BOSS-RUNS: a flexible and practical dynamic read sampling framework for nanopore sequencing

Nicola De Maio[1], Charlotte Manser[1¤], Rory Munro[2], Ewan Birney[1], Matthew Loose[2] and Nick Goldman[1*]

1 European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Cambridge CB10 1SD, UK
2 DeepSeq, School of Life Sciences, Queen's Medical Centre, University of Nottingham, Nottingham NG7 2UH, UK

¤Current Address: Murray Edwards College, Cambridge CB3 0DF, UK
*goldman@ebi.ac.uk

Abstract

Real-time selective sequencing of individual DNA fragments, or 'Read Until', allows the focusing of Oxford Nanopore Technology sequencing on pre-selected genomic regions. This can lead to large improvements in DNA sequencing performance in many scenarios where only part of the DNA content of a sample is of interest. This approach is based on the idea of deciding whether to sequence a fragment completely after having sequenced only a small initial part of it. If, based on this small part, the fragment is not deemed of (sufficient) interest it is rejected and sequencing is continued on a new fragment. To date, only simple decision strategies based on location within a genome have been proposed to determine what fragments are of interest. We present a new mathematical model and algorithm for the real-time assessment of the value of prospective fragments. Our decision framework is based not only on which genomic regions are *a priori* interesting, but also on which fragments have so far been sequenced, and so on the current information available regarding the genome being sequenced. As such, our strategy can adapt dynamically during each run, focusing sequencing efforts in areas of highest uncertainty (typically areas currently low coverage). We show that our approach

can lead to considerable savings of time and materials, providing high-confidence genome reconstruction sooner than a standard sequencing run, and resulting in more homogeneous coverage across the genome, even when entire genomes are of interest.

Author Summary

An existing technique called 'Read Until' allows selective sequencing of DNA fragments with an Oxford Nanopore Technology (ONT) sequencer. With Read Until it is possible to enrich coverage of areas of interest within a sequenced genome. We propose a new use of this technique: combining a mathematical model of read utility and an algorithm to select an optimal dynamic decision strategy (i.e. one that can be updated in real time, and so react to the data generated so far in an experiment), we show that it possible to improve the efficiency of a sequencing run by focusing effort on areas of highest uncertainty.

Introduction

Nanopore sequencing (commercially available from Oxford Nanopore Technologies, ONT) enables fast, portable and cheap long-read sequencing [1, 2]. It has a number of fundamental differences to the sequencing-by-synthesis approaches due to its entirely different sensing approach of detecting the sequence of DNA or RNA (and other small analytes) as they translocate through a small pore (nanopore). By maintaining a voltage difference across the nanopore and detecting changes in ionic current the nature of the analyte can be determined. The progression of nucleic acid through the pore is controlled by a motor protein ensuring enough readings can be taken to permit deconvolution of the contributions of all nucleic acids to the current signal. This mechanism does not change along the nucleic acid polymer, meaning that read length is determined by sample preparation and the ability to deliver the sample to the pore, and can be extremely large [1, 3]. As the change in current is specific to the precise chemistry of the bases in the pore, the sequence of the nucleic acid polymer can be determined and both DNA and RNA can be sequenced [4].

One of the most promising aspects of ONT sequencing is the fact that it provides

sequencing data in real time, allowing the device or operator to make decisions about the DNA fragments currently being sequenced by the nanopores. In the former case, the sequencer can remove analytes that may block pores. In the latter, the operator can choose to reject fragments deemed uninteresting while retaining those deemed valuable. This technology, called 'Read Until' [5] has been used, for example, for selective sequencing enrichment of pre-determined areas of a genome, normalising coverage of amplicons or removing off target sequences [5–8]. These developments have sparked interest in technological, mathematical and algorithmic methods for optimizing the decision framework for which fragments should be prioritized in Read Until [6, 9, 10].

Here, we propose new techniques that expand the potential and applicability of ONT and Read Until by

- increasing the confidence in the reconstructed genome (reducing the number of genome inference errors),
- compensating for coverage fluctuations along the genome due to amplification and sequencing biases or random chance, and deliver a more uniform coverage while increasing the minimum coverage achieved, and
- focusing sequencing efforts on regions that are *a posteriori*, but not *a priori*, more important, for example identifying regions with indels and rearrangements that could cause subsequent assembly difficulties or be more biologically interesting.

To achieve this, we introduce a mathematical and algorithmic framework for quantifying the expected value of DNA fragment reads of which only a small initial portion has already been sequenced. We propose a decision strategy that rejects reads that are not deemed sufficiently valuable, while accounting for the expected value of future reads and the costs of the decision-making process, rejection of low-value fragments and acquisition of new ones. We prove that our strategy is optimal in terms of capturing the most value at any given moment in the sequencing experiment and, finally, we illustrate the use and advantages of our strategy in a number of realistic scenarios.

Materials and Methods

There are two components to this work. The first is the definition of an objective function to quantify the value of a read, while considering possible regions of interest in the genome and considering sample preparation as well as all the sequencing information obtained so far from the sequencing run.

The second component is a dynamic (updatable) decision strategy that determines in real time which fragments are worth sequencing and which should be rejected, based on the initial portion of the fragment that has already been sequenced.

Objective Function

We propose a probabilistic framework to develop appropriate objective functions. The basic idea is to consider the information gain that we expect a new read will provide, i.e. its "expected benefit", given that we know the location and orientation of the considered fragment along a reference genome. We assume that we know a reference genome of N positions, over which reads (or partial reads) can be mapped. We do not consider cases in which a reference genome is not available — possibly these could be dealt with in future with real-time *de novo* assembly. We further assume that reads can be unambiguously mapped onto the reference, which in turn ignores complications deriving from large-scale mutational events such as rearrangements or copy number variations; these types of events could also be included in future versions of the methods (see Supplement).

The expected benefit of a candidate read is determined by the expected changes in posterior genotype probability distributions (measured by the Kullback-Leibler divergence [11]), over positions that could be covered by the considered DNA fragment if it were further sequenced. The current posterior probability of a genotype at each genome location is obtained by combining prior information about the position (e.g. reference genome, and possibly prior population data) with information from the reads sequenced so far, as discussed below. Sequencing error rates are also taken into account, and the prospective posterior genotype probabilities after sequencing the candidate fragment are calculated while additionally considering the expected fragment length distribution.

Ultimately, a DNA fragment that is expected to give a greater reduction in the uncertainty regarding the genotype being sequenced will be considered more useful than a fragment with a limited potential to alter posterior probabilities. For example, if a genome position has already been covered by many reads, and these reads support one genotype with high confidence, then the expected benefit of further interrogation of this position will usually be small. In contrast, if a genome position has been covered by very few reads, or the previous reads leave high uncertainty regarding the sequenced genotype, then sequencing further reads covering the position will have high expected benefit.

Positional Score

Here we discuss prior and posterior probabilities of different genotypes at a position of a genome and define the score for the position, which will be used later to define the expected score of a new read. We make a few simplifying assumptions to ease presentation, and discuss extensions in the Supplement. Our first simplification is that we assume that genetic diversity and sequencing errors occur only in the form of substitutions (SNPs), while in the Supplement we discuss ways to account for indels and rearrangements. We further assume that all positions in all reads are subject to sequencing errors with the same probability, independently of the genotype or particular read considered, and that sequencing errors across read positions are independent of one another.

We denote the set of possible genotypes for the considered individual at the considered genome position by G. For example, for a haploid genome, $g \in G$ is just one of the four bases $b \in B = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$; that is, $G = B$. For an unphased diploid genome, $g \in G$ is one of the unordered pairs $g = \{b_1, b_2\}$, with $b_1, b_2 \in B$. For a phased diploid genome (which we do not consider further), $g \in G$ is an ordered pair of alleles $(b_1, b_2) \in B \times B$. Similar definitions are possible also for polyploid genomes. In some circumstances, ploidy might not be known *a priori*; in such case, even more complex definitions of G would be needed.

For each position i of a reference genome of length N, we denote $\pi_i(g)$ the location-specific prior on genotypes $g \in G$ before any data have been observed. In all applications below, when considering a haploid genome, we define the prior of reference

nucleotide b_R at position i as $\pi_i(b_R) = 1 - \theta$, with θ the genetic diversity of the considered population. Conversely, $\pi_i(g) = \theta/3$ if $g \neq b_R$.

When considering diploid sequenced genomes, we still assume a haploid reference genome, with reference nucleotide at a given position denoted b_R. In the case of a diploid unphased genome being sequenced, we define $\pi_i(\{b_R, b_R\}) = 1 - \theta$, and $\pi_i(\{g, g\}) = p_{\text{homo}}\theta/3$ if $g \neq b_R$, with p_{homo} being the proportion of site differences from a reference that are expected to be homozygous, and $\pi_i(\{g, b_R\}) = (1 - p_{\text{homo}})\theta/3$ for $g \neq b_R$. We ignore the possibility of a heterozygous genome being sequenced with both alleles different from the reference genome. These prior probability definitions also ignore differences in mutation rates across nucleotides and genome positions and do not use prior knowledge on SNP locations derived from the population; when available, these aspects could however easily be included in the definition of $\pi_i(g)$.

Assume that at a given point in an experiment we have observed data D, containing n reads mapping to position i. We denote by $d_{j,i} \in B$ the nucleotide observed in read j that maps to reference position i. Then, the posterior probability of genotype $g \in G$ at position i and conditional on data D is

$$f_i(g|D) = \frac{\pi_i(g) \prod_{j=1}^{n} \phi(d_{j,i}|g)}{Z_i(D)} \,. \tag{1}$$

$Z_i(D)$ is a normalising constant, representing the likelihood of the data and ensuring that the sum of the posteriors at site i is 1:

$$Z_i(D) = \sum_{c \in G} \left(\pi_i(c) \prod_{j=1}^{n} \phi(d_{j,i}|c) \right) \,. \tag{2}$$

$\phi(d_{j,i}|g)$ is the probability of calling base $d_{j,i}$ assuming genotype g at position i, and will depend on the assumptions being made. For example, for a haploid genome in our applications below we define

$$\phi(d_{j,i}|b) = \begin{cases} 1 - e \,, & \text{if } d_{j,i} = b \in B \,, \\ \dfrac{e}{3} \,, & \text{if } d_{j,i} \neq b \in B \,. \end{cases} \tag{3}$$

where e denotes the per-base sequencing error probability, meaning that any position

along a read has a probability e of mis-representing the corresponding nucleotide of the sequenced genome.

In the scenario of an unphased diploid genome we will instead consider

$$\phi(d_{j,i}|\{b_1, b_2\}) = \begin{cases} 1-e\,, & \text{if } d_{j,i} = b_1 = b_2\,, \\ \dfrac{1-e}{2} + \dfrac{e}{6}\,, & \text{if } d_{j,i} = b_1 \neq b_2 \text{ or } d_{j,i} = b_2 \neq b_1\,, \\ \dfrac{e}{3}\,, & \text{if } d_{j,i} \neq b_1,\ b_2\,. \end{cases} \tag{4}$$

From the posterior probabilities $f_i(g|D)$ of genotypes g at position i, conditional on data D, we can define the 'expected benefit' of one new base covering position i. First, if D contains n reads covering position i with bases $d_{j,i}$ for $j = 1 \ldots n$, we denote the base from a new hypothetical read at position i by $d_{n+1,i}$. We represent D' as the union of D with the new hypothetical read, so that D' contains $n+1$ reads covering position i, with bases $d_{j,i}$ for $j = 1 \ldots n+1$. After observing the new read, the updated posterior probabilities become:

$$\begin{aligned} f_i(g|D') &= \frac{\pi_i(g) \prod_{j=1}^{n+1} \phi(d_{j,i}|g)}{\sum_{c \in G} \left(\pi_i(c) \prod_{j=1}^{n+1} \phi(d_{j,i}|c) \right)} \\ &= \frac{f_i(g|D) Z_i(D) \phi(d_{n+1,i}|g)}{\sum_{c \in G} f_i(c|D) Z_i(D) \phi(d_{n+1,i}|c)} \\ &= \frac{f_i(g|D) \phi(d_{n+1,i}|g)}{\sum_{c \in G} f_i(c|D) \phi(d_{n+1,i}|c)}\,. \end{aligned} \tag{5}$$

The Kullback-Leibler (KL) divergence (or relative entropy [11]) is a measure of how different two distributions are. We want to use, as a measure of expected benefit, the KL divergence between the posterior probability distributions before ($f_i(g|D)$) and after ($f_i(g|D')$) observing a new read, as this tells us how much informative the new read is about the genotype being sequenced. However, we don't know which base $d_{j,i}$ will be the next one observed, so, instead, we average out over the possible values of $d_{j,i}$. $P(d_{n+1,i}|D)$, the probability of observing $d_{n+1,i}$, is given by

$$P(d_{n+1,i}|D) = \sum_{g \in G} f_i(g|D) \phi(d_{n+1,i}|g)\,. \tag{6}$$

S_i, the current expected benefit of a new read at position i, is the expected KL divergence D_{KL} between distributions $f_i(b|D)$ and $f_i(b|D')$:

$$
\begin{aligned}
S_i &= \sum_{d_{n+1,i} \in B} P(d_{n+1,i}|D) D_{\mathrm{KL}}(f_i(g|D') \,||\, f_i(g|D)) \\
&= \sum_{d_{n+1,i} \in B} P(d_{n+1,i}|D) \left(\sum_{g \in G} f_i(g|D') \log \frac{f_i(g|D')}{f_i(g|D)} \right) \\
&= \sum_{d_{n+1,i} \in B} \sum_{g \in G} P(d_{n+1,i}|D) f_i(g|D') \log f_i(g|D') \\
&\qquad - \sum_{g \in G} \log f_i(g|D) \left(\sum_{d_{n+1,i} \in B} P(d_{n+1,i}|D) f_i(g|D') \right) \\
&= \sum_{d_{n+1,i} \in B} \sum_{g \in G} P(d_{n+1,i}|D) f_i(g|D') \log f_i(g|D') \\
&\qquad - \sum_{g \in G} \log f_i(g|D) \left(\sum_{d_{n+1,i} \in B} P(d_{n+1,i}|g, D) f_i(g|D) \right) \\
&= \sum_{d_{n+1,i} \in B} \sum_{g \in G} P(d_{n+1,i}|D) f_i(g|D') \log f_i(g|D') \\
&\qquad - \sum_{g \in G} f_i(g|D) \log f_i(g|D) \left(\sum_{d_{n+1,i} \in B} P(d_{n+1,i}|g, D) \right) \\
&= \sum_{d_{n+1,i} \in B} \sum_{g \in G} P(d_{n+1,i}|D) f_i(g|D') \log f_i(g|D') - \sum_{g \in G} f_i(g|D) \log f_i(g|D) \,. \qquad (7)
\end{aligned}
$$

Defining the expected benefit in terms of KL divergence as above is a technique used in Bayesian experimental design [12], and is equivalent to defining it in terms of expected reduction in Shannon entropy [13] of the posterior genotype probability distribution after observing one more read base at the position considered.

As the size of D grows, the benefit of sequencing a new base $d_{n+1,i}$ at a position i will usually become smaller and smaller. If a position i is instead covered by few, possibly discordant reads in D, then new information in the form of $d_{n+1,i}$ can shift the posterior probability considerably, leading to much higher expected benefit. The values are, or course, modified by the priors for a given case: data that tend to confirm the prior lead to decreased benefit expected from further reads; data that conflict with the prior require more reads before relative certainty is achieved. Table 1 shows different

values of expected benefit S_i for a number of examples. Table 2 lists the key parameters and variables used in our methods.

Table 1. Example benefit scores

Observed Counts				Posteriors				Score
n_{A}	n_{C}	n_{G}	n_{T}	$f_i(\mathrm{A}\|D)$	$f_i(\mathrm{C}\|D)$	$f_i(\mathrm{G}\|D)$	$f_i(\mathrm{T}\|D)$	S_i
0	0	0	0	0.99	0.0333	0.0333	0.0333	0.0347
1	0	0	0	0.9998	7.2×10^{-5}	7.2×10^{-5}	7.2×10^{-5}	7.6×10^{-4}
3	0	0	0	$1 - 10^{-7}$	3.2×10^{-8}	3.2×10^{-8}	3.2×10^{-8}	3.4×10^{-7}
0	1	0	0	0.8584	0.1358	0.0029	0.0029	0.3296
0	1	2	0	0.1163	0.0184	0.8649	3.9×10^{-4}	0.3364
0	1	2	5	1.3×10^{-6}	2.0×10^{-7}	9.6×10^{-6}	$1 - 1.1 \times 10^{-5}$	3.9×10^{-5}

Some examples of scores S_i and posteriors $f_i(g|D)$ for given counts $(n_{\mathrm{A}}, n_{\mathrm{C}}, n_{\mathrm{G}}, n_{\mathrm{T}})$ of observed bases at position i. Here we assume that the reference genome has $b_R = \mathrm{A}$ at this position, that we have no indels, and that $\theta = 0.01$ and $e = 0.06$. For the first line, i.e. in the absence of read data, posteriors and priors are identical: $f_i(g|D) = \pi_i(g)$.

Read Utility

Now that we have defined a score S_i for each individual genome position i, we need to combine the scores of multiple positions into a scoring system for reads, assuming that each read maps to a series of contiguous bases in the reference genome. For simplicity, we describe our methods in the context of a circular chromosome, as typical for bacteria; in the Supplement we relax this assumption and consider the case of one or more linear chromosomes. We assume the circular genome has length N: fragments that extend beyond position N continue from position 1 – in effect $S_j = S_{j-N}$ for $j > N$; more generally, $S_j = S_{j \bmod N} = S_{j \,\%\, N}$.

First, we define $S^l_{i,1}$ as the sum of l consecutive S_j values starting at position i, that is, the score of a forward-oriented read of length l starting at position i:

$$S^l_{i,1} = \sum_{j=i}^{i+l-1} S_j \,. \tag{8}$$

Similarly, for a reverse-oriented read we have

$$S^l_{i,0} = \sum_{j=i-l+1}^{i} S_j \,. \tag{9}$$

Table 2. Parameters and variables.

Variable	Description
N	Reference genome size
B	Set of observable characters (bases): $B = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$
G	Set of observable genotypes for the sequenced genome
θ	Prior probability that a site has a substitution relative to the reference genome
p_{homo}	Prior proportion of diploid genome sites different from the reference, that are homozygous
b_R	Reference genome nucleotide at a position
e	Probability that a nuclotide is mis-read as a different nucleotide
$L(l)$	Probability that a fragment has length l
η	Number of values used to approximate $\widetilde{CL}(l)$
ρ	Time required to reject a fragment
α	Time required to acquire a new fragment
μ	Length required for mapping a fragment
$\mathcal{S}$	Read Until strategy
$I^{\mathcal{S}}_{i,o}$	Decision function of strategy $\mathcal{S}$ for a fragment starting at i with orientation o
$F_{i,o}$	Probability that a fragment starts at i and has orientation o
Parameter	**Description**
$\pi_i(g)$	Prior probability of genotype g at position i
$f_i(g\|D)$	Posterior probability of genotype g at position i given data D
$\phi(d\|g)$	Probability of a read containing character d for a position with sequence genotype g
$P(d\|D)$	Posterior probability of sequencing character d given data D (depends also on prior)
S_i	Score, or benefit from additional sequencing, of position i
$S^l_{i,o}$	Cumulative score of read starting in position i, orientation o and length l
$U_{i,o}$	Expected score of a read starting in position i and orientation o
$\widetilde{CL}(l)$	Complementary prior cumulative distribution of fragment lengths
$\mathcal{D}_{\widetilde{CL}}$	Domain of $\widetilde{CL}$ (values where the distribution is strictly positive)
λ	Mean fragment length
$\widehat{\mathcal{S}}$	Strategy with optimal score gain rate
$U^{\mathcal{S}}_{i,o}$	Expected benefit of a fragment starting at i with orientation o under strategy $\mathcal{S}$
$t^{\mathcal{S}}_{i,o}$	Expected cost of a fragment starting at i with orientation o under strategy $\mathcal{S}$
$\bar{U}^{\mathcal{S}}$	Expected benefit of strategy $\mathcal{S}$ for next fragment
$\bar{t}^{\mathcal{S}}$	Expected cost of strategy $\mathcal{S}$ for next fragment
$\bar{S}^{\mu}_{o}$	Expected benefit of a read of length μ and orientation o

Description of variables and parameters used in the methods.

If we knew in advance the total length l of the fragment under consideration starting at position i, we could use the above $S^l_{i,1}$ or $S^l_{i,0}$ as a measure the expected benefit of this fragment. However, we usually only only know the length of the part of the fragment that has already been sequenced, and therefore we have to account for the uncertainty in l. To do this, we assume a single distribution of fragment lengths applies to all DNA fragments available for sequencing, irrespective of the genomic location or orientation of the fragment. In the Supplement we discuss the case of linear chromosomes, where this assumption does not hold. We denote the fragment length

distribution by $L(l)$ for lengths $l = 1 \dots N$, with mean $\lambda = \sum_{l=1}^{N} L(l)l$. Since there will be lower and upper limits on the length of fragments in a given experiment, it is convenient to define $\mathcal{D}_L$ to be the domain of L, i.e. the set of values of l with $L(l) > 0$. In many realistic sequencing scenarios, $\min \mathcal{D}_L$ (i.e. the smallest plausible fragment length) will be $\gg 1$; in some scenarios (short genomes/chromosomes) it is possible that $\max \mathcal{D}_L$ (longest plausible fragment) will be $\approx N$; for large genomes/chromosomes, it may be $\ll N$.

Finally, we define the expected benefit $U_{i,1}$ of a read starting at position i, and oriented in forward direction, as

$$U_{i,1} = \sum_{l \in \mathcal{D}_L} L(l) S_{i,1}^{l} \,. \tag{10}$$

This is, equivalently, the sum of the S_j scores for all positions $j \geq i$, each weighted by the probability that the read will reach position j. Considering the cumulative distribution of fragment lengths $CL(l) = \sum_{j=1}^{l} L(j)$ and its corresponding complementary cumulative distribution $\widetilde{CL}(l) = 1 - CL(l) = \sum_{j=l+1}^{N} L(j)$ (Note that $\mathcal{D}_{\widetilde{CL}}$ runs from 1 to $\max \mathcal{D}_L$), Eq. 10 can also be rewritten as

$$U_{i,1} = \sum_{l \in \mathcal{D}_{\widetilde{CL}}} \widetilde{CL}(l) S_{i+l-1} \,. \tag{11}$$

Reverse reads are dealt analogously, with expected score

$$U_{i,0} = \sum_{l \in \mathcal{D}_{\widetilde{CL}}} \widetilde{CL}(l) S_{i+1-l} \,. \tag{12}$$

Calculating $U_{i,1}$ and $U_{i,0}$ for all genome positions with a naive algorithm would require, in many scenarios, quadratic time in genome length, which would be excessive for our purposes. In the next section we describe how to efficiently and accurately approximate their values, with total cost linear in genome size, using an approach based on approximating $\widetilde{CL}$ with a piecewise constant or linear function.

Fast Approximation to Read Utility

Calculating $U_{i,1}$ as shown in eq. 11 requires time proportional to $|\mathcal{D}_{\widetilde{CL}}|$. As $U_{i,1}$ needs to be calculated for each i, the total cost for the whole genome would be in the order of $O(N \times |\mathcal{D}_{\widetilde{CL}}|)$, which is excessively slow in many scenarios. For this reason, we consider approximations to reduce the computational demand of calculating $U_{i,1}$ and $U_{i,0}$. These approximations are based on the idea of substituting $\widetilde{CL}(l)$ with an approximating function.

Here, we present the simpler case of approximating $\widetilde{CL}(l)$ with a piecewise constant function. This is also the approximation that we use in all applications considered here. In the Supplement we also discuss an approximation using a piecewise linear function. Assuming that $\widetilde{CL}(l)$ is a piecewise constant function means that there are values $1 = x_1 < x_2 < \ldots < x_\eta = \max \mathcal{D}_{\widetilde{CL}} + 1$ such that for all $1 \leq \nu < \eta$ and for all $x \in [x_\nu, x_{\nu+1})$ we have $\widetilde{CL}(x) = \widetilde{CL}(x_\nu)$. As before, we have that

$$U_{1,1} = \sum_{l \in \mathcal{D}_{\widetilde{CL}}} \widetilde{CL}(l) S_l \,. \tag{13}$$

This still requires time proportional to $|\mathcal{D}_{\widetilde{CL}}|$; however, calculating $U_{i,1}$ for every other genome position $i > 1$ now requires only time $O(\eta)$ for each i, with η the number of different values taken by $\widetilde{CL}$. In full, if we know $U_{i,1}$ we can calculate $U_{i+1,1}$ as

$$U_{i+1,1} = U_{i,1} - S_i + S_{i+x_\eta-1}\, \widetilde{CL}(x_{\eta-1}) + \sum_{2 \leq \nu < \eta} \left(\widetilde{CL}(x_{\nu-1}) - \widetilde{CL}(x_\nu) \right) S_{i+x_\nu-1} \,. \tag{14}$$

The same approach can be used for efficiently calculating the scores $U_{i,0}$ of reverse reads. In all following applications, we approximate $\widetilde{CL}(l)$ with a piecewise constant function taking $\eta = 11$ different values.

Decision framework for accepting or rejecting fragments

Using the read scores $U_{i,1}$ and $U_{i,0}$ defined in the previous sections, we now define our 'Read Until strategy' for deciding which reads to reject and which reads to fully sequence, and show an efficient algorithm to find this strategy in practice. Our aim is to optimise the rate of accumulation of 'expected benefit' over pores and over time. As

read scores $U_{i,1}$ and $U_{i,0}$ depend on genotype priors and on data D observed so far, our strategy will do so as well. This means in particular that, as the sequencing run progresses and D grows larger, the optimal strategy will also change; our aim in practice will not only be to find such a strategy, but also to update it dynamically during each sequencing run.

We assume that all fragments traverse pores at the same rate, independent of their original genomic location and composition. To simplify exposition, we measure time in units of fragment bases that could be read by a pore. For example, a time t is the time taken by a pore to read through t bases when a fragment is already translocating through that pore. This choice of time unit has the advantage for us of not requiring separate parameterization of the rate at which fragments pass through pores. We assume that, if the decision to reject a fragment is made, the process of rejecting a fragment takes a constant time ρ. We assume that acquiring a new fragment to read, either after a fragment rejection or the completion of the reading of an accepted fragment, takes constant time α.

We also assume that the initial part of a DNA fragment that is sequenced and used to assess the genomic location of a DNA fragment has a constant length $\mu < \min \mathcal{D}_L$. This means that, as a new DNA fragment enters a pore, we assume we always read its first μ bases, and that these μ bases from the fragment are sufficient to map the fragment onto the genome, that is, to infer the genome position i at which the fragment starts and its orientation. The decision of accepting or rejecting the new fragment will then be based on i and on the orientation of the fragment, and not directly on the μ sequenced bases of the considered DNA fragment. See Figs 1 and 2 for a graphical summary of the parameters of our sequencing model.

We define a strategy $\boldsymbol{\mathcal{S}}$ to be a function $I^{\boldsymbol{\mathcal{S}}}_{i,o}$ returning a 0 or 1 value for all $i = 1 \ldots N$ and for $o \in \{0, 1\}$. Boolean variable o represents the orientation of a read (1 for forward, 0 for reverse), and i the position along the reference genome of its first base. Here, $I^{\boldsymbol{\mathcal{S}}}_{i,1} = 0$ indicates that a forward fragment starting at position i should be rejected, while $I^{\boldsymbol{\mathcal{S}}}_{i,0} = 1$ indicates that a reverse fragment starting at position i should be read to its end, and so on. We say that $\boldsymbol{\mathcal{S}}$ *includes* (i, o) if $I^{\boldsymbol{\mathcal{S}}}_{i,o} = 1$, and define $|\boldsymbol{\mathcal{S}}|$, the *size* of $\boldsymbol{\mathcal{S}}$, to be the number of position-orientation pairs (i, o) at which $I^{\boldsymbol{\mathcal{S}}}_{i,o} = 1$. Good strategies $\boldsymbol{\mathcal{S}}$ are not known *a priori*, and our aim here is to determine an optimal (or close to

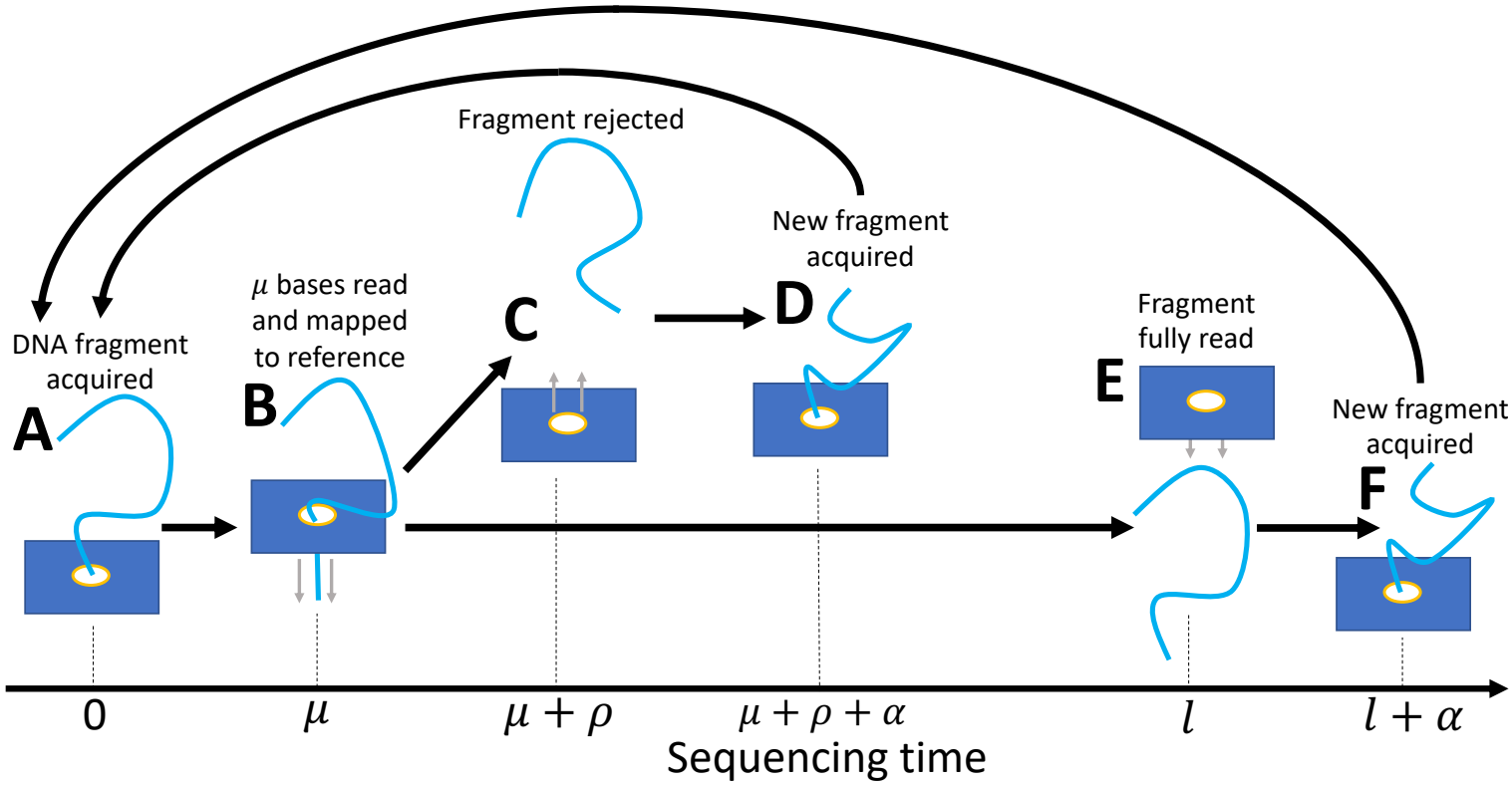

Fig 1. Graphical representation of our Read Until model, and parameters involved. **A** At time 0 a new fragment is acquired by a pore. **B** μ bases of the fragment have been read at time μ, and these bases are used to find the location of the fragment on the genome and to decide whether to read or reject it. **C** If at step **B** rejection was decided then this takes place, completing at time $\mu+\rho$. **D** Following rejection, a new fragment is acquired at time $\mu+\rho+\alpha$, and a new iteration starts with this new fragment. **E** If step **B** was not a rejection, then the fragment is finished reading at time l, where l is the length of the fragment. **F** Following completion of fragment reading, a new fragment is acquired at time $l+\alpha$, and a new iteration starts with this new fragment. To aid visualization, distances along the x-axis are not represented to a realistic scale.

optimal) strategy $\widehat{\mathcal{S}}$ given the current data D. We again assume that we have a circular genome or chromosome and are interested in knowing the whole sequenced genotype. The case of linear chromosome, the case of multiple chromosomes, and the case that one is interested in knowing only part of the genome, are all presented in the Supplement.

$\frac{1}{2}(\bar{S}_1^{\mu}+\bar{S}_0^{\mu})$

Given the definitions above, the expected benefit of a DNA fragment of orientation o starting at position i is

$$U_{i,o}^{\mathcal{S}} = S_{i,o}^{\mu} + I_{i,o}^{\mathcal{S}}\big(U_{i,o} - S_{i,o}^{\mu}\big)\,, \tag{15}$$

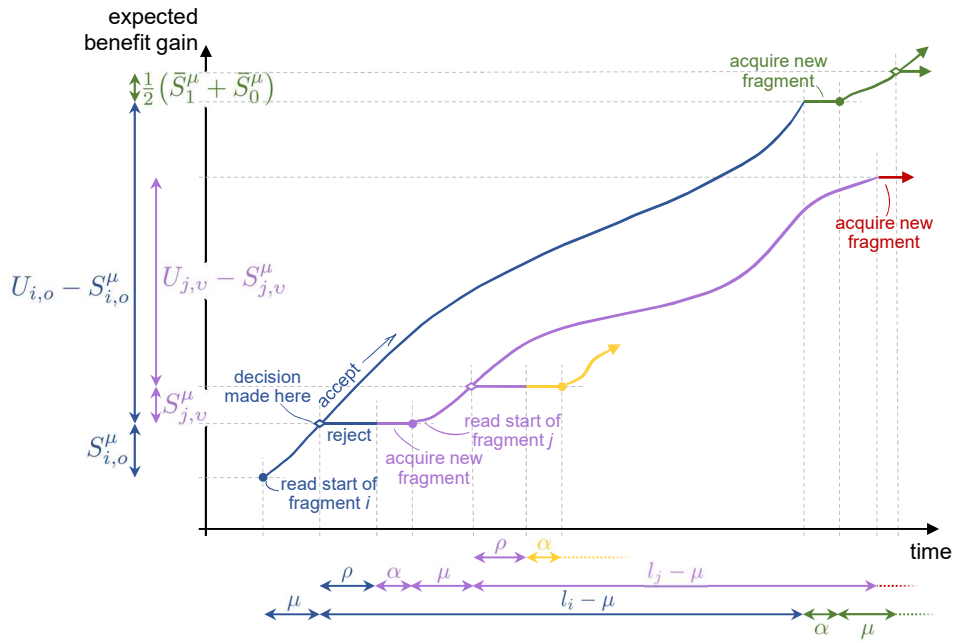

Fig 2. Schematic plot of our model of accumulated benefit against sequencing time. Expected benefit gain is shown on the y-axis. For simplicity, we again use an unrealistic scale for time on the x-axis. Colors are used to show contributions related to different DNA fragments. Starting when the pore has acquired a fragment (blue), after time μ this is mapped, its genomic location and orientation (i, o) determined, benefit $S_{i,o}^\mu$ recorded, and the decision made whether to read the remainder of the fragment. If so, this takes time $l_i - \mu$ (where l_i is the fragment length, with expectation λ) and generates further benefit $U_{i,o} - S_{i,o}^\mu$, after which (green) a new DNA fragment is acquired (taking time α), mapped (after time μ, with benefit having expected value $1/2(\bar{S}_1^\mu + \bar{S}_0^\mu)$), etc. Otherwise, the (blue) fragment is rejected (time ρ, no benefit), a new fragment (mauve) acquired and mapped (time $\alpha + \mu$, location j, orientation v, benefit $S_{j,v}^\mu$) and a decision on whether to continue with this fragment made. Initial effects of other potential fragments are shown in gold and red. Filled circles mark points where new fragments have been acquired, corresponding to labels A, D or F in Fig 1; decision points are marked with open diamonds and correspond to label B in Fig 1.

achieved in time

$$t_{i,o}^{\boldsymbol{S}} = \mu + I_{i,o}^{\boldsymbol{S}}(\lambda - \mu) + \left(1 - I_{i,o}^{\boldsymbol{S}}\right)\rho + \alpha = \alpha + \mu + \rho + I_{i,o}^{\boldsymbol{S}}\left(\lambda - \mu - \rho\right) \ . \qquad (16)$$

Calculating the strategy-wise average time cost $\bar{t}^{\boldsymbol{S}}$ and benefit $\bar{U}^{\boldsymbol{S}}$ requires knowing how often fragments from certain positions i and orientation o are captured by pores. In all our example applications, we assume that both orientations and all starting positions

are equally likely. However, for generality here we use the notation $F_{i,o}$ to refer to the probability that a random fragment's first base maps on genome position i and its orientation is o (1 for forward and 0 for reverse as usual), so that $\sum_{o=1,0}\sum_{i=1}^{N} F_{i,o} = 1$. The average benefit per fragment $\bar{U}^{\mathcal{S}}$ of strategy $\mathcal{S}$ then becomes

$$\begin{aligned}\bar{U}^{\mathcal{S}} &= \sum_{o=1,0}\sum_{i=1}^{N} F_{i,o} U_{i,o}^{\mathcal{S}} \\ &= \sum_{o=1,0}\sum_{i=1}^{N} F_{i,o}\left(S_{i,o}^{\mu} + I_{i,o}^{\mathcal{S}}\big(U_{i,o} - S_{i,o}^{\mu}\big)\right) \end{aligned} \tag{17}$$

and its average fragment-wise cost $\bar{t}^{\mathcal{S}}$ is

$$\begin{aligned}\bar{t}^{\mathcal{S}} &= \sum_{o=1,0}\sum_{i=1}^{N} F_{i,o} t_{i,o}^{\mathcal{S}} \\ &= \alpha + \mu + \rho + (\lambda - \mu - \rho)\sum_{o=1,0}\sum_{i=1}^{N} F_{i,o} I_{i,o}^{\mathcal{S}} \,. \end{aligned} \tag{18}$$

For the special case of uniform distribution of fragments, $F_{i,o} = 1/2N$, eqs. 17 and 18 become

$$\bar{U}^{\mathcal{S}} = \frac{\bar{S}_1^{\mu} + \bar{S}_0^{\mu}}{2} + \frac{1}{2N}\sum_{o=1,0}\sum_{i=1}^{N} I_{i,o}^{\mathcal{S}}\big(U_{i,o} - S_{i,o}^{\mu}\big) \tag{19}$$

and

$$\bar{t}^{\mathcal{S}} = \alpha + \mu + \rho + \frac{|\mathcal{S}|}{2N}(\lambda - \mu - \rho)\,, \tag{20}$$

where

$$\bar{S}_o^{\mu} = \sum_{i=1}^{N} F_{i,o} S_{i,o}^{l}\,. \tag{21}$$

If nanopores are used for short amounts of time since a strategy $\mathcal{S}$ has been calculated, benefit is expected to accumulate at rate $\bar{U}^{\mathcal{S}}/\bar{t}^{\mathcal{S}}$. In practice, we continuously update the chosen strategy as more sequencing data is generated. To find the current strategy $\widehat{\mathcal{S}}$ that maximises the expected benefit per unit time $\bar{U}^{\mathcal{S}}/\bar{t}^{\mathcal{S}}$ given the sequencing data already generated, we need to find

$$\widehat{\mathcal{S}} = \arg\max_{\mathcal{S}} \frac{\bar{U}^{\mathcal{S}}}{\bar{t}^{\mathcal{S}}}\,. \tag{22}$$

We call our optimal strategy approach 'BOSS-RUNS', for Benefit-Optimizing

Short-term Strategy for Read Until Nanopore Sequencing'. We now describe an algorithm that finds $\widehat{\boldsymbol{\mathcal{S}}}$; in the Supplement we include a proof that this algorithm indeed provides the optimal strategy. First, rank all the $2N$ position-orientation pairs (i, o) according to decreasing value of $U_{i,o} - S^{\mu}_{i,o}$ and index them such that (i_1, o_1) takes the highest value, (i_2, o_2) the next and so on: so $U_{i_1,o_1} - S^{\mu}_{i_1,o_1} \geq U_{i_2,o_2} - S^{\mu}_{i_2,o_2} \geq \ldots \geq U_{i_{2N},o_{2N}} - S^{\mu}_{i_{2N},o_{2N}}$. Strategy $\boldsymbol{\mathcal{S}}^{\sigma}$ is defined by setting $I_{\imath}^{\boldsymbol{\mathcal{S}}^{\sigma}} = 1$ for $\imath = (i_1, o_1), \ldots, (i_{\sigma}, o_{\sigma})$ and 0 otherwise, and it is the optimal strategy of size σ. Starting with $\sigma = 0$, we successively increase σ, at each stage testing whether

$$\frac{U_{i_{\sigma+1},o_{\sigma+1}} - S^{\mu}_{i_{\sigma+1},o_{\sigma+1}}}{\lambda - \mu - \rho} > \frac{\bar{U}^{\boldsymbol{\mathcal{S}}^{\sigma}}}{\bar{t}^{\boldsymbol{\mathcal{S}}^{\sigma}}} \tag{23}$$

to discover whether $\boldsymbol{\mathcal{S}}^{\sigma+1}$ gives an improvement over $\boldsymbol{\mathcal{S}}^{\sigma}$. Once we reach a value σ^* such that there is no further improvement, we have the optimal $\widehat{\boldsymbol{\mathcal{S}}} = \boldsymbol{\mathcal{S}}^{\sigma^*}$.

Simulations

To test the proposed BOSS-RUNS strategy and investigate its potential in a range of plausible scenarios, we performed simulations of nanopore sequencing with and without it. At present, with ONT sequencing, DNA translocates through the pore at approximately 450 b/s. In line with typical performance of the devices currently available to us, we simulate a rejection time cost of $\rho = 500$, a fragment acquisition cost $\alpha = 300$, and a mapping fragment length of $\mu = 500$. Given the sequencing speed, ρ equates to approximately 1 s (in reality ρ can be configured by the user from 0.1 s upwards), α to approximately 0.5 s, and the mapping fragment length to approximately 1 s. α is dependent on properties of the library including fragment length and the number of molecules available to sequence on the flowcell surface. In principle, it can be estimated by measuring the proportion of time a pore on a flowcell is sequencing, taking into account the mean read length. For example, with a mean read length of 4.5 kb, 95% occupancy would imply a fragment acquisition time of approximately 0.5 s. Choice of μ is dependent on the efficiency of basecalling and mapping and $\mu = 500$ is consistent with our experiences with real time analysis using GPU basecalling [7]. Genetic diversity between the reference and sequenced genomes is taken as $\theta = 0.01$, with a deletion-to-SNPs ratio of $r = 0.4$. See Supplement for detailed description of indel

parameters. The fragment length distribution was modelled as a Normal distribution centered on $l = 3{,}000$, with standard deviation 6,000, truncated to enforce $l > \mu = 500$. This results in a realistic [14] average fragment length of about $\lambda = 6{,}300$ bp. In practical applications, a decision strategy is not required for fragments shorter than μ.

Throughout the first set of scenarios simulated, we assume a circular bacterial genome of size 4Mb:

- 'normal' scenario: uniform (unbiased) expected coverage and the whole genome is considered of interest.

- 'sequencing bias' scenario: we simulate variation in the proportions and orientations of acquired fragments from different locations across the genome. We modelled realistic 10-fold variation in sequencing bias (realized coverage for naive sequencing) between the regions with highest and lowest sequencing bias [14], with 10 sequencing bias peaks and troughs, by setting $F_{i,1} = 5.5 + 4.5\sin(20\pi i/N)$ and $F_{i,0} = 5.5 + 4.5\sin(20\pi(i-\lambda)/N)$. While the simulated $F_{i,o}$ varied across the genome, for selecting the strategy we still used $F_{i,o} = 1/2N$ to mimic a scenario in which sequencing bias is not known *a priori*.

- 'MLST' scenario: we assume that we are interested in sequencing only a small fraction (0.25%) of the genome, consisting of 10 equally spaced loci each of 1 kb. This scenario resembles the case in which one is interested in a multi-locus sequence typing of a bacterial sample [15].

- 'cgMLST' scenario: we assume that we are interested in sequencing one quarter of the genome consisting of 100 equally spaced loci each of 10 kb. This scenario resembles the case in which one is interested in a core-genome multi-locus sequence typing [16].

In a second set of simulations, we consider a genome made of 16 linear chromosomes respectively of sizes 230, 813, 315, 1532, 577, 270, 1091, 563, 440, 745, 667, 1078, 924, 784, 1091, and 948 kb, similar to a yeast genome [17], for a total length of 12,068 kb:

- 'yeast haploid' scenario: we simulate sequencing of a haploid yeast genome with no sequencing bias.

- 'yeast diploid' scenario: we simulate the sequencing of a diploid yeast genome with probability of homozygous SNPs p_{homo} equal to the expected value from a randomly mixing population of 1,000 individuals at neutrality.

In total, therefore, we simulate six scenarios, four bacterial and two yeast ones.

For each scenario we simulate reads produced by an ONT sequencer which is capable of providing real time base calling (the MinIT, MK1C, GridION or PromethION). From these devices, basecalls from completed fragments are written to disk in user-defined batches for subsequent analysis. The default batch size is 4,000 reads per fastq file. As a consequence, the BOSS-RUNS strategy is not updated on a per-read basis, but per batch instead. There is a practical trade-off between batch size and total fastq file number on disk. Therefore we simulate reads in batches of 4,000 to match current default settings. To test possible improvements in strategy performance by reducing batch size (i.e. increased frequency of updates), we also simulate 500 reads per batch. These considerations only apply to analysis of reads once the molecule has finished translocating through the pore — the Read Until data stream is considered on a per-read basis and is not limited by these batch sizes.

Bacterial sequencing is then simulated up to a total pore time of 200,000,000 (corresponding to the time it would take one pore to read a 200 Mb fragment). For yeast sequencing, we considered times up to 600,000,000. We consider four possible strategies:

- 'naive' strategy: all reads are always accepted, corresponding to a standard sequencing run without Read Until.

- 'full BOSS-RUNS' strategy: our BOSS-RUNS strategy is updated every batch, or after a threshold of time if batches arrive too quickly. This time threshold is 4,000,000 for bacterial genomes with batch size 4,000; 1,000,000 for bacterial genomes with batch size 500; 12,000,000 for yeast genomes with batch size 4,000; or 3,000,000 for yeast genomes with batch size 500. This threshold makes sure that there is sufficient time to compute updates to the strategy.

- 'partial BOSS-RUNS' strategy: we sequence a genome using the initial Read Until strategy (i.e. derived at the start of the experiment), but do not make updates to that strategy for some time. We illustrate these strategies using updates only

during the final half or quarter of the full simulation experiments for bacterial scenarios, and during the final two-thirds or one-third for yeast. Once updating starts, it follows the same methods as with full BOSS-RUNS, using all the data accumulated so far. We indicate these strategies using '⅔-BOSS-RUNS', etc. These options mimic scenarios in which a first part of the sequencing run is used to naturally accumulate coverage according to initial expectations of accumulating benefit (dependent on regions of interest but independent of any sequencing data), and the final part is used to fine-focus the sequencing effort, for example on regions with low coverage or higher remaining uncertainty about the sample genotype.

So, in total, we have two batch sizes and four strategies, giving a total of eight sequencing settings. Combined with the six genomic scenarios above, this gives 48 simulation scenarios; for each of those we ran 30 replicates.

Results

Focusing Sequencing Efforts on Regions of Interest

A naive enrichment of specific regions of interest defined *a priori* by rejecting unwanted reads has been previously demonstrated [5, 7, 8]. We began our simulations with a similar goal, adding our refined BOSS-RUNS strategies to seek improvements in performance over earlier approaches. We first focus on two specific scenarios, resembling a bacterial MLST study (10 regions of interest, each 1 kb long, covering in total 0.25% of the genome) and a bacterial core genome MLST (100 regions of interest, each 10 kb long, covering in total 25% of the genome). All of our Read Until strategies enrich the coverage and minimum coverage of the regions of interest, and reduce the error of genotype reconstruction, compared to a naive sequencing run (see Fig 3 and Supplementary Figs 1 and 2). Minimum coverage is increased approximately 5-fold in the MLST scenario (Fig 3A), and about 2-fold in the cgMLST scenario (Fig 3C). This in turn leads to a dramatic reduction in the uncertainty of the reconstructed genome, with BOSS-RUNS strategy requiring far less sequencing to achieve the same quality of genome reconstruction than naive sequencing: about one quarter of the time in the

MLST scenario (Fig 3B) although more than half in the cgMLST scenario (Fig 3D).

In all our analyses, we simulated sequencing continuing for extended periods, to enable observation of both the short- and long-term performance of each approach. Indeed, BOSS-RUN strategies do seem to outperform naive sequencing in both the short- and long-term. While our strategies lead to higher coverage over the regions of interest than over the rest of the genome (Supplementary Figs 1C and 2C), they do not necessarily lead to higher coverage in these regions than a naive sequencing run (Supplementary Fig 2A) — an important feature showing that our strategies can adapt during a sequencing run to reject reads from regions of interest that have already achieved sufficiently high coverage compared to other regions of interest.

In both MLST and cgMLST scenarios, partial BOSS-RUNS (activating updates to the initial strategy only for the final ½ or ¼ of the sequencing run) seems preferable to updating the strategy from the start (full BOSS-RUNS; see Fig 3). This might seem counter-intuitive, but our strategy is optimized to gain the most benefit in the short term, and so may reject fragments early on that later might be more useful. For example, our strategy might reject reads from regions that have currently average coverage to focus instead on regions with currently low coverage; however, regions with currently average coverage might become regions with low coverage in the future, and so rejecting reads from these regions might not be advantageous in the long term.

Compensating for Sequencing Biases

One promising potential application of our strategy is the possibility of compensating for the inherent tendency of some genomic regions to be sequenced at higher coverage than others, possibly due to GC content among other factors [18, 19]. This could deliver a more homogeneous coverage, with the potential benefit of reducing genotype calling error and uncertainty in regions with low coverage. To explore these potential gains, we simulated bacterial genome sequencing runs with 10 peaks and troughs of coverage, and with about 10-fold difference in coverage rate (the rate at which fragments from different genomic regions are acquired by nanopores) between the peaks and the dips (our "coverage variability" scenario, see Methods).

BOSS-RUNS strategies, by focusing sequencing effort on regions of higher

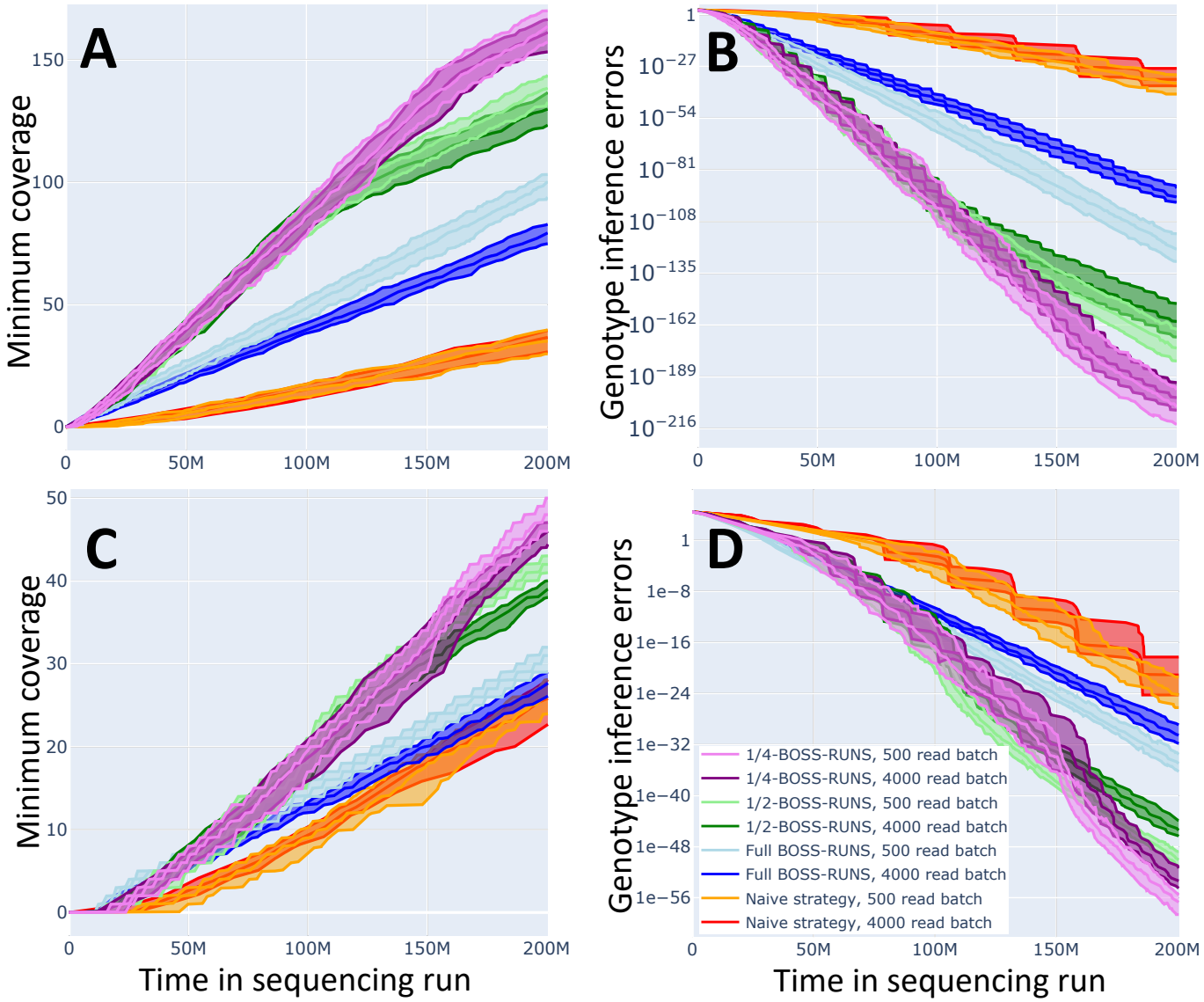

Fig 3. Different strategies' performance in the MLST and cgMLST sequencing scenarios. We compare the performance of different sequencing strategies in cases where we are only interested in a small part of a bacterial genome (0.25%, MLST scenario, plots **A** and **B**) or in a substantial portion of it (25%, cgMLST scenario, plots **C** and **D**). Plots **A** and **C** show the minimum coverage achieved within the regions of interest, as a function of time (x-axis). Plots **B** and **D** show the sum of the posterior probabilities of all wrong genotypes, over the regions of interest, and thus represent the expected total numbers of genotype reconstruction errors within those regions. Line colors show different strategies (legend in plot **D**; see Methods for details). Each color summarizes 30 replicates, with upper, central and lower boundary lines representing respectively the 90th, 50th and 10th percentiles. Red and orange lines represent naive strategies (respectively with batches of 4,000 and 500 reads); blue and azure lines represent full BOSS-RUNS strategies (updates from the start; respectively with batches of 4,000 and 500 reads); dark and light green lines represent ½-BOSS-RUNS strategies (updates only done in the final half of the sequencing run; respectively with batches of 4,000 and 500 reads); dark and light purple lines represent ¼-BOSS-RUNS strategies (updates only in the final ¼ of the run; respectively with batches of 4,000 and 500 reads).

uncertainty and thus typically of lower coverage, can substantially increase the minimum achieved coverage across the genome. While naive sequencing achieves a minimum coverage of 1–3x by the end of this hypothetical experiment, our BOSS-RUNS strategies achieve a minimum coverage of at least 4x (Fig 4A); the full BOSS-RUNS strategy, in particular, achieves a minimum coverage between 8–10x. Regions of low coverage are also typically the ones with the highest uncertainty, and, as a consequence, our strategies considerably decrease the number of errors in genome reconstruction. While at the end of the simulated runs naive sequencing shows about 100 errors, BOSS-RUNS strategies usually have less than one error (Fig 4B). Furthermore, the full BOSS-RUNS strategy is the fastest at achieving the mark of (e.g.) one error per genome, reaching it while naive sequencing still has usually more than 1000 errors. In this particular scenario, partial BOSS-RUNS results are not so good: it appears more efficient to perform the first strategy update as early as possible, probably because because now the strategy will not change much once it becomes clear which regions tend to have lower coverage. These improvements come at the sacrifice of overall average coverage, and in particular of coverage in regions with positive sequencing bias (regions with typically higher coverage): see Fig 5. This is typically not a problem, as these regions have sufficient data to infer the sequenced genotype with high confidence.

Whole Genome Sequencing in the Absence of Sequencing Biases

In the absence of inherent sequencing biases, and if we do not focus on specific regions of interest, it is harder to see how a dynamically updated Read Until sequencing strategy could be beneficial. However, there are some additional factors that should be considered. Even in the absence of inherent sequencing biases, some areas of the genome might receive higher coverage than others, simply due to random sampling of DNA fragments. Also, with linear chromosomes, as in our yeast sequencing scenarios, coverage tends to drop near the ends of chromosomes due to both mapping and library preparation effects. Further, even with uniform coverage, some sites might be of higher interest or might require more sequencing effort to genotype with certainty, for example heterozygous sites in a diploid genome or sites with indels. To investigate the usefulness of our strategy in this less favorable scenario and its ability to assist with the points

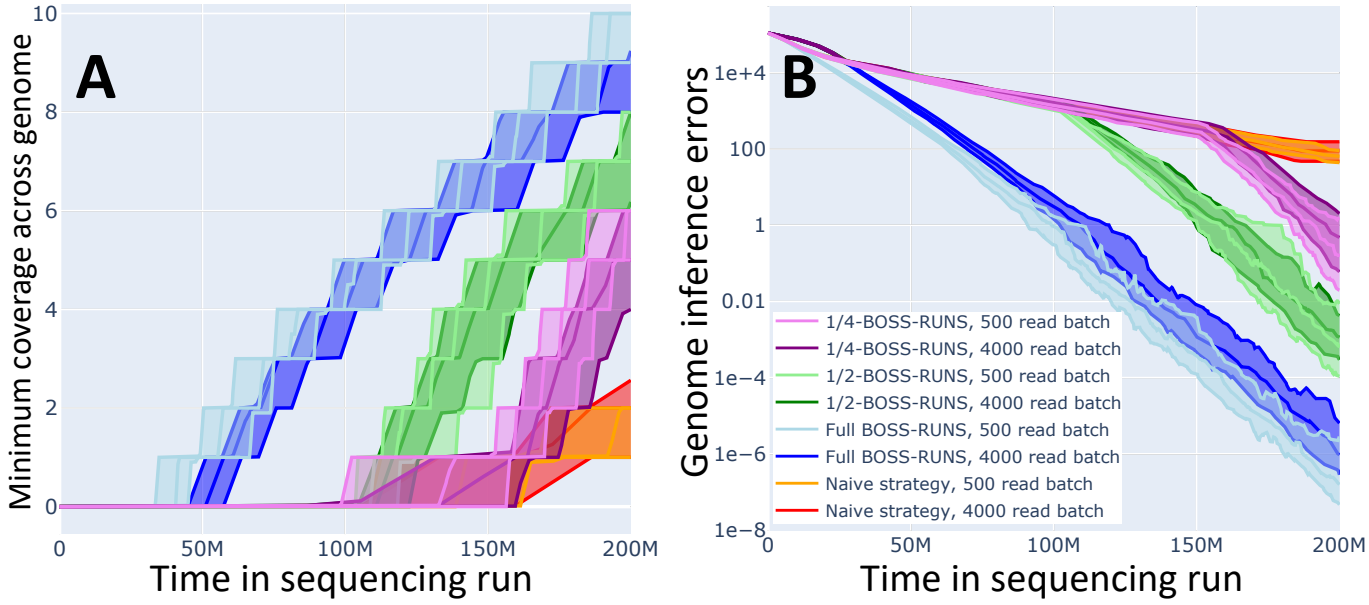

Fig 4. Different strategies' performance in the coverage variability sequencing scenario. We compare the performance of different sequencing strategies in a case where coverage varies substantially across the genome. Plot **A** shows the minimum coverage achieved, over time (x-axis), across the genome. Plot **B** shows the sum, over the genome, of the posterior probabilities of the wrong genotypes, and represents expected numbers of genotype reconstruction errors, as the sequencing run proceeds (time on the x-axis). Different line colors represent different strategies and batch sizes (legend in plot **B**; see Methods for details) as in Fig 3.

above, we simulated the sequencing of a bacterial genome without either specific regions of interest or inherent sequencing biases. Similarly, we simulated the sequencing of a haploid yeast genome and a diploid yeast genome.

In all these scenarios, BOSS-RUNS strategies can lead to significant benefits over a naive sequencing run, increasing minimum coverage and improving genome reconstruction (Fig 6 and Supplementary Figs 3, 4 and 5). Using smaller read batches (i.e. more frequent strategy updates) and performing the first strategy update later on, in particular, usually lead to the best results, up to almost doubling minimum coverage (Fig 6E). Overall, the benefits of all BOSS-RUNS strategies are consistent in the case of yeast genome, probably because of their ability to increase coverage towards the ends of chromosomes. In the case of a bacterial genome, running our full BOSS-RUNS strategy consistently throughout a long sequencing run can be counter-productive, as many reads are rejected early on that would have been more useful later (the strategy behaves 'greedily'; see Fig 6A and B). However, performing the first strategy update later in the

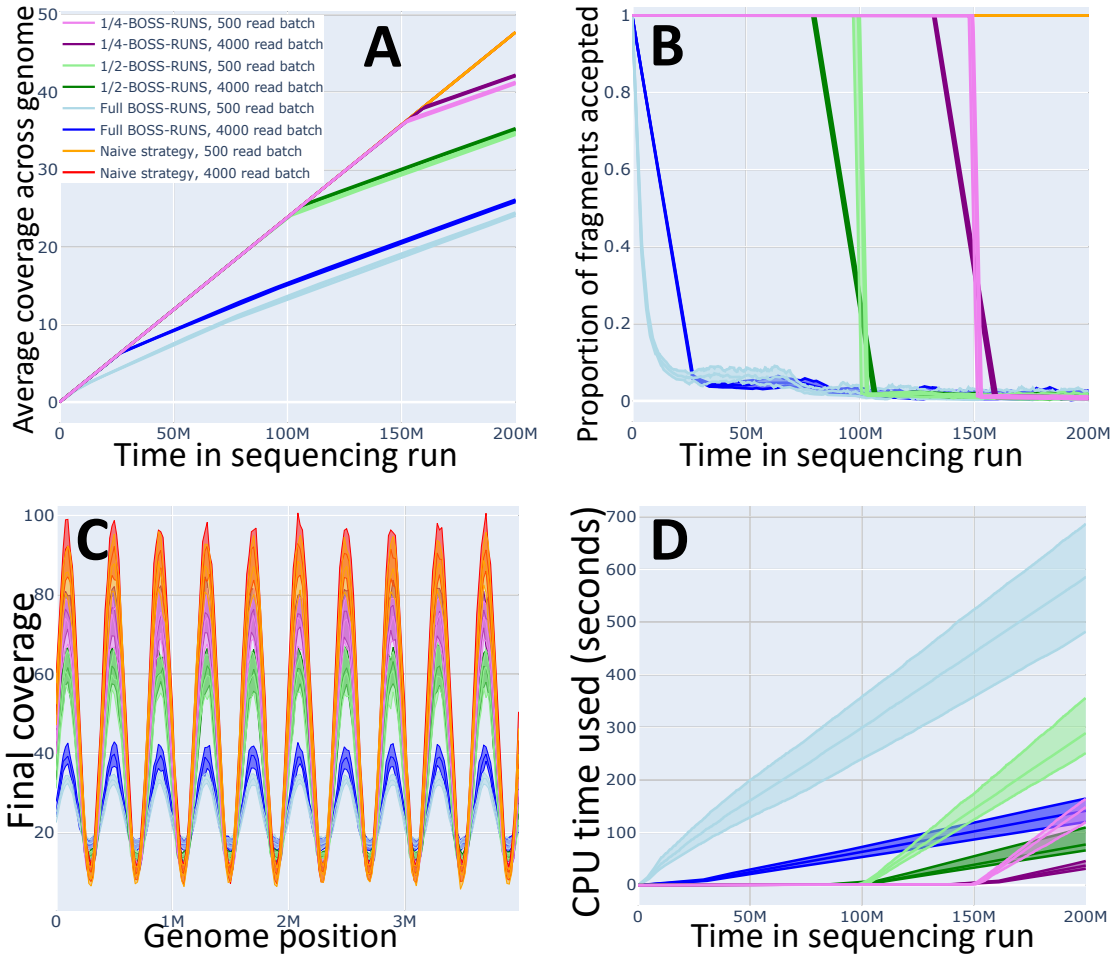

Fig 5. Other features of different strategies in the coverage variability sequencing scenario. Plot **A** shows how average coverage (y-axis) increases over time (x-axis) with different strategies. Plot **B** shows the proportion of fragments that are accepted by each strategy over time. Plot **C** shows the final coverage (y-axis) along the genome (x-axis), averaged over windows of 20 kb. Plot **D** shows the cumulative computational running times for updating the BOSS-RUNS strategy (y-axis, in seconds) throughout the sequencing run (x-axis, time in units of sequenced base pairs). Different line colors represent different strategies and batch sizes (legend in plot **A**; see Methods for details) as in Fig 3.

sequencing run (partial BOSS-RUNS) can more than compensate for this; furthermore, in the short term, the full BOSS-RUNS strategy is always beneficial compared to a naive sequencing run, and so can be useful even in the worst scenario if one aims to sequence up to a coverage of about 15x (compare Figs 6A and 3A).

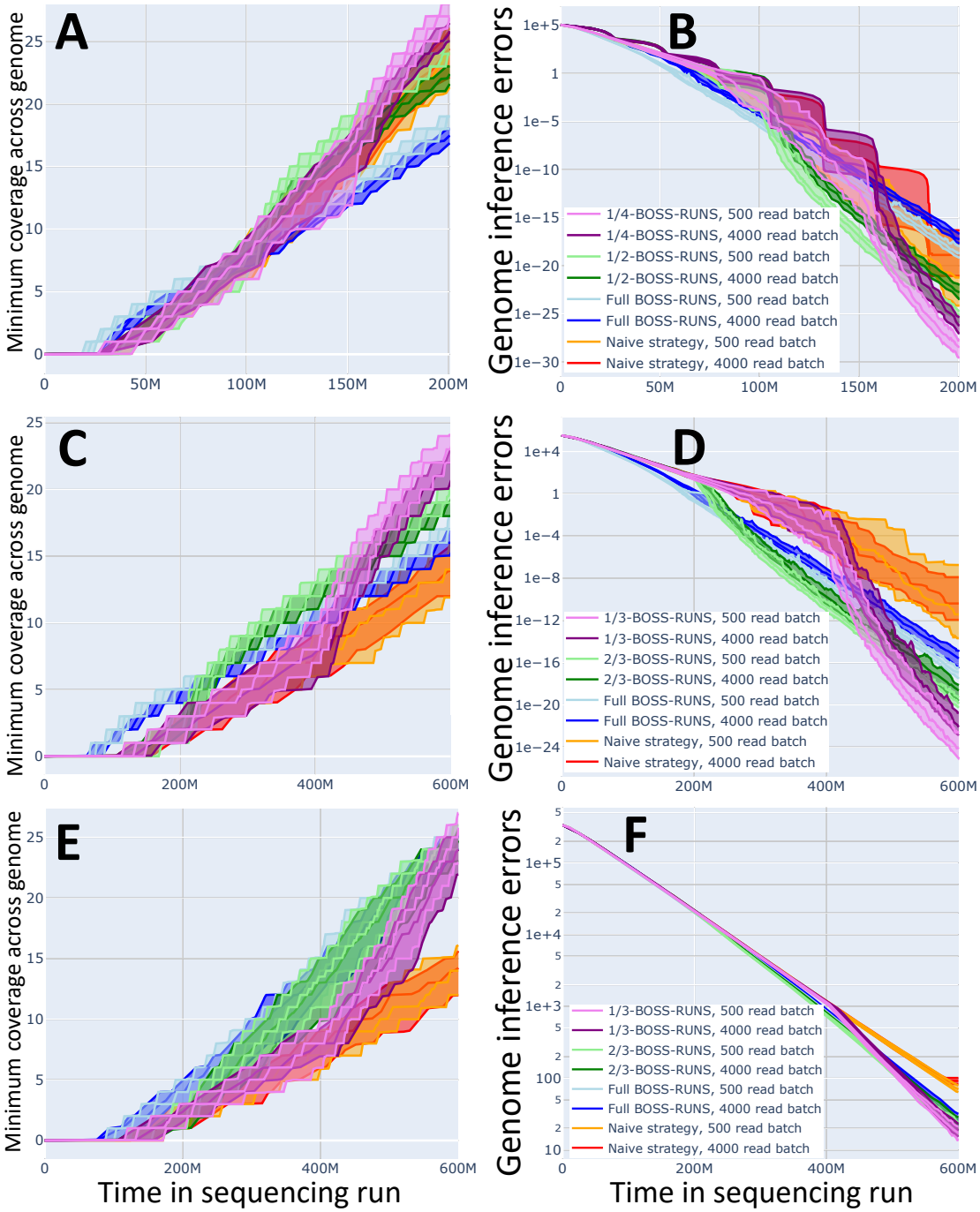

Fig 6. Different strategies' performance in the absence of coverage variability or regions of particular interest. We compare the performance of different sequencing strategies in the case of normal sequencing runs in bacterial (**A**–**B**), haploid yeast (**C**–**D**) and diploid yeast (**E**–**F**) simulation cases. Plots **A**, **C** and **E** show minimum coverage achieved across the genome, over time (x-axis). Plots **B**, **D** and **F** show the expected numbers of genotype reconstruction errors over time. Different line colors represent different strategies and batch sizes (legends in plots **B**, **D** and **F**; see Methods for details), similarly to Fig 3 although note the use of ⅔- and ⅓-BOSS-RUNS strategies in **C**–**F**.

Discussion

We have shown that, using our dynamic Read Until strategy (BOSS-RUNS), we can obtain great improvements over a naive nanopore sequencing run, in particular when sequencing is intended to be focused on specified regions of interest. Good results are also achieved in homogenizing coverage in the presence of inherent variability in coverage. Further, we have shown that a Read Until strategy can still be advantageous even in the absence of these factors, by focusing sequencing efforts in regions that, due to random chance, have received low coverage during the sequencing run, or by focusing on sites that have higher uncertainty in genome reconstruction.

In the future, we should explore the possibilities of parameterizing (or reparameterizing) the duration of the sequencing run and of modeling and estimating any inherent variability in coverage in real time, to further improve the strategy by not rejecting early on fragments that might later be considered useful. Similarly, real time updates to the regions defined as being of interest (as in the MLST and cgMLST scenarios) or to inferred sequencing biases could lead to improved Read Until strategies.

One aspect that we do not model is the possibility that fragment rejections would cause excessive pore blockage and thus loss of sequencing capacity [7]. More effort is needed in this respect both on our modeling side and on an engineering side. Another possible extension could be the modeling of variation in partial fragment length (μ), acquisition time (α), and rejection time (ρ), each currently assumed to take a constant time.

Our strategy works well on small genomes, for example for bacteria or yeast, but would suffer from slow update speed and high memory demand with larger genomes such as human. Further effort will be needed in future to devise faster, low-memory strategy updates. While our current implementation (in Python) makes heavy use of the NumPy package [20], and as such benefits from good computational performance, in the future further optimization could be possible by coding our methods in a fast, compiled language such as C or C++. However, scaling our methods to the size of the human genome might require re-thinking fundamental aspects of our strategy.

Another aspect that would benefit our strategy is the inclusion of more complex mutational events, such as insertions and rearrangements (see Supplement). These

events could also be given higher scores to reflect the fact that regions with such events are expected to be more important for genome assembly. Also, in some cases, a reference might not be available at all, requiring a different approach, possibly based on real-time *de novo* assembly.

Conclusions

We have shown that dynamically updated sequencing strategies that accept or reject potential DNA fragments based on their expected benefit can lead to considerable improvements in the performance within the context of nanopore sequencing, for example using ONT technology. Our methods expand the applicability of ONT's Read Until to encompass multiple standard sequencing scenarios: beyond simple enrichment in pre-selected areas of a genome, we show that it is possible, and convenient, to dynamically focus on areas with higher uncertainty, for example genomic regions that currently have lower coverage. This leads to sequencing runs with overall more homogeneous coverage and less uncertainty and error in genome reconstruction, or improved time-to-answer, or both. We think this has the potential to improve the quality and efficiency of ONT sequencing in the majority of its applications.

Supporting information

Code used for this project is available at
`https://bitbucket.org/nicofmay/readuntilstrategy/`

Supplement. File containing extensions of the methods and additional results.

Funding

This work was supported by the Biotechnology and Biological Sciences Research Council [grant number BB/N017099/1 to ML]. NDM, CM, EB and NG were supported by the European Molecular Biology Laboratory. CM was also supported by Murray Edwards College, Cambridge, and by the Cambridge Mathematics Placements (CMP) programme.

Conflict of Interest: ML was a member of the Oxford Nanopore Technologies 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. EB is a long-term paid consultant to Oxford Nanopore Technologies and a small-scale equity and options holder in Oxford Nanopore Technologies.

References

1. Jain M, Olsen HE, Paten B, Akeson M. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. Genome Biology. 2016;17(1):239.

2. Lu H, Giordano F, Ning Z. Oxford Nanopore MinION sequencing and genome assembly. Genomics, Proteomics & Bioinformatics. 2016;14(5):265–279.

3. Payne A, Holmes N, Rakyan V, Loose M. BulkVis: a graphical viewer for Oxford Nanopore bulk FAST5 files. Bioinformatics. 2018;35(13):2193–2198.

4. Workman RE, Tang AD, Tang PS, Jain M, Tyson JR, Razaghi R, et al. Nanopore native RNA sequencing of a human poly(A) transcriptome. Nature Methods. 2019;16(12):1297–1305.

5. Loose M, Malla S, Stout M. Real-time selective sequencing using nanopore technology. Nature Methods. 2016;13(9):751.

6. Edwards HS, Krishnakumar R, Sinha A, Bird SW, Patel KD, Bartsch MS. ReAl-time Selective Sequencing with RUBRIC: read until with basecall and reference-informed criteria. Scientific Reports. 2019;9(1):1–11.

7. Payne A, Holmes N, Clarke T, Munro R, Debebe B, Loose M. Nanopore adaptive sequencing for mixed samples, whole exome capture and targeted panels. bioRχiv. 2020;https://doi.org/10.1101/2020.02.03.926956.

8. Kovaka S, Fan Y, Ni B, Timp W, Schatz MC. Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED. bioRχiv. 2020;https://doi.org/10.1101/2020.02.03.931923.

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

10. Hui-Feng W, Huang F, Zhen G, Zheng-Li H, Yi-Lun Y, Bing-Yong Y, et al. Real-time event recognition and analysis system for nanopore study. Chinese Journal of Analytical Chemistry. 2018;46(6):843–850.

11. Kullback S, Leibler RA. On information and sufficiency. Annals of Mathematical Statistics. 1951;22(1):79–86.

12. Chaloner K, Verdinelli I. Bayesian experimental design: a review. Statistical Science. 1995;10(3):273–304.

13. Shannon CE. A mathematical theory of communication. Bell System Technical Journal. 1948;27(3):379–423.

14. Bowden R, Davies RW, Heger A, Pagnamenta AT, de Cesare M, Oikkonen LE, et al. Sequencing of human genomes with nanopore technology. Nature Communications. 2019;10(1):1869.

15. Maiden MC, Bygraves JA, Feil E, Morelli G, Russell JE, Urwin R, et al. Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. Proceedings of the National Academy of Sciences USA. 1998;95(6):3140–3145.

16. Pearce ME, Alikhan NF, Dallman TJ, Zhou Z, Grant K, Maiden MC. Comparative analysis of core genome MLST and SNP typing within a European *Salmonella* serovar Enteritidis outbreak. International Journal of Food Microbiology. 2018;274:1–11.

17. Goffeau A, Barrell BG, Bussey H, Davis R, Dujon B, Feldmann H, et al. Life with 6000 genes. Science. 1996;274(5287):546–567.

18. Ross MG, Russ C, Costello M, Hollinger A, Lennon NJ, Hegarty R, et al. Characterizing and measuring bias in sequence data. Genome Biology. 2013;14(5):R51.

19. Krishnakumar R, Sinha A, Bird SW, Jayamohan H, Edwards HS, Schoeniger JS, et al. Systematic and stochastic influences on the performance of the MinION nanopore sequencer across a range of nucleotide bias. Scientific Reports. 2018;8(1):3159.

20. Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. Computing in Science & Engineering. 2011;13(2):22.