective sequencing, targets can be updated by a single configuration file. Developing a new panel is as straightforward as compiling a list of target regions. Here we also illustrate the concept of adaptive sequencing, as in our metagenomics examples, where targets can be dynamically adjusted during a run. In theory, a panel could be updated in response to observations of the data in real time, perhaps adding targets where candidate novel structural variants have been identified or removing targets where sufficient evidence is available to eliminate the possibility of a structural variant existing.

Of course, throughput achievable on platforms such as PromethION at scale provides efficient whole-genome sequencing[22]. Thus, any effective method for enrichment must be as efficient, including the additional computation required. By utilizing the available GPU computational capacity during the sequencing run, we address this issue. There is no reason, in theory, why samples could not be multiplexed on a single flow cell as long as sufficient yield can be obtained to address the biological question.

Although we have focused exclusively on applications for Read Until, we believe that a real-time sequence data stream as bases has significant advantages for future pipelines. If sequence data can be streamed directly into an analysis pipeline and conclusions drawn without the requirements for data storage, then field deployment of sequencing for detection of specific sequences might be accelerated. Ultimately, it may be possible to stream sequence data for calling of structural variants and further analysis in real time.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of

data and code availability are available at https://doi.org/10.1038/s41587-020-00746-x.

Received: 7 February 2020; Accepted: 21 October 2020;
Published online: 30 November 2020

References

1. Loose, M., Malla, S. & Stout, M. Real-time selective sequencing using nanopore technology. *Nat. Methods* **13**, 751–754 (2016).
2. Masutani, B. & Morishita, S. A framework and an algorithm to detect low-abundance DNA by a handy sequencer and a palm-sized computer. *Bioinformatics* **35**, 584–592 (2019).
3. Kovaka, S., Fan, Y., Ni, B., Timp, W. & Schatz, M. C. Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-020-0731-9 (2020).
4. Edwards, H. S. et al. Real-time selective sequencing with RUBRIC: Read Until with Basecall and Reference-Informed Criteria. *Sci. Rep.* **9**, 11475 (2019).
5. Rang, F. J., Kloosterman, W. P. & de Ridder, J. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biol.* **19**, 90 (2018).
6. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
7. Kim, D., Song, L., Breitwieser, F. P. & Salzberg, S. L. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**, 1721–1729 (2016).
8. Tate, J. G. et al. COSMIC: the Catalogue of Somatic Mutations in Cancer. *Nucleic Acids Res.* **47**, D941–D947 (2019).
9. Mozziconacci, M.-J. et al. Molecular cytogenetics of the acute promyelocytic leukemia-derived cell line NB4 and of four all-trans retinoic acid–resistant subclones. *Genes Chromosomes Cancer* **35**, 261–270 (2002).
10. Jain, M. et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338–345 (2018).
11. Charalampous, T. et al. Nanopore metagenomics enables rapid clinical diagnosis of bacterial lower respiratory infection. *Nat. Biotechnol.* **37**, 783–792 (2019).
12. Marotz, C. A. et al. Improving saliva shotgun metagenomics by chemical host DNA depletion. *Microbiome* **6**, 42 (2018).
13. Nicholls, S. M., Quick, J. C., Tang, S. & Loman, N. J. Ultra-deep, long-read nanopore sequencing of mock microbial community standards. *Gigascience* **8**, giz043 (2019).
14. Kolmogorov, M., Yuan, J., Lin, Y. & Pevzner, P. A. Assembly of long, error-prone reads using repeat graphs. *Nat. Biotechnol.* **37**, 540–546 (2019).
15. Kozarewa, I., Armisen, J., Gardner, A. F., Slatko, B. E. & Hendrickson, C. L. Overview of target enrichment strategies. *Curr. Protoc. Mol. Biol.* **112**, 7.21.1–7.21.23 (2015).
16. Gnirke, A. et al. Solution hybrid selection with ultra-long oligonucleotides for massively parallel targeted sequencing. *Nat. Biotechnol.* **27**, 182–189 (2009).
17. Gilpatrick, T. et al. Targeted nanopore sequencing with Cas9-guided adapter ligation. *Nat. Biotechnol.* **38**, 433–438 (2020).
18. Loose, M. Finding the needle: targeted nanopore sequencing and CRISPR-Cas9. *CRISPR J.* **1**, 265–267 (2018).
19. Cunningham, F. et al. Ensembl 2019. *Nucleic Acids Res.* **47**, D745–D751 (2019).
20. Heller, D. & Vingron, M. SVIM: structural variant identification using mapped long reads. *Bioinformatics* **35**, 2907–2915 (2019).
21. Sedlazeck, F. J. et al. Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Meth.* **15**, 461–468 (2018).
22. Beyter, D., Ingimundardottir, H. & Eggertsson, H. P. Long read sequencing of 1,817 Icelanders provides insight into the role of structural variants in human disease. Preprint at *bioRxiv* https://doi.org/10.1101/848366 (2019).
23. Pedersen, B. S. & Quinlan, A. R. Mosdepth: quick coverage calculation for genomes and exomes. *Bioinformatics* **34**, 867–868 (2018).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2020

Methods

Library preparation and sequencing. Standard LSK-109 (ONT) sequencing libraries were prepared from either the ZymoBIOMICS HMW DNA Standard (DS6322 ZymoBIOMICS) or DNA extracted from GM12878 cells (Coriell) or NB4 cells (gift from M. Hubank) as described in Jain et al.[10]. Human DNA for exon enrichment or gene targeting was sheared to approximately 12 kb using g-TUBE (Covaris). Sequencing runs used either the GridION Mk1 or a MinION with an NVIDIA GeForce GTX 1080 GPU (see Supplementary Table 2). Standard scripts for sequencing were used with one modification, namely that the size of the data chunk delivered by MinKNOW was reduced from 1 s to 0.4 s by changing the value of the break_reads_after_seconds parameter in the relevant TOML file (located in ../minknow/conf/package/sequencing/ for MinKNOW core version 3.6). All sequencing used FLO-MIN106 R9.4.1 flow cells.

When running Read Until experiments seeking to maximize the yield, throughput on the flow cell should be monitored closely. Our practice has been to nuclease flush flow cells every 24 h to maximize throughput. For maximizing occupancy on the flow cell, users should experiment with loading more library than they might otherwise do. For example, where a user might load 400 ng of library with a read length N50 of 10–15 kb, we would recommend loading 600 ng of library. This assumes R9.4 flow cells. This protocol has not yet been tested on R10.

Detection of single-nucleotide variants. Single-nucleotide polymorphisms (SNPs) in NA12878 read data were called using Nanopolish in methylation-aware mode[23]. Reads were mapped to hg38 removing ALTs with minimap2 using standard settings for ONT reads[6]. High-confidence gold-standard SNPs were identified from the Genome In A Bottle truth set[24]. SNPs were compared with a 35× WGS NA12878 reference set recalled using the same Guppy base-caller model[10]. SNP comparisons were made using hap.py with default settings and the same target sites used for selective sequencing (https://github.com/Illumina/hap.py).

Structural variant detection and concordance. Reads were mapped to the hg38 primary assembly with minimap2 and standard ONT settings. Variants were called using SVIM and Sniffles with default settings and the minimum variant length set as 50 (refs. [20,21]). Only SVIM variant calls with QUAL above 10 and longer than 50 bp were kept. Variants of the same type present in both SVIM and Sniffles call sets were selected as the final call set using SURVIVOR and a maximal distance between break points was set to 500 (ref. [25]). Only insertions and deletions intersecting the COSMIC target panel were considered for concordance calculations in the whole-genome-sequence dataset, run 1 and run 2. Concordance calculations were performed with Truvari (https://github.com/spiralgenetics/truvari) with the reference distance set as 1.5 kb and the percentage size similarity set as 0.3, and only insertions and deletions larger than 50 bp within the COSMIC target panel were considered. For analysis of the translocation in the NB4 cell lines, variant calls were filtered with a quality score of 10 and non-BND (break-point end) structural variant types were ignored. Structural variants were visualized with Ribbon[26].

Target lists. The exact target list used to configure exon capture can be obtained at http://jan2020.archive.ensembl.org/biomart/martview/59d93fb27bdffa53152236c6cb12c4b1?VIRTUALSCHEMANAME=default&ATTRIBUTES=hsapiens_gene_ensembl.default.structure.ensembl_gene_id%7Chsapiens_gene_ensembl.default.structure.ensembl_gene_id_version%7Chsapiens_gene_ensembl.default.structure.ensembl_transcript_id%7Chsapiens_gene_ensembl.default.structure.ensembl_transcript_id_version%7Chsapiens_gene_ensembl.default.structure.chromosome_name%7Chsapiens_gene_ensembl.default.structure.exon_chrom_start%7Chsapiens_gene_ensembl.default.structure.exon_chrom_end&FILTERS=hsapiens_gene_ensembl.default.filters.biotype.%22protein_coding%22%7Chsapiens_gene_ensembl.default.filters.chromosome_name.%221,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,X,Y%22%7Chsapiens_gene_ensembl.default.filters.with_hgnc_trans_name.only&VISIBLEPANEL=attributepanel.

Read Until cache configuration and chunk size. A read begins with adapter sequences as well as optional barcodes. Additionally, read starts sometimes stall as DNA engages with the pore before signal-containing sequence data are available. The first chunk of data may not provide an optimal base-call and additional data may be required. Calling any single fragment of data in isolation is less informative than calling the entire signal, and so we implement a read cache concatenating adjacent signal data from the same read. This enables base-calling the complete signal for each read since it started. As of MinKNOW version 3.6, the sequencing platform is effectively limited to a lower-bound chunk size of 0.4 s. As shown in Supplementary Fig. 2 and Supplementary Table 1, more than 80% of human reads can be base-called and aligned within 2 chunks or 0.8 s worth of data. For bacterial sequences, more than 40% of reads can be base-called and aligned within a single chunk or 0.4 s worth of data. Thus, by observation, the smallest possible chunk size will enable the fastest decision-making for any given sequence. In a typical experiment, we find that 90% of reads can be processed (called, mapped and decision made) within three chunks (1.2 s; Supplementary Fig. 2 and Supplementary Table 1).

Base-caller configuration. The Guppy base-caller contains several models for base-calling that trade speed (fast) for accuracy (high-accuracy model, hac) and can optionally call methylation. For selective sequencing, the goal is speed, and so we investigated the efficacy of both the fast and hac models, finding the GridION Mk1 easily powerful enough to use the hac model. Across all experiments shown here, the average batch of reads was called in 0.28 s and contained 30 reads. At the maximum load, individual reads are processed in less than 0.002 s. Thus, we call at least 100 read fragments per second and even at the peak load can typically call all 512 reads (see Supplementary Figs. 3–7 and 10).

Experimental configuration. Depending on the configuration of the experiment, the response to read mapping varies (see Methods). If depleting contaminants (host depletion), then reads mapping to that reference should be rejected. For enrichment, reads mapping to a target should be sequenced. The action for non-mapping reads will depend on the experiment. If the experimental goal is enriching low-abundance or unknown targets, non-mapping reads should be sequenced. If enriching for subsets of a known reference, non-mapping reads might be rejected in favor of sampling more. Given the variety of options, we provide a configuration file allowing any mapping result to trigger any action. We include the option to dynamically update this file during sequencing, enabling target switches while sequencing. The configuration also allows different experiments on regions of the same flow cell (see https://github.com/LooseLab/readfish/blob/master/TOML.md).

readfish code availability. The ONT Read Until API is required for running Read Until. The results presented here used an updated version of this API, available from our GitHub (https://github.com/LooseLab/read_until_api_v2; Git commit cff0f52). These changes were required for Python3 compatibility and also change the behavior of the read cache, enabling consecutive chunks of data to be stored for calling. As the ONT tool chain matures to Python3, such changes will no longer be required. pyguppyclient (v.0.0.5), a python interface to the Guppy base-calling server, is currently available on PyPI. Our code is available open source at http://www.github.com/LooseLab/readfish and installable via PyPI.

readfish scripts. readfish is a set of scripts that control sequencing in real time. Each script is accessed as a sub-command, and a description is given below.

targets. This script runs the core Read Until process as specified in the experiment's TOML file. It can select specific regions of a genome, mapping reads in real time using minimap2 and rejecting reads appropriately. This script should be started once the initial mux scan has completed. The experiment's TOML file can be updated during a sequencing run to change the configuration of the Read Until process. It is through this mechanism that the align and centrifuge commands can change Read Until behavior during a run. The configuration parameters are available under the help flag. Tables 1 and 2 describe the mapping parameters and configuration options for various possible experiment types.

align. This script runs an instance of the 'Run Until' monitoring system that watches as completed reads are written to disk. When new data are detected, this pipeline will map the data against the target reference genome (specified in the experiment's TOML file) and compute the cumulative coverage for the sequencing run. Once a genomic target reaches sufficient coverage, it will be added to the unblock list. Optionally, the user can provide additional targets from the start of the run to implement 'host depletion'. Finally, the user can configure 'align' to stop the entire run if all samples have reached the required coverage depth. At present, this coverage depth is uniform for all samples, so it is not possible to have variable coverage over a target set.

centrifuge. This script runs an instance of the 'Run Until' monitoring system. As completed reads are written to disk, this program (Supplementary Fig. 1c) will classify the reads using centrifuge and a user-defined index. When 2,000 reads are uniquely classified, the corresponding reference genome is downloaded from RefSeq[27] and incorporated into a minimap2 index. At this point, the same process

Table 1 | Description of possible read mapping conditions

Mapping condition	Description
multi_on	The read fragment maps to multiple locations including a region of interest.
multi_off	The read fragment maps to multiple locations not including a region of interest.
single_on	The read fragment maps only to a region of interest.
single_off	The read fragment maps to one location but it is not a region of interest.
no_map	The read fragment does not map to the reference.
no_seq	No sequence was obtained for the signal fragment.

Table 2 | Example configurations for different experiment types

Experiment type	Region of interest for alignments	Mapping condition					
		multi_on	multi_off	single_on	single_off	no_map	no_seq
Host depletion	Known host genome	unblock	proceed	unblock	proceed	proceed	proceed
Targeted sequencing	Known regions from one or more genomes	stop receiving	proceed	stop receiving	unblock	proceed	proceed
Target coverage depth (known sample composition)	All known genomes within the sample, tracked for coverage depth	stop receiving	proceed	stop receiving	unblock	proceed	proceed
Low-abundance enrichment (unknown sample composition)	All genomes within the sample that can be identified as well as those that cannot	stop receiving	proceed	stop receiving	unblock	proceed	proceed

'unblock' causes a read to be ejected from the pore; 'proceed' means that a read continues to sequence and serve data through the API for later decisions; 'stop receiving' allows the read to continue sequencing with no further data served through the API.

as in 'align' is used to determine the coverage depth. The new alignment index is passed to the core Read Until script ('targets') by updating the experiment's TOML file, allowing dynamic updates for both the unblock list and the genomic reference.

unblock-all. This script is provided as a test of the Read Until API where all incoming read fragments are immediately unblocked. It allows a user to quickly determine whether their MinKNOW instance is able to provide and process unblock signals at the correct rate. Users should provide a bulk FAST5 file for playback for this testing process.

validate. This script is a standalone tool for validating an experiment's TOML file. We provide an ru_schema.json (https://github.com/LooseLab/readfish/blob/14ddf60c60c2697e86cf870f406751c7cd26daf8/ru/static/readfish_toml.schema.json) file that describes the required configuration format.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

All reads generated in the course of this study are available from the ENA under project ID PRJEB36644.

Code availability

Our code is available open source at http://www.github.com/LooseLab/readfish. See also "readfish code availability" above.

References

24. Zook, J. M. et al. An open resource for accurately benchmarking small variant and reference calls. *Nat. Biotechnol.* **37**, 561–566 (2019).
25. Jeffares, D. C. et al. Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat. Commun.* **8**, 14061 (2017).
26. Nattestad, M., Aboukhalil, R., Chin, C.-S. & Schatz, M. C. Ribbon: intuitive visualization for complex genomic variation. *Bioinformatics* https://doi.org/10.1093/bioinformatics/btaa680 (2020).
27. Pruitt, K. D. & Maglott, D. R. RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Res.* **29**, 137–140 (2001).

Acknowledgements

We thank J. Quick, J. Tyson, J. Simpson and N. Loman for helpful comments and (mainly) criticisms and E. Birney, N. Goldman and A. Senf for helpful insights and discussion on these approaches. We thank M. Hubank and L. Gallagher for access to materials and reagents as well as general boundless enthusiasm. We thank M. Jain for assisting in manipulating data. We also thank S. Reid, C. Wright, C. Seymour, J. Pugh and G. Pimm from ONT for advice on MinKNOW and Guppy operations as well as extensive troubleshooting. This work was supported by the Biotechnology and Biological Sciences Research Council (grant numbers BB/N017099/1, R.M. and M.L.; BB/M020061/1, M.L.; and BB/M008770/1, 1949454 A.P.), the Wellcome Trust (grant number 204843/Z/16/Z, N.H. and M.L.) and the Defence Science and Technology Laboratory (grant number DSTLX-1000138444, R.M. and M.L.).

Author contributions

M.L. and A.P. conceived the study. A.P., N.H. and M.L. acquired data. T.C. and R.M. designed and implemented metagenomics applications. A.P., B.J.D. and M.L. analyzed and interpreted data. All authors discussed the results and contributed to the final manuscript.

Competing interests

M.L. was a member of the MinION access program and has received free flow cells and sequencing reagents in the past. M.L. has received reimbursement for travel, accommodation and conference fees to speak at events organized by ONT.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41587-020-00746-x.

Correspondence and requests for materials should be addressed to M.L.

Peer review information *Nature Biotechnology* thanks Jan Korbel and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

Rapid-CNS^2: Rapid comprehensive adaptive nanopore-sequencing of CNS tumors, a proof of concept study

Areeba Patel[1,2]*, Helin Dogan[1,2]*, Alexander Payne[3], Philipp Sievers[1,2], Natalie Schoebe[1,2], Daniel Schrimpf[1,2], Damian Stichel[1,2], Nadine Holmes[3], Philipp Euskirchen[4], Jürgen Hench[5], Stephan Frank[5], Violaine Rosenstiel-Goidts[6], Miriam Ratliff[7], Nima Etminan[7], Andreas Unterberg[8], Christoph Dieterich[9], Christel Herold-Mende[8], Stefan M Pfister[10,11,12], Wolfgang Wick[14], Matthias Schlesner[15], Matthew Loose[3], Andreas von Deimling[1,2], Martin Sill[10,11]*, David TW Jones[10,13]*, Felix Sahm[1,2,10]*

Affiliations

[1]Dept. of Neuropathology, University Hospital Heidelberg, Heidelberg, Germany.

[2]Clinical Cooperation Unit Neuropathology, German Cancer Consortium (DKTK), German Cancer Research Center, Heidelberg, Germany.

[3]DeepSeq, School of Life Sciences, University of Nottingham, Nottingham, United Kingdom,

[4] Department of Neurology, Charité-Universitätsmedizin Berlin, Berlin, Germany.

[5] Division of Neuropathology, Institute of Pathology, University Hospital Basel, Basel, Switzerland.

[6]Brain Tumor Translational Targets, German Cancer Research Center (DKFZ), Heidelberg, Germany.

[7]Dept. of Neurosurgery, University Hospital Mannheim, Mannheim, Germany.

[8]Dept. of Neurosurgery, University Hospital Heidelberg, Heidelberg, Germany.

[9]Department of Cardiology, Angiology, and Pneumology, University Hospital Heidelberg, University of Heidelberg, Heidelberg, Germany.

[10]Hopp Children's Cancer Center (KiTZ), Heidelberg, Germany.

[11]Division of Pediatric Neurooncology, German Cancer Research Center (DKFZ) and German Cancer Consortium (DKTK), Heidelberg, Germany

[12]Department of Pediatric Hematology and Oncology, Heidelberg University Hospital, Heidelberg, Germany

[13]Pediatric Glioma Research Group, German Cancer Research Center (DKFZ), Heidelberg, Germany

[14]Clinical Cooperation Unit Neurooncology, German Consortium for Translational Cancer Research (DKTK), German Cancer Research Center (DKFZ), Heidelberg, Germany and Department of Neurology and Neurooncology Program, National Center for Tumor Diseases, Heidelberg University Hospital, Heidelberg, Germany

[15]Biomedical Informatics, Data Mining and Data Analytics, Augsburg University, Augsburg, Germany

NOTE: This preprint reports new research that has not been certified by peer review and should not be used to guide clinical practice.

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

Abstract

Background:

The 2021 WHO classification of central nervous system tumors includes multiple molecular markers and patterns that are recommended for routine diagnostic use in addition to histology. Sequencing infrastructures for complete molecular profiling require considerable investment, while batching samples for sequencing and methylation profiling can delay turnaround time. We introduce RAPID-CNS^2, a nanopore adaptive sequencing pipeline that enables comprehensive mutational, methylation and copy number profiling of CNS tumours with a single, cost-effective sequencing assay. It can be run for single samples and offers highly flexible target selection that can be personalized per case with no additional library preparation.

Methods:

Utilizing ReadFish, a toolkit enabling targeted nanopore sequencing without the need for library enrichment, we sequenced DNA from 22 diffuse glioma samples on a MinION device. Target regions comprised our Heidelberg brain tumor NGS panel and pre-selected CpG sites for methylation classification using an adapted random forest classifier. Pathognomonic alterations, copy number profiles, and methylation classes were called using a custom bioinformatics pipeline. The resulting data were compared to their corresponding standard NGS panel sequencing and EPIC methylation array results.

Results:

Complete concordance with the EPIC array was found for copy number profiles. The vast majority (94%) of pathognomonic mutations were congruent with standard NGS panel-seq data. *MGMT* promoter status was correctly identified in all samples. Methylation families from the random forest classifier were detected with 96% congruence. Among the alterations decisive for rendering a WHO 2021 classification-compatible integrated diagnosis, 97% of the alterations were consistent over the entire cohort (completely congruent in 19/22 cases and sufficient for unequivocal diagnosis in all 22 samples).

Conclusions:

RAPID-CNS^2 provides a swift and highly flexible alternative to conventional NGS and array-based methods for SNV/InDel analysis, detection of copy number alterations, target gene methylation analysis (e.g. *MGMT*) and methylation-based classification. The turnaround time of ~5 days for this proof-of-concept study can be further shortened to < 24h by optimizing target sizes and enabling real-time computational analysis. Expected advances in nanopore sequencing and analysis hardware make the prospect of integrative molecular diagnosis in an intra-operative setting a feasible prospect in future. This low-capital approach would be cost-efficient for low throughput settings or in locations with less sophisticated laboratory infrastructure, and invaluable in cases requiring immediate diagnoses.

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

Molecular markers are now unequivocally a requirement for integrative brain tumor diagnostics. The 2021 WHO classification of CNS tumors substantially increases the set of genes required in routine evaluation, and significantly increases the relevance of DNA methylation analysis in the diagnostic process [10]. Multiple approaches are available for such analyses. However, neuropathology labs cannot rely on current off-the-shelf products, since these do not cover all genes relevant for neuro-oncology, while including large target regions that are dispensable. Thus, custom assays have typically been set-up in neuropathology labs where the equipment for next-generation sequencing (NGS) is available. In turn, the advantages of custom neuropathology NGS panels can only be efficiently exploited when case numbers are sufficient for batchwise processing. Labs with lower specimen submission numbers hence may have to pool samples over multiple weeks. Here we introduce RAPID-CNS^2 - a custom neurooncology molecular diagnostic workflow using third generation sequencing for parallel copy-number profiling, mutational and methylation analysis that is highly flexible in target selection, runs efficiently on single samples, and can be initiated immediately upon receipt of frozen sections.

Nanopore sequencing has an advantage over current NGS methods in terms of longer read lengths, shorter and easier library preparation protocols, ability to call base modifications natively from extracted nucleic acids, real time analysis, and portablility of sequencing devices – all at relatively low cost [3]. However, smaller devices like the MinION yield low-coverage data when run genome-wide, that makes it difficult to detect pathognomonic genetic alterations or hard-to-map regions like the *TERT* promoter [6]. Nanopore provides a "ReadUntil" adaptive sampling toolkit that can reject reads in real-time during sequencing [7]. ReadFish harnesses this functionality to enable targeted adaptive sequencing with no additional steps in library preparation [9]. This considerably increases coverage over "target" regions by real-time enrichment during sequencing, to allow confident detection of clinically relevant alterations.

RAPID-CNS^2 leverages adaptive nanopore sequencing through ReadFish and is run here as a proof-of-concept using a portable MinION device. We formulated target regions covering the Heidelberg brain tumour NGS panel and CpG sites required for methylation-based classification [4, 11]. We performed ReadFish-based sequencing on 22 diffuse glioma samples that had previously undergone brain tumor NGS panel and Infinium MethylationEPIC array (EPIC) analysis [2, 4, 11]. Samples were selected to cover a variety of the most clinically-relevant pathognomonic alterations (*IDH1*, 1p/19q codeletion, chr7 gain/chr10 loss, *TERT* promoter, *EGFR* amplification, *CDKN2A/B* deletion, *MGMT* status) and relevant methylation classes identified by conventional methods. Cryoconserved brain tumour tissue was prepared for Nanopore sequencing with the SQK-LSK109 Ligation Sequencing Kit from ONT. Incubation time and other parameters were optimized to improve quality, amount of data generated and on-target rate of the libraries (Supplementary methods). Single samples were loaded onto FLO-MIN106 R9.4.1 flow cells and run on a MinION 1B. ReadFish controlled the sequencing in real-time and was run using a consumer notebook powered by an 8GB NVIDIA RTX 2080 Ti GPU. Samples were sequenced for up to 72 hours. Our selected target regions covered 5.56% of the entire genome. Sequencing time can be reduced to less than 24 hours by further optimizing the size of the targeted regions. Sequenced data was analysed using a bioinformatics pipeline customized for neurooncology targets (which will be available on https://github.com/areebapatel/RAPID-CNS2). SNVs were filtered for clinical relevance by their 1000 genomes population frequency (<0.01) and COSMIC annotations [1, 13, 14]. Copy number alterations were estimated using depth-of-coverage of the mapped reads [12]. Nanopore sequencing provides the additional advantage of natively estimating base modifications from a single DNA sequencing assay. Methylated bases were identified using

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

megalodon, a deep neural network-based modified base caller [8]. Megalodon's output was used to compute methylation values over targeted CpG sites and assess *MGMT* promoter methylation status. A random forest classifier based on the previously published reference set [4] was trained to predict methylation classes for the samples. *MGMT* promoter methylation status was assigned by averaging methylation values over all CpG sites in the *MGMT* promoter region (Supplementary results). Mean run time from tissue collection to reporting for RAPID-CNS^2 was < 5 days. Nanopore sequencing considerably reduced library preparation time to 3.5 hours, as opposed to 48 hours for panel sequencing and 72 hours for the EPIC array (Figure 1). Additionally, it merged both data categories (sequencing and methylation) into one lab workflow. Despite the differences in sequencing technology and method-specific data analysis pipelines, congruence of detected SNVs, regardless of clonality and clinical relevance, was 78% (Supplementary data). Importantly, diagnostically relevant, pathognomonic mutations like *IDH1* R132H/S and *TERT* promoter were congruent in 22/22 and 19/22 samples respectively (Figure 1b). In addition, we derived copy-number-plots (CNP) from calculated copy number levels for the Nanopore data (Supplementary figure 1a). Plots generated using Nanopore data displayed markedly better resolution than those obtained using panel sequencing data (Supplementary figure 1b). Complete concordance with EPIC array analysis was found for CNV levels in all samples. RAPID-CNS^2 also enabled gene-level CNV detection (Supplementary data). Among the alterations decisive for rendering an integrated molecular diagnosis, 217/220 were consistent over the entire cohort (completely congruent in 19/22 cases).

Including CpG sites relevant for methylation-based classification in the ReadFish targets also allowed for methylation class prediction. The ability of nanopore sequencing to reliably provide a methylation classification using low-pass whole genome sequencing has previously been demonstrated by nanoDx [5]. Methylation families predicted by RAPID-CNS^2 (the level most relevant for treatment decisions) matched their corresponding EPIC array-based classification in 21/22 cases, while precise methylation sub-classes were concordant in 14 cases. *MGMT* promoter status was also congruent with its corresponding EPIC array analysis for all cases [2]. Nanopore identified the *MGMT* promoter status as unmethylated in one sample in line with the EPIC array, which was assigned as methylated by pyrosequencing.

Targeted regions for RAPID-CNS^2 can be easily altered by editing a BED file, in principle allowing lower sequencing times than in this study. With no additional library preparation steps required, it is possible to modify targeted regions for each individual sample as required. The MinION is a portable, handheld device which makes it a rational option for smaller neuropathology labs or in lower-infrastructure locations. While we used a GPU to run ReadFish, it can also be run using a sufficiently powerful CPU. Collectively, the RAPID-CNS^2 approach can be set-up at low capital expense, is cost-efficient even in a low throughput setting, and provides a swift and highly flexible alternative to conventional NGS methods for SNV/InDel analysis, methylation classification and detection of copy number alterations.

Acknowledgment:

This study was supported by the Deutsche Forschungsgemeinschaft (DFG) via Comprehensive Research Center (SFB) 1389 Unite Glioblastoma.

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

References:

1. Auton A, Abecasis GR, Altshuler DM, Durbin RM, Abecasis GR, Bentley DR, Chakravarti A, Clark AG, Donnelly P, Eichler EE, Flicek P, Gabriel SB, Gibbs RA, Green ED, Hurles ME, Knoppers BM, Korbel JO, Lander ES, Lee C, Lehrach H, Mardis ER, Marth GT, McVean GA, Nickerson DA, Schmidt JP, Sherry ST, Wang J, Wilson RK, Gibbs RA, Boerwinkle E, Doddapaneni H, Han Y, Korchina V, Kovar C, Lee S, Muzny D, Reid JG, Zhu Y, Wang J, Chang Y, Feng Q, Fang X, Guo X, Jian M, Jiang H, Jin X, Lan T, Li G, Li J, Li Y, Liu S, Liu X, Lu Y, Ma X, Tang M, Wang B, Wang G, Wu H, Wu R, Xu X, Yin Y, Zhang D, Zhang W, Zhao J, Zhao M, Zheng X, Lander ES, Altshuler DM, Gabriel SB, Gupta N, Gharani N, Toji LH, Gerry NP, Resch AM, Flicek P, Barker J, Clarke L, Gil L, Hunt SE, Kelman G, Kulesha E, Leinonen R, McLaren WM, Radhakrishnan R, Roa A, Smirnov D, Smith RE, Streeter I, Thormann A, Toneva I, Vaughan B, Zheng-Bradley X, Bentley DR, Grocock R, Humphray S, James T, Kingsbury Z, Lehrach H, Sudbrak R, Albrecht MW, Amstislavskiy VS, Borodina TA, Lienhard M, Mertes F, Sultan M, Timmermann B, Yaspo M-L, Mardis ER, Wilson RK, Fulton L, Fulton R, Sherry ST, Ananiev V, Belaia Z, Beloslyudtsev D, Bouk N, Chen C, Church D, Cohen R, Cook C, Garner J, Hefferon T, Kimelman M, Liu C, Lopez J, Meric P, O'Sullivan C, Ostapchuk Y, Phan L, Ponomarov S, Schneider V, Shekhtman E, Sirotkin K, Slotta D, Zhang H, McVean GA, Durbin RM, Balasubramaniam S, Burton J, Danecek P, Keane TM, Kolb-Kokocinski A, McCarthy S, Stalker J, Quail M, Schmidt JP, Davies CJ, Gollub J, Webster T, Wong B, Zhan Y, Auton A, Campbell CL, Kong Y, Marcketta A, Gibbs RA, Yu F, Antunes L, Bainbridge M, Muzny D, Sabo A, Huang Z, Wang J, Coin LJM, Fang L, Guo X, Jin X, Li G, Li Q, Li Y, Li Z, Lin H, Liu B, Luo R, Shao H, Xie Y, Ye C, Yu C, Zhang F, Zheng H, Zhu H, Alkan C, Dal E, Kahveci F, Marth GT, Garrison EP, Kural D, Lee W-P, Fung Leong W, Stromberg M, Ward AN, Wu J, Zhang M, Daly MJ, DePristo MA, Handsaker RE, Altshuler DM, Banks E, Bhatia G, del Angel G, Gabriel SB, Genovese G, Gupta N, Li H, Kashin S, Lander ES, McCarroll SA, Nemesh JC, Poplin RE, Yoon SC, Lihm J, Makarov V, Clark AG, Gottipati S, Keinan A, Rodriguez-Flores JL, Korbel JO, Rausch T, Fritz MH, Stütz AM, Flicek P, Beal K, Clarke L, Datta A, Herrero J, McLaren WM, Ritchie GRS, Smith RE, Zerbino D, Zheng-Bradley X, Sabeti PC, Shlyakhter I, Schaffner SF, Vitti J, Cooper DN, Ball E v, Stenson PD, Bentley DR, Barnes B, Bauer M, Keira Cheetham R, Cox A, Eberle M, Humphray S, Kahn S, Murray L, Peden J, Shaw R, Kenny EE, Batzer MA, Konkel MK, Walker JA, MacArthur DG, Lek M, Sudbrak R, Amstislavskiy VS, Herwig R, Mardis ER, Ding L, Koboldt DC, Larson D, Ye K, Gravel S, Consortium T 1000 GP, authors C, committee S, group P, Medicine BC of, BGI-Shenzhen, Harvard BI of MIT and, Research CI for M, European Molecular Biology Laboratory EBI, Illumina, Genetics MPI for M, University MGI at W, Health USNI of, Oxford U of, Institute WTS, group A, Affymetrix, Medicine AEC of, University B, College B, Laboratory CSH, University C, Laboratory EMB, University H, Database HGM, Sinai IS of M at M, University LS, Hospital MG, University M, National Eye Institute NIH (2015) A global reference for human genetic variation. Nature 526:68–74. doi: 10.1038/nature15393

2. Bady P, Sciuscio D, Diserens A-C, Bloch J, van den Bent MJ, Marosi C, Dietrich P-Y, Weller M, Mariani L, Heppner FL, Mcdonald DR, Lacombe D, Stupp R, Delorenzi M, Hegi ME (2012) MGMT methylation analysis of glioblastoma on the Infinium methylation BeadChip identifies two distinct CpG regions associated with gene silencing and outcome, yielding a prediction model for comparisons across datasets, tumor grades, and CIMP-status. Acta neuropathologica 124:547–560. doi: 10.1007/s00401-012-1016-2

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

3. Bowden R, Davies RW, Heger A, Pagnamenta AT, de Cesare M, Oikkonen LE, Parkes D, Freeman C, Dhalla F, Patel SY, Popitsch N, Ip CLC, Roberts HE, Salatino S, Lockstone H, Lunter G, Taylor JC, Buck D, Simpson MA, Donnelly P (2019) Sequencing of human genomes with nanopore technology. Nature Communications 10:1869. doi: 10.1038/s41467-019-09637-5

4. Capper D, Jones DTW, Sill M, Hovestadt V, Schrimpf D, Sturm D, Koelsche C, Sahm F, Chavez L, Reuss DE, Kratz A, Wefers AK, Huang K, Pajtler KW, Schweizer L, Stichel D, Olar A, Engel NW, Lindenberg K, Harter PN, Braczynski AK, Plate KH, Dohmen H, Garvalov BK, Coras R, Hölsken A, Hewer E, Bewerunge-Hudler M, Schick M, Fischer R, Beschorner R, Schittenhelm J, Staszewski O, Wani K, Varlet P, Pages M, Temming P, Lohmann D, Selt F, Witt H, Milde T, Witt O, Aronica E, Giangaspero F, Rushing E, Scheurlen W, Geisenberger C, Rodriguez FJ, Becker A, Preusser M, Haberler C, Bjerkvig R, Cryan J, Farrell M, Deckert M, Hench J, Frank S, Serrano J, Kannan K, Tsirigos A, Brück W, Hofer S, Brehmer S, Seiz-Rosenhagen M, Hänggi D, Hans V, Rozsnoki S, Hansford JR, Kohlhof P, Kristensen BW, Lechner M, Lopes B, Mawrin C, Ketter R, Kulozik A, Khatib Z, Heppner F, Koch A, Jouvet A, Keohane C, Mühleisen H, Mueller W, Pohl U, Prinz M, Benner A, Zapatka M, Gottardo NG, Driever PH, Kramm CM, Müller HL, Rutkowski S, von Hoff K, Frühwald MC, Gnekow A, Fleischhack G, Tippelt S, Calaminus G, Monoranu C-M, Perry A, Jones C, Jacques TS, Radlwimmer B, Gessi M, Pietsch T, Schramm J, Schackert G, Westphal M, Reifenberger G, Wesseling P, Weller M, Collins VP, Blümcke I, Bendszus M, Debus J, Huang A, Jabado N, Northcott PA, Paulus W, Gajjar A, Robinson GW, Taylor MD, Jaunmuktane Z, Ryzhova M, Platten M, Unterberg A, Wick W, Karajannis MA, Mittelbronn M, Acker T, Hartmann C, Aldape K, Schüller U, Buslei R, Lichter P, Kool M, Herold-Mende C, Ellison DW, Hasselblatt M, Snuderl M, Brandner S, Korshunov A, von Deimling A, Pfister SM (2018) DNA methylation-based classification of central nervous system tumours. Nature 555:469–474. doi: 10.1038/nature26000

5. Euskirchen P, Bielle F, Labreche K, Kloosterman WP, Rosenberg S, Daniau M, Schmitt C, Masliah-Planchon J, Bourdeaut F, Dehais C, Marie Y, Delattre J-Y, Idbaih A (2017) Same-day genomic and epigenomic diagnosis of brain tumors using real-time nanopore sequencing. Acta neuropathologica 134:691–703. doi: 10.1007/s00401-017-1743-5

6. Jain M, Olsen HE, Paten B, Akeson M (2016) The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. Genome Biology 17:239. doi: 10.1186/s13059-016-1103-0

7. Loose M, Malla S, Stout M (2016) Real-time selective sequencing using nanopore technology. Nature methods 13:751–754. doi: 10.1038/nmeth.3930

8. Oxford Nanopore Technologies Ltd. (2021) Oxford Nanopore Technologies GitHub - Megalodon. In: Github. https://github.com/nanoporetech/megalodon

9. Payne A, Holmes N, Clarke T, Munro R, Debebe BJ, Loose M (2021) Readfish enables targeted nanopore sequencing of gigabase-sized genomes. Nature Biotechnology 39:442–450. doi: 10.1038/s41587-020-00746-x

10. Rushing EJ (2021) WHO classification of tumors of the nervous system: preview of the upcoming 5th edition. memo - Magazine of European Medical Oncology. doi: 10.1007/s12254-021-00680-x

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

11. Sahm F, Schrimpf D, Jones DTW, Meyer J, Kratz A, Reuss D, Capper D, Koelsche C, Korshunov A, Wiestler B, Buchhalter I, Milde T, Selt F, Sturm D, Kool M, Hummel M, Bewerunge-Hudler M, Mawrin C, Schüller U, Jungk C, Wick A, Witt O, Platten M, Herold-Mende C, Unterberg A, Pfister SM, Wick W, von Deimling A (2016) Next-generation sequencing in routine brain tumor diagnostics enables an integrated diagnosis and identifies actionable targets. Acta Neuropathologica 131:903–910. doi: 10.1007/s00401-015-1519-8

12. Suvakov M, Panda A, Diesh C, Holmes I, Abyzov A (2021) CNVpytor: a tool for CNV/CNA detection and analysis from read depth and allele imbalance in whole genome sequencing. bioRxiv 2021.01.27.428472. doi: 10.1101/2021.01.27.428472

13. Tate JG, Bamford S, Jubb HC, Sondka Z, Beare DM, Bindal N, Boutselakis H, Cole CG, Creatore C, Dawson E, Fish P, Harsha B, Hathaway C, Jupe SC, Kok CY, Noble K, Ponting L, Ramshaw CC, Rye CE, Speedy HE, Stefancsik R, Thompson SL, Wang S, Ward S, Campbell PJ, Forbes SA (2019) COSMIC: the Catalogue Of Somatic Mutations In Cancer. Nucleic acids research 47:D941–D947. doi: 10.1093/nar/gky1015

14. Wang K, Li M, Hakonarson H (2010) ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic acids research 38:e164. doi: 10.1093/nar/gkq603

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

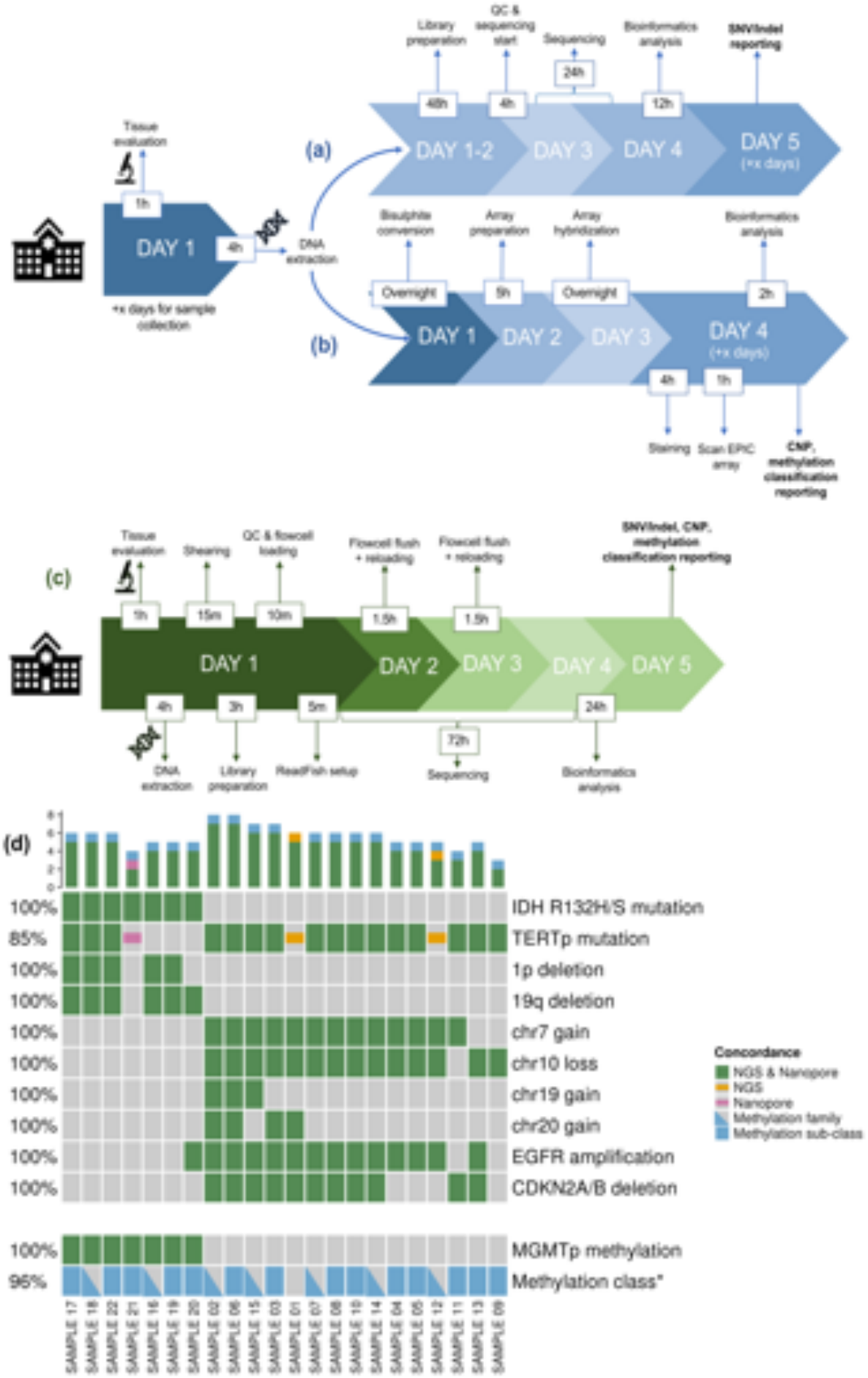


Figure 1: RAPID-CNS[2] timeline and concordance. Timeline for (a) NGS panel sequencing and analysis pipeline, and (b) EPIC array analysis pipeline for neuropathology diagnostics (x denotes number of days required to pool sufficient samples). (c) Timeline for RAPID-CNS[2] sequencing and analysis pipeline for a single sample. (d) Concordance of clinically relevant alterations & classification. Coloured blocks indicate presence of alteration, concordance for detected alterations is denoted in the legend. Triangular denotations for methylation class

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

indicate samples where methylation families were concordant and blocks indicate concordance for sub-classes as well. Percentages on the left indicate concordance for the alteration over all samples.

Supplementary methods

Nanopore library prep optimized for adaptive sampling

Sections of 40x10 µm were prepared from cryoconserved tumor tissues with established molecular markers (IRB approval 2018-614N-MA, 005/2003) with tumor cell content (based on a H&E stain) > 60%. DNA was then extracted using the Promega Maxwell RSC Blood DNA Kit (catalogue # AS1400, Promega) on a Maxwell RSC 48 instrument (AS8500, Promega) per manufacturer's instructions. DNA concentrations were measured on a microplate reader (FLUOStar Omega, BMG Labtech) using the Invitrogen Qubit DNA BR Assay Kit (Q32851, Thermo Fisher Scientific). Next, the DNA was sheared to approximately 9 to 11 kb in a total volume of 50 µl using g-TUBEs (Covaris) at 7200 rpm for 120 sec. The fragment length was assessed on an Agilent 2100 Bioanalyzer (catalogue # G2939A, Agilent Technologies) with the Agilent DNA 12000 Kit (catalogue # 5067-1508, Agilent Technologies). Sequencing libraries were prepared with the SQK-LSK109 Ligation Sequencing Kit with the following modifications: 48 µl of the sheared DNA (2-2.5 µg) were taken into the end-prep reaction, leaving out the control DNA. The end-prep reaction was changed to an incubation for 30min at 20°C followed by 30min at 65°C followed by a cool down to 4°C in a thermal cycler. The clean-up was performed using AMPure XP beads and 80% ethanol, elution time was changed to 5min. Adapter ligation was extended to an incubation for 60min at room temperature. The ligation mix was then incubated with AMPure XP beads at 0.4x for 10min, clean-up was performed using the Long Fragment Buffer (LFB) and the final library was eluted in a total volume of 31 µl. Library concentrations were measured using the Invitrogen Qubit DNA HS Assay Kit (Q32851, Thermo Fisher Scientific) on a benchtop Quantus fluorometer (Promega). The libraries were loaded (500-600 ng) onto FLO-MIN106 R9.4.1 flow cells with a minimum of 1100 pores available according to the FC Check prior to loading. The flow cells were flushed after around 24 hours for a total of two times per sample with the Flow Cell Wash Kit (EXP-WSH003) per manufacturer's instructions. All sequencing was carried out on a MinION 1B (Oxford Nanopore Technologies).

ReadFish

Targeted nanopore sequencing was performed in real-time using a custom panel with ReadFish on an 8 GB NVIDIA RTX 2080 Ti powered consumer notebook [9]. The targets included regions from the neuropathology panel and CpG sites instrumental in classification by the random forest methylation classifier (available on GitHub). A 25kbp flank was added to the sites to ensure optimal targeting by ReadFish. Guppy 4.2.2's fast basecalling (config dna_r9.4.1_450bps_fast) mode was used to run ReadFish.

Bioinformatics analysis

FAST5 files were basecalled using Guppy 4.4.1's high accuracy configuration. QC and coverage analyses were performed by pycoQC and deepTools respectively. Adapter trimming by Porechop was followed by minimap2 v2.18 alignment to the hg19 genome, samtools sorting and indexing. SNVs were called using longshot v0.4.1 and PEPPER-Margin-DeepVariant r0.4 . TERT promoter mutations were detected by mpileup and bcftools. Variant

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

annotation was performed by ANNOVAR. Filtering for clinical relevance was based on the 1000 Genomes (Aug 2015) frequencies and COSMIC 68 database. Copy number plots (100kb bin size) and gene-level copy number files (1kb, 10kb and 100kb bin sizes) were generated using CNVpytor and a custom script. Megalodon v2.3.1 was used to obtain methylation values.

Methylation classification

To classify nanopore sequencing derived DNA-methylation profiles of central nervous system tumors, a random forest classifier was trained on publicly available 450k methylation array reference data set of the MNP classifier version 11 (GSE90496). This data set was preprocessed as described in [4].

For a batch of 22 nanopore sequencing samples, intersection of CpG probes measured for all samples were selected to train the classifier. The methylation array data set was reduced to these 3,285 probes.

Often nanopore sequencing measures CpG probes with low coverage, which leads to discrete distributed methylation values, i.e. (0, 0.5, 1) for coverage 3. As finer methylation differences can often not be detected with nanopore sequencing for all CpG probes, we trained the RF classifier on dichotomized methylation values. This followed the assumption that splitting rules learned on binary data are more robust and can be applied to methylation signals from nanopore sequencing data.

After dichotomizing the reduced reference methylation data set, a RF was trained with 1000 trees and the resulting permutation based variable importance measure was applied to select the 1000 CpGs with highest variable importance to train a final RF with again 1000 trees. The out-of-the-bag accuracy of this classifier was 96%.

Supplementary results

RAPID-CNS^2 analysis pipeline

The bioinformatics pipeline requires raw FAST5 files as input. Complete instructions for setting up the analysis are available on GitHub. Post set-up, RAPID-CNS^2 runs the entire analysis with a single command. It can be run on an LSF cluster or a GPU workstation. Basecalling followed by SNV and CNV detection completes within 10 hours while methylation calling and classification requires an additional 12 hours.

SNV detection

ANNOVAR annotated tables for all Nanopore sequenced samples and their corresponding panel sequencing results are attached.

CNV detection

Copy number plots obtained using the RAPID-CNS^2 pipeline demonstrate higher resolution and clear visualization of the copy number levels as compared to NGS panel sequencing (Supplementary figure 1a (left and centre)). Calculating depth of mapped reads, copy number variations detected are comparable to EPIC array results

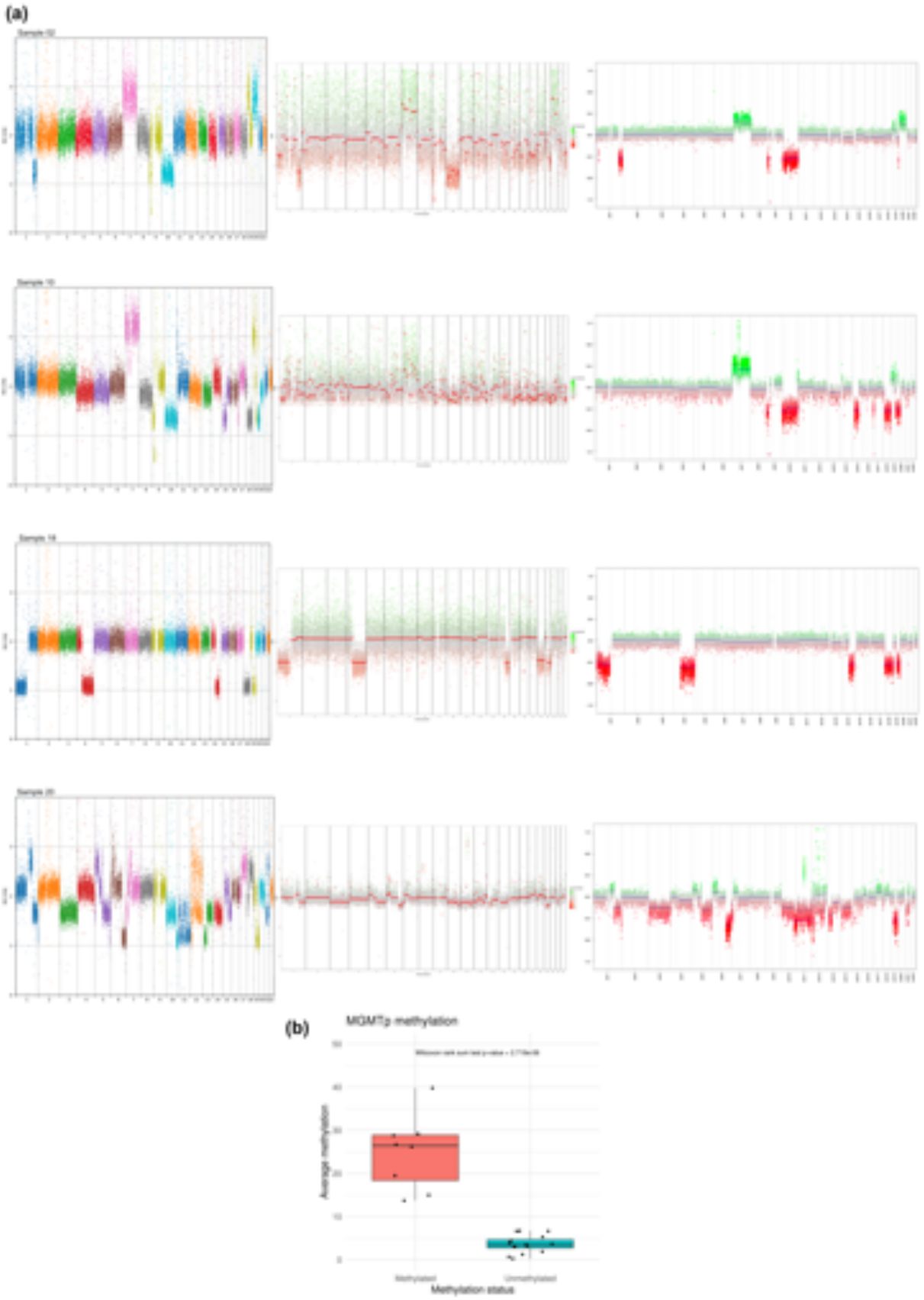
medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .

(Supplementary figure 1a (left and right)). Normalised read depths are indicated on the Y-axis with "2" indicating mean autosomal level. Additionally, genes covered by the copy number variations and their zygosity are annotated and output as excel files (Supplementary files).

MGMT promoter methylation

Two probes used by the MGMT-STP27 approach were not reliably covered in all analysed samples [2]. Methylation frequencies over all CpG sites covering the *MGMT* promoter region were therefore averaged as an alternative measure. Using pyrosequencing as gold standard, methylated and unmethylated samples were found to have a significant difference in their average methylation (Wilcoxon rank sum test p-value= 2.719e-06). As shown in Suppl. Figure 1b, a threshold of 10% was assigned for *MGMT* promoter methylation status.

medRxiv preprint doi: https://doi.org/10.1101/2021.08.09.21261784; this version posted August 10, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity.
It is made available under a CC-BY-NC-ND 4.0 International license .



Supplementary Figure 1: (a) CNV plots obtained using RAPID-CNS[2] (left), panel-sequencing (centre) and EPIC array analysis (right). (b) MGMT promoter methylation values averaged over the MGMT promoter region.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Barcode aware adaptive sampling for Oxford Nanopore sequencers.

Alexander Payne[$], Rory Munro[$], Nadine Holmes, Christopher Moore, Matt Carlile, Matthew Loose*
[$]These authors contributed equally to this paper.
*Author for Correspondence: matt.loose@nottingham.ac.uk

Abstract

Adaptive sampling enables selection of individual DNA molecules from sequencing libraries, a unique property of nanopore sequencing. Here we develop our adaptive sampling tool readfish to become “barcode-aware” enabling selection of different targets within barcoded samples or filtering out individual barcodes. We show that multiple human genomes can be assessed for copy number and structural variation on a single sequencing flow cell using sample specific customised target panels.

Main Text

Adaptive sampling is the process by which individual DNA molecules within a library can be dynamically selected for sequencing, a property unique to Oxford Nanopore Technologies (ONT) sequencers [1]. Recently we developed readfish, which uses real-time base calling to analyse read data as molecules are being sequenced [2]. Using readfish, it is possible to enrich target regions of human genomes as well as manipulate sequencing coverage of metagenomic samples [2–4]. Here we show this method can be extended by enabling the use of barcoded samples with readfish. This allows for individual barcodes to be switched off during a run or enables the use of targets specific to each sample and barcode.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

An advantage of sequence based approaches to adaptive sampling is that existing tools, such as barcode demultiplexers, can be easily incorporated into the readfish workflow. Although signal based methods to identify barcodes exist, no sufficiently fast methods are currently available [5]. We therefore adapted our existing readfish pipeline to be compatible with built-in Guppy demultiplexing (ONT) and incorporated barcode classifications into the data readfish uses to make a decision about sequencing or rejecting a read.

An important consideration in adaptive sampling is what duration of signal data is needed for an accurate mapping of a read fragment. Previously we used chunks of 0.4 seconds of data [2], (roughly 1,600 samples) but reasoned the inclusion of additional barcode sequence at the start of each read would require additional data. To test this we took a set of reads (see methods) and sampled signal from the start of each representing data seen when running adaptive sampling. We then used a variety of base caller models (see methods) and two signal alignment tools, Uncalled and Sigmap, to analyse mappings from each of these synthetic reads and methods [3,6]. As readfish uses the start coordinate of a mapping to determine if a read is on target, we compared the predicted mapping coordinate with that from the full length read (high accuracy mode - HAC). A correct mapping is defined as one where the start mapping coordinates are within 100 bases of one another. We found that 3,600 samples (or 0.8 seconds of data) was sufficient to correctly place reads (F1>0.9, fast model) (Figure 1A). Similarly, this same window also enabled appropriate barcode mapping accuracy (F1>0.9, fast model) (Figure 1B). Therefore we configured all our experiments to use data in chunks of 0.8 seconds (3,200 samples of data).

Readfish can be configured to handle barcodes in two ways. For simple experiments, the user can identify a list of barcodes to be either rejected or accepted. In this way users can exclude or include a subset of barcodes on a sequencing run (Figure 2A). For more complex experiments, the user can configure a set of targets for each individual barcode in a library and so sequence specific regions from each. For example, a cancer gene panel for sample

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

A, a developmental disorders panel for sample B and a neuropathology panel for sample C. Figure 2B illustrates a simple barcoded sample where different regions of a bacterial genome are selected on each barcode in real-time. As readfish maps using sequences it is only limited by available memory and easily handles gigabase genomes. In addition there is no requirement for each sample to be from the same organism and so readfish can target multiple references. To simplify creation and dynamic update of readfish configuration files, we provide a set of command line tools to configure options for multiple barcodes (https://github.com/looselab/readfish-tools).

To test the performance of this approach, we used three previously described cell lines: GM12878, from the Utah/CEPH pedigree; NB4, a cell line carrying a fusion between PML and RARA representing an acute promyelocytic leukemia (APL); and 22Rv1, a prostate cancer derived cell line containing significant chromosomal abnormalities [7–9]. For each sample, we chose a specific gene panel. GM12878 was targeted using a panel defined by the gene list in the commercially available TruSight 170 Tumor panel [10]. As the NB4 cell line contains an APL fusion, we selected the TruSight RNA Fusion Panel [11]. For the more complex 22Rv1 prostate cancer line we used the previously described COSMIC panel [2,12]. Samples were barcoded and sequenced on a single flow cell, and run for 72 hours (see methods). Every 24 hours the flow cell was washed with nuclease flush and another aliquot of the library loaded [2]. In a single experiment using a flow cell with 1,330 pores, 18.1 Gb of data were generated, with a total of 15 Gb successfully demultiplexed into barcoded data (Table 1).

Across the whole experiment, the on target read N50 was 7 kb, with the rejected read N50 being 579 bases, or approximately 1.3 seconds of sequencing. This results in mean read coverage on target regions of between 11x and 15x. Inspection of individual targets including BRCA1, NBR1, PML and RARA demonstrates the ability to specifically target unique regions on each sample (Figure 3). Current best practice for variant calling requires higher minimal

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

depth than we achieve when looking at three samples. However, long range structural variants can be measured and so we used cuteSV [13] to analyse these three samples. As expected, multiple reads supporting the detection of a fusion between PML and RARA were detected in the NB4 cell line (Figure 4). In contrast, this rearrangement was not found in the 22Rv1 line. We cannot formally exclude the presence of this variant in GM12878 as neither PML or RARA were within the gene panel used for this cell line (Figures 3,4).

Finally, we turned to a natural application for adaptive sampling which considers the mappings of rejected reads. Various approaches have been developed using binning of short reads to detect copy number variation by applying a variety of statistical approaches [14]. These methods also work with nanopore sequencing [15], but the resolution of detection will be dependent on the total number of reads generated during a sequencing run. Adaptive sampling increases read count as a consequence of rejecting molecules once they are confidently mapped to an off-target region. We therefore developed a simple approach to bin read counts across the genome such that, on average, each bin would contain 100 reads, and monitored this in real-time using our minoTour tool [16]. For each barcoded sample changes in copy number are immediately apparent and can be visualised using any change point detection approach, here we use Ruptures (Figure 5) [17]. As expected, GM12878 (barcode 1) does not show significant copy number changes, whereas NB4 (barcode 2) and 22Rv1 (barcode 3) both closely recapitulate results generated by Bionano optical mapping (Figure 6).

Discussion

Extending readfish to become “barcode aware” enables more sophisticated selection experiments that are better able to exploit adaptive sampling in a variety of contexts. Here we demonstrate that individual samples can be targeted with unique panels of genes, selected based on knowledge of the sample, enabling the user to ask and answer specific

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

questions. On a single MinION flow cell, 3 human genomes can be analysed in real-time with coverage sufficient to detect structural and copy number variation. In this case, yield limitations prevent a realistic assessment of SNPs. However, we anticipate higher yield or running two human samples per flow cell would enable this. Of course, smaller genomes will generate proportionally higher coverage enabling more samples to be run on a single flow cell as well as providing greater depth for variant calling. Similarly, as flow cell yield increases, and these features become available on platforms such as the PromethION, it will become possible to target multiple human samples on single flow cells.

Alongside these targeted experiments, this approach also allows users to simply switch off barcodes for which sufficient data have been generated. This will enable dynamic adjustment of yields obtained from individual samples in barcoded libraries. Our initial testing shows these approaches will work with the full 96 barcodes currently available on nanopore platforms. Coupling multiple samples with barcode aware readfish and real-time analysis of the data obtained will enable faster experimental turn around times, more efficient use of flow cell resources and more comprehensive analysis pipelines.

Methods

Synthetic read generation and analysis.

To demonstrate our choice of parameters for read mapping and barcode calling, we obtained reads mapping to either chromosome 15 or 17 from the sequenced subset of reads ending up in the pass folder from NB4 (barcode02). Using the ONT Fast5 API (https://github.com/nanoporetech/ont_fast5_api), we generated varying sizes of chunks of signal from the start of these reads incrementing in 0.1 second equivalents (400 samples) to 1 second, then 0.25 seconds to a total of 10,000 samples per read (2.5 seconds). These reads were base called using Guppy (v5.0.16+b9fcd7b) and mapped using minimap2 to the

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

target regions of chromosomes 15 and 17 defined by the trusight RNA fusion panel [11]. Mapping used minimap2 [18] with the -x map-ont and --paf-no-hit option to retain all reads regardless of mapping. We chose the high accuracy model as our truth set as this is the current standard base caller. By using the subset of reads from chromosome 15 and 17 targets only, and hence a smaller reference, we could also test signal based methods for mapping reads including uncalled (v2.2) and sigmap (v0.1) [3,6].

For determining alignment accuracy we considered read starts mapping within 50 bases of the truth set as true positives, although for many applications this may be overly stringent. At this stringency, the fast base calling model recovered true mappings with an F1 score of 0.903524 (precision = 0.927901, recall = 0.880395). The code is available in the accompanying data notebooks. As a result we selected 0.8 seconds of data for analysis. Neither sigmap nor uncalled were optimised beyond the default settings and performance could likely be improved further.

For barcoding of data, we used Guppy demultiplexing and tested no other approach. Truth sets were defined using the full length reads as above. We compared the impact of the base caller model on barcode detection and found the fast model recovers the correct barcode with an F1 > 0.9 at 1,600 samples.

Running readfish Barcoding

Running read until and adaptive sampling requires the ONT Read Until API (version 3.0.0, https://github.com/nanoporetech/read_until_api/tree/release-3.0) and the ONT PyGuppy Client library (version 5.0.13, https://pypi.org/project/ont-pyguppy-client-lib/5.0.13/). Readfish (https://github.com/LooseLab/readfish; commit 9e8794a) was run using a GridION MK1 (MinKNOW v4.3.2; Guppy v5.0.13; minimap2 v2.22), the MinKNOW configuration scripts were configured to serve data in 0.8 second chunks.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

The readfish script carrying out the selective sequencing was “readfish barcode-targets”. This script runs the core Read Until process as specified in the experiment’s TOML file. With a single reference genome the script can select specific target regions on each barcode by using Guppy to base call and demultiplex the raw signal in real-time. The resultant read is then aligned to the reference using minimap2 and is determined to be on or off target depending on it's barcode assignment and mapping start.

Library Preparation, Sequencing and Analysis

Barcoded LSK-110 (ONT) sequencing libraries were prepared from either GM12878 cells (Coriell), NB4 cells (gift from M. Hubank) or 22Rv1 cells (ATCC) as described in Jain et al. [7]. For test experiments bacterial DNA was extracted using genomic tip (QIAGEN). Extracted DNA was sheared to approximately 12 kb using g-Tube (Covaris). All sequencing used FLO-MIN106 R9.4.1 flow cells. Flow cells were run with flushing and reloading as previously described in Payne et al. [2].

To investigate structural variation across the dataset, we ran CuteSV on each barcoded sample using standard options but varying the -s MIN SUPPORT values. No SVs in known fusion genes were reported in NA12878 or 22Rv1 (-s 2), known fusions including PML RARA were readily detected in NB4 (-s 5)[13]. SVs were visualised using Ribbon [19].

To visualise changes in copy number, reads were mapped to hg38, filtered to mapping scores >20 and uniquely mapping. Then the first primary mapping for any read was determined and mappings binned into windows along the genome such that on average each bin contains 100 reads. Runs were monitored in real-time using minoTour (https://github.com/LooseLab/minotourapp/; commit: 1f9c678), providing coverage statistics, mappings and estimates of copy number variation in real-time [16]. During real-time analysis reads were mapped to Chm13 telomere-to-telomere assembly [20,21]. Post-run copy number plots were generated using matplotlib with data mapped to hg38 to compare with the output of the Bionano copy number pipeline (see notebooks).

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

To visualise coverage over specific targets reads were divided into those actively sequenced and those unblocked using the unblocked read ids file generated by readfish. Reads were mapped to hg38, coverage depth calculated using mosdepth v0.3.1[22] and visualised using matplotlib (v3.4.3).

Bionano Methods

DNA extraction and labelling for Bionano

DNA was prepared from frozen cell pellets of 1.5 million cells using the Bionano Prep SP Blood and Cell Culture DNA Isolation Kit (Bionano Genomics; 80042) according to the manufacturer's instructions. DNA was homogenised and quantified using Qubit dsDNA BR Kit (Thermo Fisher; Q32853) on a Qubit 4 Fluorometer (Thermo Fisher; Q33238). 750 ng of gDNA was then labelled with Direct Label Enzyme 1 (DLE-1) and DNA backbone stain using the Bionano Prep Direct Label and Stain (DLS) kit (Bionano Genomics; 80005) according to the manufacturer’s instructions. Labelled DNA was quantified using the Qubit dsDNA HS Kit (Thermo Fisher; Q32851) on a Qubit 4 Fluorometer. Labelled DNA was loaded onto a Bionano Saphyr G2.3 chip (Bionano Genomics; 20366) and run on a Gen 2 Bionano Saphyr System (Bionano Genomics; 60325) until 1.320 Tbp of data had been collected for each of NB4 and 22Rv1. This data had respective mapping rates to hg38 reference sequence of 89% and 79%, equating to 382x and 337x coverage respectively.

Data analysis

Post run data filtering and analysis was carried out using Bionano Access 1.5.2. For each sample the data set was filtered and sub-sampled to produce 320 Gbp of data with 150 kb minimum length and at least 9 labels per molecule. Filtered data was processed to produce annotated *de novo* assemblies using the default parameters, but with masking using the hg38 DLE-1 SV Mask BED file. Structural variant (SV) and copy number variants (CNV) coordinates were then visualised using Bionano Access. All described analysis was

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

performed on dedicated Bionano compute with the following versions installed: Bionano Access1.5.2, Bionano Tools 1.5.3, Bionano Solve Solve3.5.1_01142020, RefAligner 10330.10436rel, HybridScaffold 12162019, SVMerge 12162019 , VariantAnnotation 12162019, Compute on Demand 1.5.1.

Acknowledgements

The authors thank Mike Hubank and Nigel Mongan for gifts of cells and useful discussions. This work was supported by BBSRC iCASE studentship awards to RM and AP. In addition we acknowledge funding from the BBSRC (BB/N017099/1) and Wellcome Trust (grant number 204843/Z/16/Z).

Competing Interest Statement

ML was a member of the MinION access program and has received free flow cells and sequencing reagents in the past. ML has received reimbursement for travel, accommodation and conference fees to speak at events organized by Oxford Nanopore Technologies.

References

1. Loose, M., Malla, S. & Stout, M. Real-time selective sequencing using nanopore technology. *Nat. Methods* **13**, 751–754 (2016).
2. Payne, A. *et al.* Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nat. Biotechnol.* 1–9 (2020).
3. Kovaka, S., Fan, Y., Ni, B., Timp, W. & Schatz, M. C. Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED. doi:10.1101/2020.02.03.931923.
4. Miller, D. E. *et al.* Targeted long-read sequencing identifies missing disease-causing variation. *Am. J. Hum. Genet.* **108**, 1436–1449 (2021).
5. Wick, R. R., Judd, L. M. & Holt, K. E. Deepbinner: Demultiplexing barcoded Oxford

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Nanopore reads with deep convolutional neural networks. *PLoS Comput. Biol.* **14**, e1006583 (2018).

6. Zhang, H. *et al.* Real-time mapping of nanopore raw signals. *Bioinformatics* **37**, i477–i483 (2021).
7. Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338–345 (2018).
8. Mozziconacci, M.-J. *et al.* Molecular cytogenetics of the acute promyelocytic leukemia-derived cell line NB4 and of four all-trans retinoic acid--resistant subclones. *Genes Chromosomes Cancer* **35**, 261–270 (2002).
9. Liu, T. *et al.* Establishment and characterization of multi-drug resistant, prostate carcinoma-initiating stem-like cells from human prostate cancer cell lines 22RV1. *Mol. Cell. Biochem.* **340**, 265–273 (2010).
10. Na, K. *et al.* Targeted next-generation sequencing panel (TruSight Tumor 170) in diffuse glioma: a single institutional experience of 135 cases. *J. Neurooncol.* **142**, 445–454 (2019).
11. Siegfried, A. *et al.* EWSR1-PATZ1 gene fusion may define a new glioneuronal tumor entity. *Brain Pathol.* **29**, 53–62 (2019).
12. Tate, J. G. *et al.* COSMIC: the Catalogue Of Somatic Mutations In Cancer. *Nucleic Acids Res.* **47**, D941–D947 (2019).
13. Jiang, T. *et al.* Long-read-based human genomic structural variation detection with cuteSV. *Genome Biol.* **21**, 189 (2020).
14. Zhang, L., Bai, W., Yuan, N. & Du, Z. Correction: Comprehensively benchmarking applications for detecting copy number variation. *PLoS Comput. Biol.* **15**, e1007367 (2019).
15. Magi, A. *et al.* Nano-GLADIATOR: real-time detection of copy number alterations from nanopore sequencing data. *Bioinformatics* **35**, 4213–4221 (2019).
16. Munro, R. *et al.* MinoTour, real-time monitoring and analysis for Nanopore Sequencers. *bioRxiv* 2021.09.10.459783 (2021) doi:10.1101/2021.09.10.459783.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

17. Truong, C., Oudre, L. & Vayatis, N. Selective review of offline change point detection methods. *Signal Processing* **167**, 107299 (2020).

18. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).

19. Nattestad, M., Aboukhalil, R., Chin, C.-S. & Schatz, M. C. Ribbon: intuitive visualization for complex genomic variation. *Bioinformatics* **37**, 413–415 (2021).

20. Miga, K. H. *et al.* Telomere-to-telomere assembly of a complete human X chromosome. *Nature* **585**, 79–84 (2020).

21. Nurk, S. *et al.* The complete sequence of a human genome. *bioRxiv* 2021.05.26.445798 (2021) doi:10.1101/2021.05.26.445798.

22. Pedersen, B. S. & Quinlan, A. R. Mosdepth: quick coverage calculation for genomes and exomes. *Bioinformatics* **34**, 867–868 (2018).

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Figures and Tables

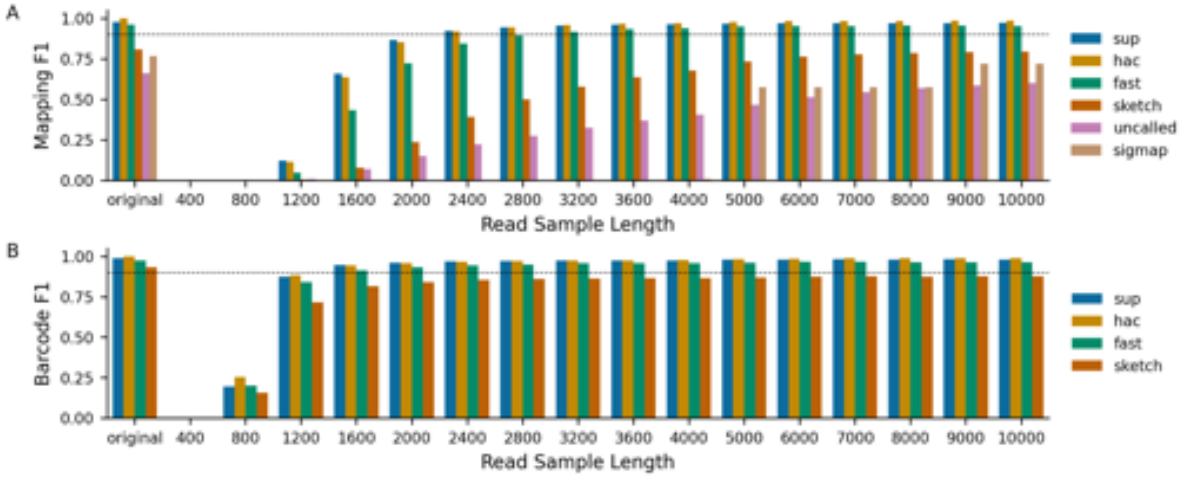
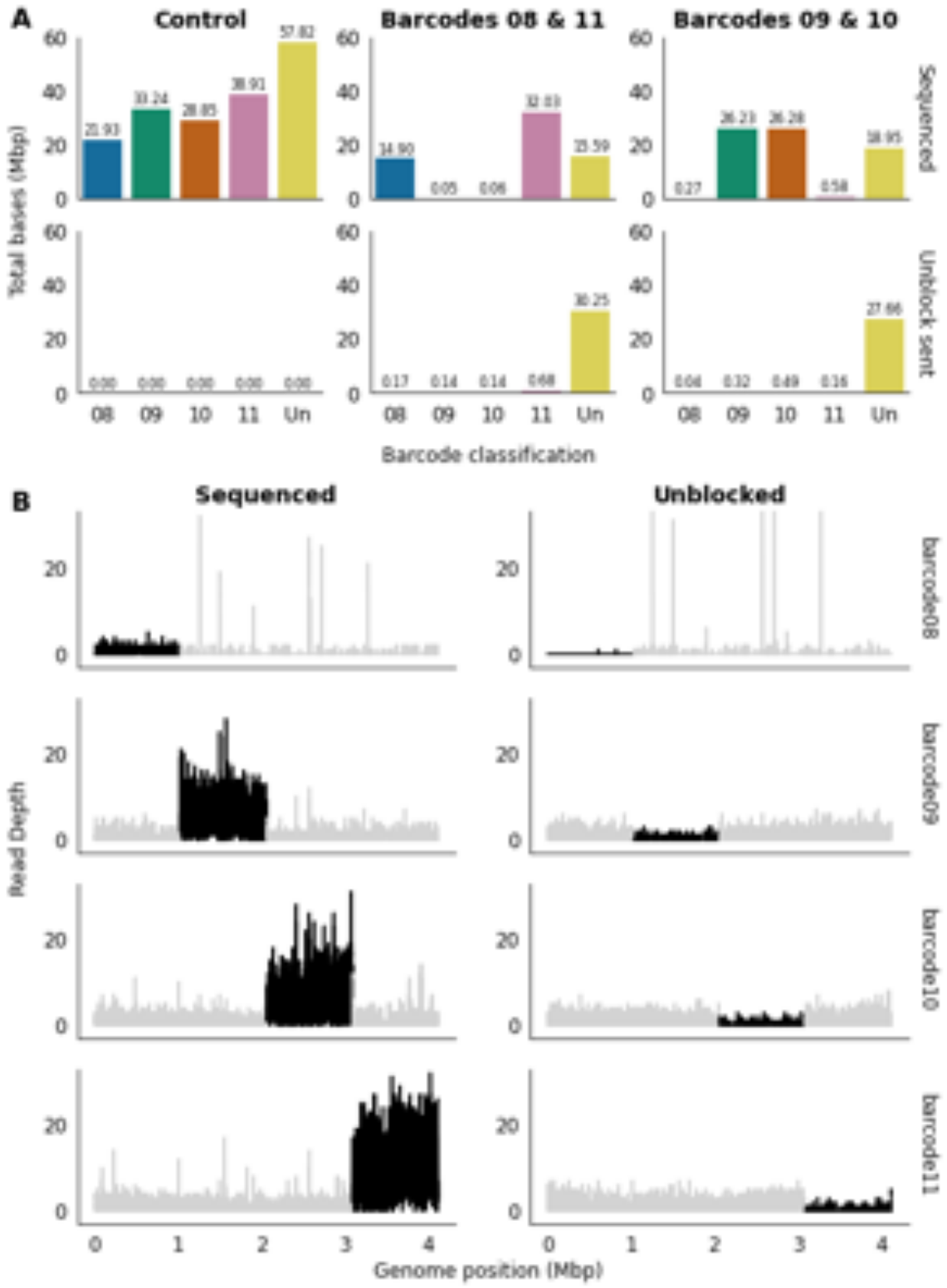


Figure 1

Comparison of base callers, alignments and barcode classifications. A set of pass reads derived from chromosome 15 and 17 targets from the truSight Fusion panel were generated. A) Reads were base called using the super accuracy (sup), high accuracy (hac), fast (fast) or sketch (sketch) models of guppy and mapped to a synthetic genome containing only the target regions for those read targets. These same reads were also mapped using the signal aligners Uncalled and Sigmap. Truth was defined as the start mapping coordinate for the full length read (original). Read fragments were scored as mapping correctly if the start mapping coordinates were within 50 bases of the true start mapping position. 0.9 F1 is exceeded at 0.8 seconds of data (3200 samples) for the fast model. B) F1 score as measured by concordance in barcode identified where truth is the HAC model.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

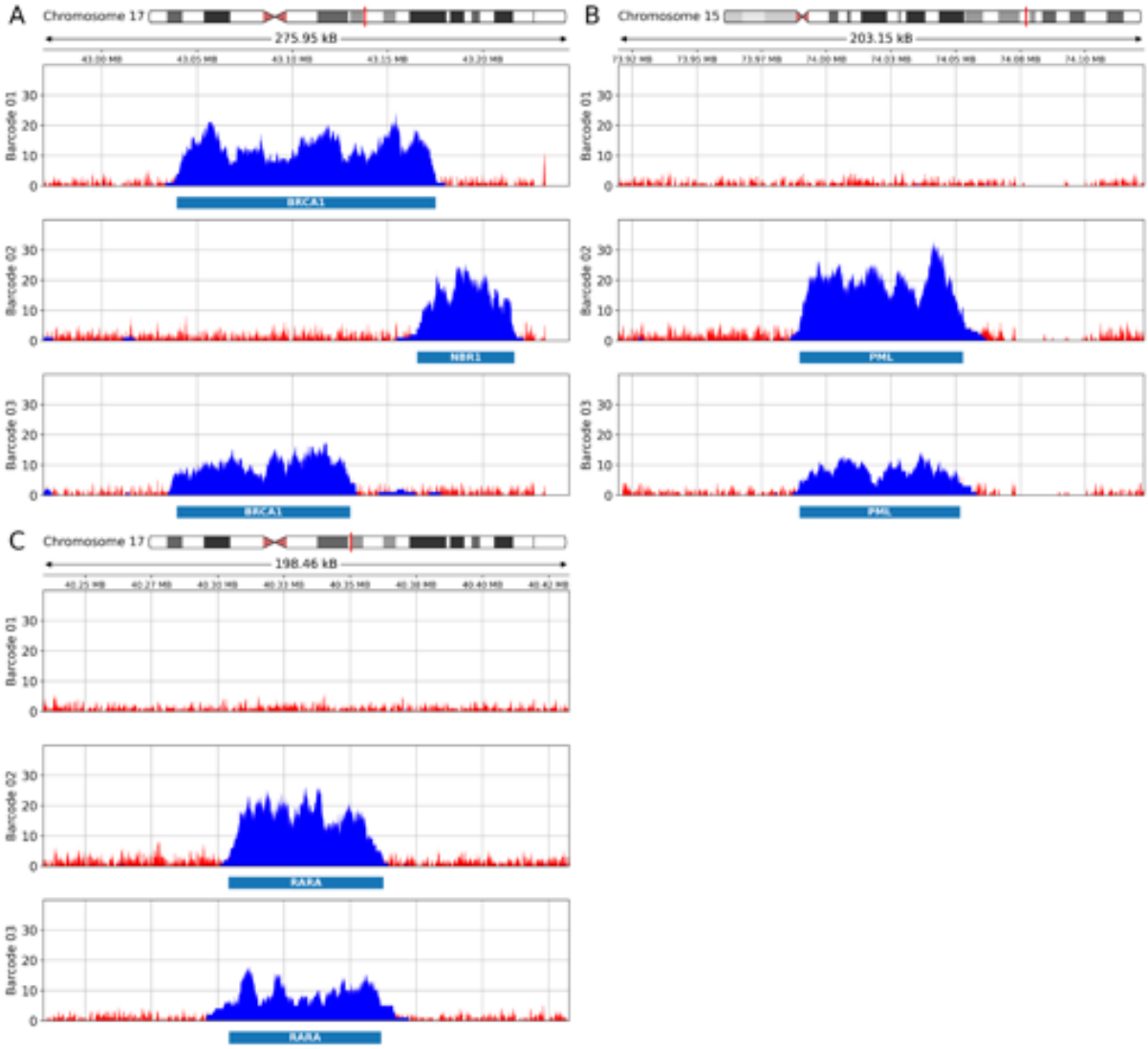
Figure 2



Naive barcoding selective sequencing A) Demonstration of “switching off” individual barcodes from a sequencing library. Selected barcodes identified in the panel titles. Top row shows sequenced reads, lower panel shows the rejected or unblocked reads. As barcoding both ends is used to specify barcode, all rejected reads become unclassified (Un) by default. B) Switching the mode of operation for readfish from simple barcode rejection to differential targets. Sample shown is *Clostridioides difficile*. Targeted regions are shown in black.

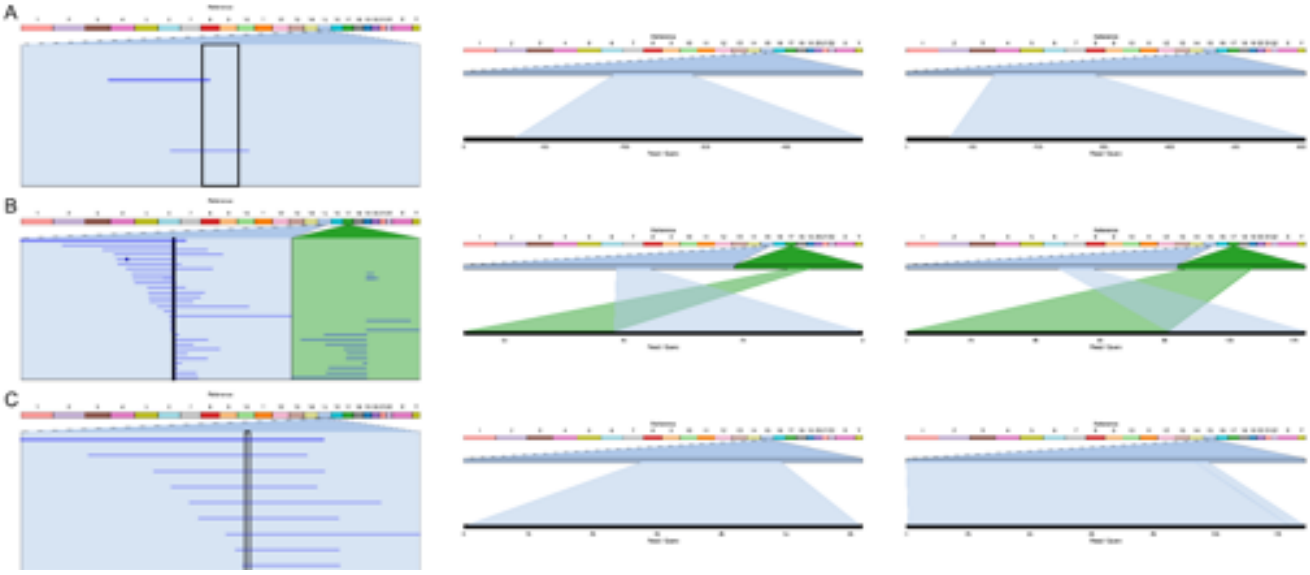
bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Figure 3



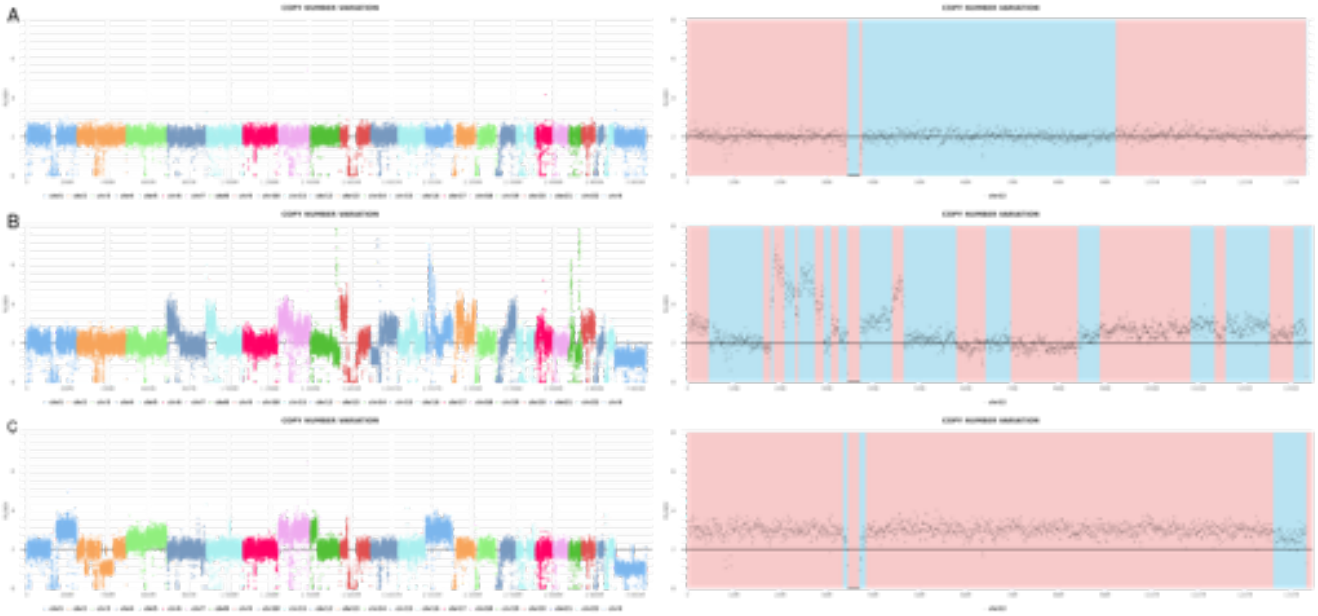
Target and barcode specific gene coverage. Illustration of coverage over each barcoded sample for each target in the panel. Blue is sequenced read coverage, red illustrates coverage of rejected reads. A) shows coverage over BRCA1 and the adjacent gene NBR1. BRCA1 was a target for barcode 1 and 3, but not 2. The targeted regions are illustrated below the coverage plots. Note that the region representing BRCA1 differs in barcode 1 and 3 by design. NBR1 was only targeted on barcode 2. B) and C) illustrate coverage over PML and RARA respectively, which were only targeted on barcodes 2 and 3.

Figure 4



Visualising Structural Variation. Using Ribbon, we visualise reads covering PML (chromosome 15) and any known fusions. A) Barcode 01, GM12878, has only two reads in the candidate region as PML is not included within the targets for this sample. B) Barcode 02, NB4, shows multiple reads spanning PML and linking to RARA (chromosome 17) as expected for this fusion cell line. C) Barcode 03, 22Rv1, also had PML within the target gene list, but had no structural variant in this region as expected. SVs were identified using CuteSV (supplementary file 1).

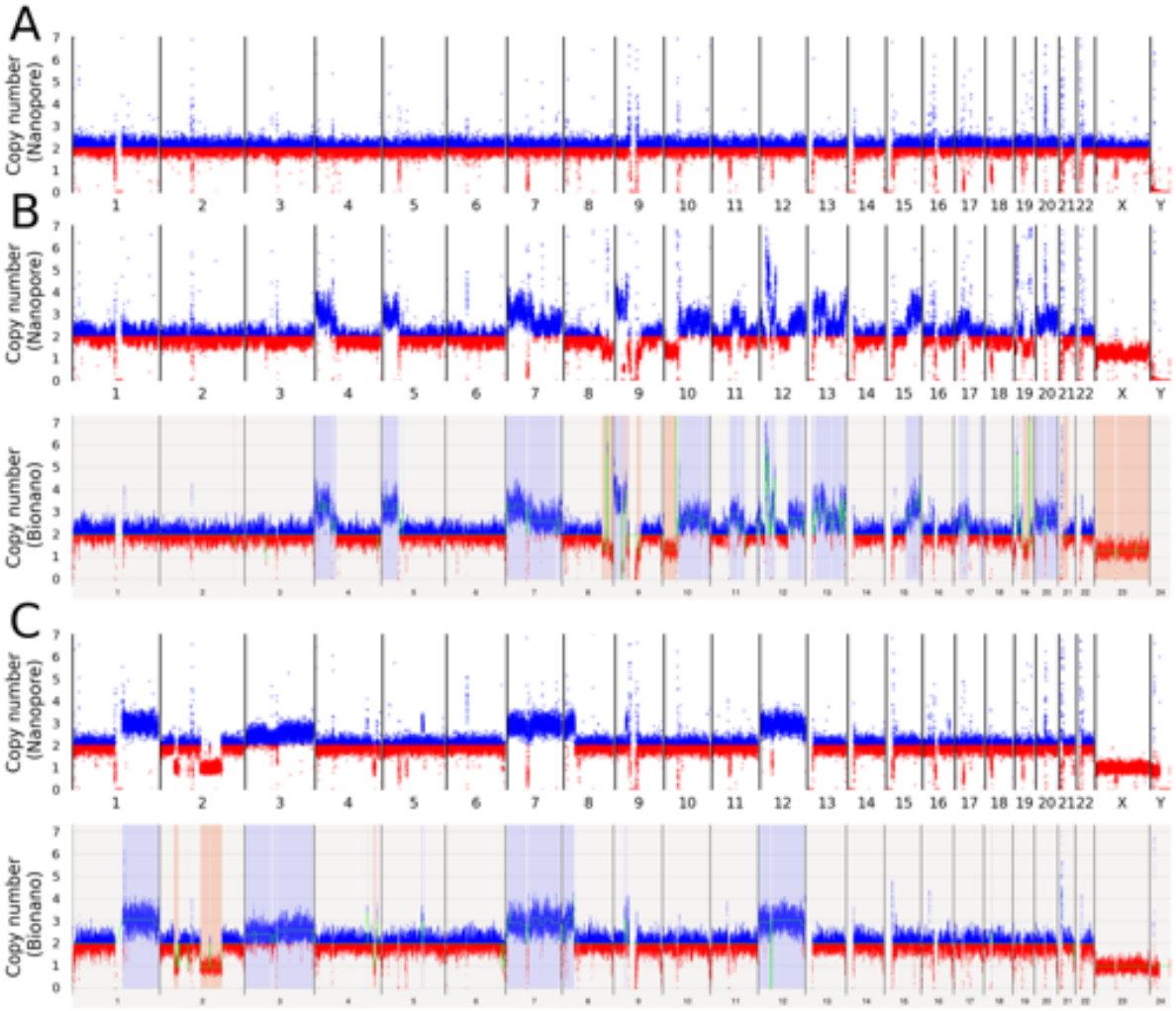
Figure 5



Real-time monitoring of copy number change. MinoTour generates real-time counts of reads dynamically binned such that each bin contains on average 100 reads. Samples shown here mapped to Chm13 T2T reference. Left hand plots show coverage over all chromosomes, right hand plots show just chromosome 12. Red Blue banding indicates change points as dynamically detected by Ruptures. A) barcode 01, GM12878, bin width 86,600 bases. B) barcode 02, NB4, bin width 60,570 bases C) barcode 03, 22Rv1, bin width 76,470 bases.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Figure 6



Matched Nanopore Bionano CNV visualization. Nanopore read data and Bionano optical reads both mapped against hg38. Blue points show where binned data indicates greater than expected copy number, red points where binned data indicates lower than expected copy number. A) NA12878 showing Nanopore adaptive sampling data only from barcode 01. B) NB4 and C) 22Rv1 showing Nanopore adaptive sampling data and Bionano optical mapping data.

bioRxiv preprint doi: https://doi.org/10.1101/2021.12.01.470722; this version posted December 1, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Table 1

Barcode	Sample	Panel	Gene Number	Yield (Gb)	On Target (Gb)	On Target N50	On Target Mean	Off Target Mean	Mean Target Coverage
01	GM12878	TruSight 170 Tumor Panel	170	3.8	0.355	8,149	1,926	554	11.0
02	NB4	TruSight RNA Fusion Panel	508	6.1	1.240	7,191	4,203	551	15.0
03	22Rv1	COSMIC	717	5.1	1.250	6,858	5,065	556	11.5
Unclassified				3.1				736	
Total				**18.79**			**3,221**	**587**	

Sample Performance. Run metric performance per barcode and over the entire flow cell. Metrics are derived from real-time monitoring with minoTour.