SMURF-seq: efficient short-read sequencing on long-read sequencers

Rishvanth K. Prabakar

Quantitative and Computational Biology Section,
Department of Biological Sciences,
University of Southern California,
1050 Childs Way,
Los Angeles 90089, USA

April 21, 2020

Contents

List of Figures

Chapter 1

Introduction

Chapter 2

Background

2.1 Nanopore sequencing

2.2 Copy number variation and profiling

2.3 Prior protocols based on concatenating DNA molecuels

The concept of ligating short DNA molecueles to improve the efficiency of sequencing was introduced in serial analysis of gene expresson (SAGE) [], and subsequenty its variants such as LongSAGE and SuperSAGE []. SAGE

Chapter 3

Sampling molecules using re-ligated fragments (SMURF)-seq

3.1 Motivation

CNV profiling, and read-counting in general, can be done on nanopore sequencers with long reads following the standard sequencing procedure [1]. A typical Oxford MinION sequencing run generates approximately 500k reads (length $\sim$8 kb) [2, 3]. Read-counting applications in general do not benefit from longer reads beyond what is necessary for unique mapping to the reference genome. In these applications, for any fixed number of nucleotides sequenced, more information is obtained if those nucleotides are organized as more DNA molecules, rather than longer contiguous fragments.

In general, for a given sample of DNA, a nanopore instrument will generate more reads if the corresponding molecules are shorter. Once a molecule is loaded into a pore, the time spent sequencing is less for shorter reads. In addition, for a fixed amount of DNA, shorter molecules result in higher molar concentration when loaded onto the machine, increasing the rate at which each pore captures molecules [4, 5]. We verified this rationale by sequencing short DNA molecules (restriction enzyme digested normal diploid genome) using the Oxford MinION instrument. The sequencing run produced 2.58 million reads with a mean read length of 630.93 bp (data not shown). Using the same instrument, with SMURF-seq, we report here an average of 6.2 million mapped fragments per run, which is substantially more fragments than

directly sequencing short reads.

The most important factor in the performance of SMURF-seq over sequencing short molecules directly is that sequencing concatenated fragments effectively eliminates the pore reload time for all but the first fragment in each read. However, there are a variety of additional factors that favor further optimization of the approach employed by SMURF-seq. First, reduction of resources spent on technical nucleotides: SMURF-seq uses a single barcode and sequencing adapter per read consisting of multiple fragments; sequencing short reads uses one barcode and adapter per fragment, adding approximately 50 bases to each fragment. This increases the time to sequence each short read. In sequencing short reads, as the reads get shorter the time consumed by these technical bases increases. In SMURF-seq, sequencing either shorter fragments in fixed length reads, or longer reads containing fragments of fixed average length, both reduce the time consumed sequencing these technical bases. In the limit, assuming 100bp DNA fragments, sequencing those fragments as short-reads corresponds to 33% technical nucleotides; for SMURF-seq, the portion of technical nucleotides remains low. Second, more nucleotides sequenced at full speed: We observed that the speed of sequencing was lower when sequencing short molecules. For example, the average sequencing speed was 315.54 bases per second for sequencing the diploid genome without SMURF-seq, and 400.29 bases per second when sequencing using SMURF-seq on the MinION sequencer. Third, leveraging optimizations to long-read protocols: The rapidly evolving nanopore library construction kits are continually optimized for long-read sequencing, and would likely require significant ad-hoc modifications to optimize sequencing of short molecules of length optimal for read-counting applications. SMURF-seq alleviates these drawbacks by using the nanopore instrument as intended for long-read sequencing, while generating the desired short fragments.

3.2 SMURF-seq protocol

The SMURF-seq protocol involves cleaving genomic DNA into short fragments, with length just sufficient for an acceptable rate of uniquely mapping fragments in the reference genome. These fragmented molecules are then randomly ligated back together to form artificial long DNA molecules, as required for long-read sequencing. The long re-ligated molecules are sequenced following the standard MinION library preparation protocol. After (or possibly concurrent with) sequencing, the SMURF-seq reads are mapped to the reference

genome in a way that simultaneously splits them into their constituent fragments, each aligning to a distinct location in the genome (Fig. 3.1).

More specifically, genomic DNA is fragmented using restriction enzymes and ligated with T4 DNA ligase, with clean-up steps in between. SMURF-seq protocol is completely enzymatic and takes less than 90 minutes to complete (Fig. 3.2). The details of these steps are given below:

1. Restriction enzyme digestion: restriction enzymes recognize and cleave specific DNA sequences, typically producing sticky-ended DNA molecules. The choice of restriction enzyme used is primarily dependent on the size of the fragmented molecules produced. Based on the downstream application, they could also be influenced by other factors such as any bias they could introduce. An advantage of using restriction enzymes to fragment DNA molecules, over other fragmentation techniques, is that the fragmented molecules have a uniform ends (either overhangs with the same sequence or blunt-ends) and are thus compatible for ligation without an end-repair step in between.
2. Clean-up: the reaction containing the restriction enzymes and the fragmented DNA molecules is cleaned to wash out the enzymes and retain the DNA molecules. The choice of clean-up kit used, also determines the length of the DNA molecules that are retained. We used a spin-column based clean-up that typically

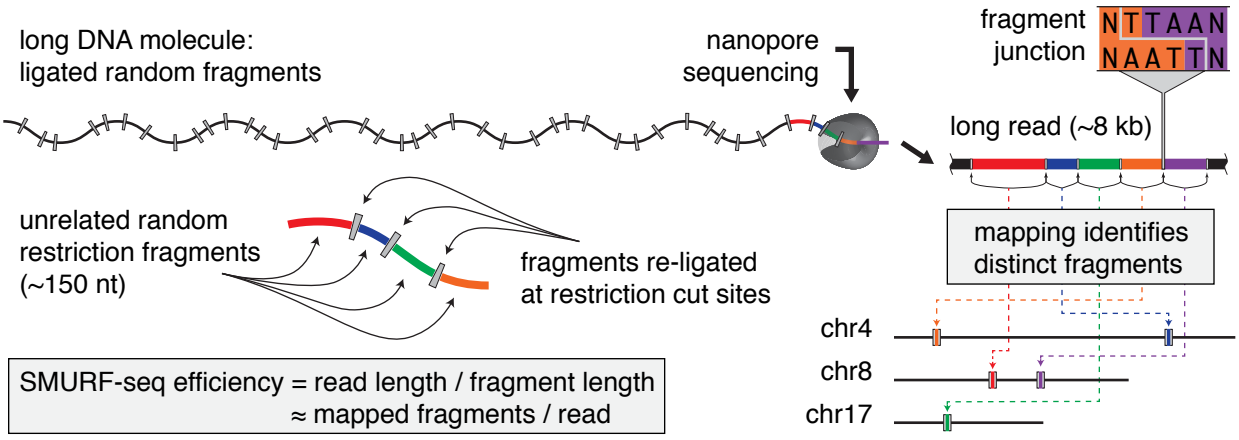

Figure 3.1: SMURF-seq efficiently sequences short fragments of DNA for read-counting applications with a reference genome on long-read sequencers, and yields up to 30 countable fragments per sequenced read. SMURF-seq sequences short DNA molecules by generating long concatenated molecules from these. SMURF-seq reads are aligned by splitting them into multiple fragments, each aligning to a distinct region in the genome.

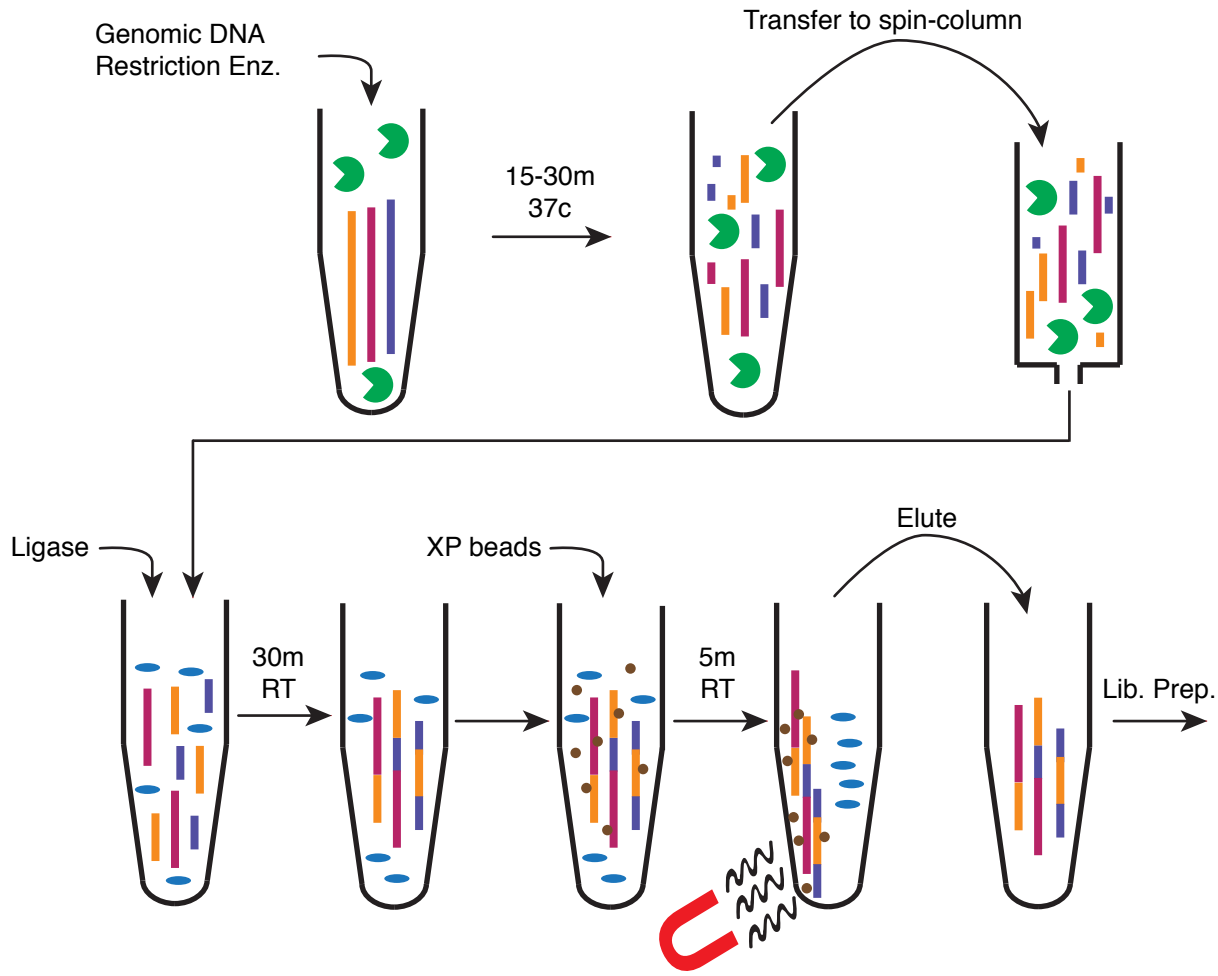

Figure 3.2: Schematic of SMURF-seq protocol. SMURF-seq consists of four steps: restriction enzyme digestion, spin-column clean-up, re-ligation of fragmented DNA, and Ampure XP beads clean-up.

retains molecules that are over ~70 bp. However, other clean-up kits, such as bead-based kits, could also be used at this step.

3. Re-ligation: fragmented DNA molecules with uniform ends are ligated at random with T4 DNA ligase enzymes. The most important factor in a ligation reaction is the concentration of compatible DNA ends [6]. At high concentrations, the chances are higher for ligation between two molecules than a molecule self-ligating. At low concentrations, the chances are higher for self-ligation. Thus, the main consideration during the ligation step is the duration of the ligation reaction, as the molar concentration of DNA molecules decrease with time. Too little time would lead to insufficient ligation, resulting in molecules

of length that do not achieve optimal SMURF-seq efficiency. On the other extreme, too much time would result in circular molecules that are incompatible with the most downstream library preparation process. A typical ligation reaction would contain both short and circularized molecules, and achieving a balance between these determines the efficiency of SMURF-seq. Other factors such as the temperature and buffer contents also affect the ligation process. In our experiments, the ligation reaction was performed at a DNA concentration of $25\,\mathrm{ng}/\mu\mathrm{l}$ ($500\,\mathrm{ng}$ of DNA in $10\,\mu\mathrm{l}$ nuclease free water and $10\,\mu\mathrm{l}$ DNA ligase) for $30\,\mathrm{min}$.

4. Bead-based clean-up: the reaction containing the ligase enzymes and ligated DNA molecules is cleaned to retain only the ligated molecules. We used a bead-based clean-up at this step to avoid damage to long DNA molecules that are typical of spin-column based methods.

DNA molecules processed with the SMURF-seq protocol are compatible with any standard long-read library preparations kits that are available.

We also tested dsDNA Fragmentase enzymes (New England Biolabs) and acoustic shearing (Covaris) to fragment DNA. However, these methods require an additional end-repair step after fragmentation and the ligated molecules failed to reach the lengths we obtained by using restriction fragmentation (data not shown).

In our applications, we used SaqAI restriction enzyme, which recognizes the sequence TTAA and produces molecules with mean lengths of 150.2 bp (Fig. 3.3a). The fragmented DNA molecules are then ligated randomly to form longer molecules using T4 DNA ligase enzyme (Fig. 3.3b). In our experiments, the resulting long DNA molecules were sequenced using the Oxford Nanopore Technologies 1D DNA by ligation kit (SQK-LSK108) or the rapid sequencing kit (SQK-RAD003) following the standard manufacturers protocol.

3.3 Mapping SMURF-seq reads

The reads sequenced using SMURF-seq can be mapped to a reference genome by first identifying short matches within the reads, corresponding to parts of the individual fragments, and then extending those to locate fragment boundaries. This is handled nicely using the seed-and-extend paradigm implemented in many existing long-read mapping tools.

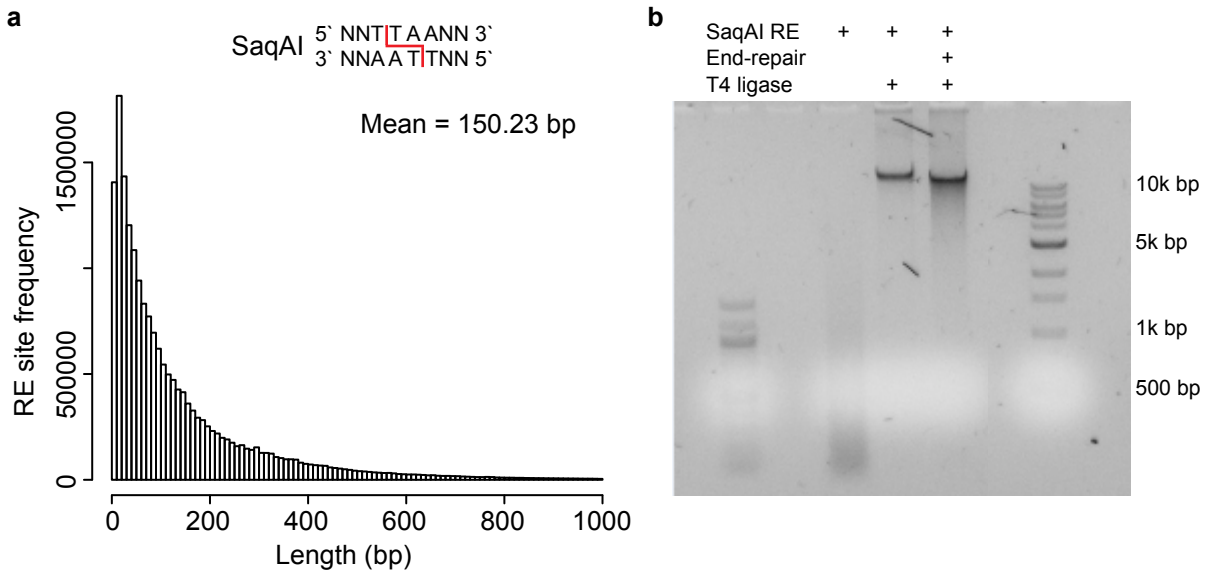

Figure 3.3: Restriction digestion and ligation of DNA molecules. (a) Distribution of length between restriction sites computed by measuring the distance between the recognition sites on the human reference genome. SaqAI recognizes the sequence TTAA and leaves a 2 bp overhang. (b) Negative gel image of fragmented and ligated normal diploid DNA using SaqAI restriction enzyme and T4 DNA ligase. Sticky-end and blunt-end ligation (by end-repair) of fragmented DNA are shown, and both yield ligated molecules of approximately the same length.

3.3.1 Simulating SMURF-seq reads to evaluate mapping programs

To test these mapping tools, we chose to create simulated reads with the technical characteristics we expect in idealized SMURF-seq data. We first selected a fragment length ℓ and a number k of fragments per read. Then, for a given WGS nanopore data set, we took the set of mapped long reads as determined by BWA-MEM (with `-x ont2d` option). Each of the mapped reads was split into fragments of length ℓ (with a random offset of 0 to $\ell - 1$ at the start of the long read). Each fragment was validated by requiring that it did not overlap a deadzone in the genome (as determined by the deadzone program available from https://github.com/smithlabcode/utils for 40 bp). The reason for excluding deadzones is that even when a short fragment has a "known" mapping location when it is part of a longer read, we cannot compare its reported mapping location as a short fragment with that known location, since we expect any good mapping algorithm to identify that the fragment maps ambiguously. Among these validated fragments, subsets of k were sampled uniformly at random and concatenated (in random order and orientation) to form simulated SMURF-seq reads.

The first and last fragments in a read should be slightly easier to identify and map than the rest, since one of their boundaries is known. Using the above procedure, we select $k = 20$ so that the simulated reads have a sufficient number of fragments to eliminate the influence of the first and last fragments in each read on the results. There is no need to have large k otherwise.

By lowering ℓ and making the fragments shorter, the task of mapping the fragments becomes more challenging. Real SMURF-seq reads have fragment lengths determined by restriction site density, size selection and other aspects of the experiments. But in testing mapping algorithms and optimizing parameters, there is no disadvantage to making the task more challenging. We only need to be able to distinguish the relative performance of different mapping tools and parameter combinations. Real SMURF-seq reads have varying fragment lengths, but in evaluating mapping tools, there is no need to randomize fragment lengths. None of the algorithms we evaluated are capable of either deducing or leveraging the fact that all simulated fragments have the same length. We selected $\ell = 100$, which begins to challenge the various mapping strategies. These values of ℓ are slightly lower than the average in real SMURF-seq data.

3.3.2 Evaluating performance using simulated SMURF-seq reads

Within the simulated reads, the boundaries of each fragment are known *a priori*, as are their mapping locations. We used this information to evaluate mapping tools in terms of (1) how well they identify fragments purely for the purpose of counting molecules, which is the primary information used in CNV analysis, and (2) how well they identify individual mapping bases within reads. The latter criteria becomes important in challenging cases and will be increasingly important as fragment sizes are reduced.

Performance on identifying fragments: After mapping these simulated reads, each mapping result is called a predicted fragment. Each predicted fragment is considered a positive prediction, and we assume an arbitrary order over positive predictions. A positive prediction is a true positive if:

- The predicted fragment maps uniquely.
- The mapping locations of at least half the bases in the predicted fragment are equal to the original mapping locations for those bases, and those bases are all part of the same original fragment (we assume that it is unlikely for two fragments on a simulated read to have the same mapping location but opposite orientation, and thus do not check for the orientation of a fragment). In this case, we say the predicted fragment is

associated with that original fragment.

- The predicted fragment is the first among predicted fragments associated the same original fragment.

False positives are predicted fragments that are not true positives. Any original fragment with no associated predicted fragment is a false negative. These criteria penalize splitting one original fragment or merging two original fragments. By defining true positives, false positives and false negatives we are able to calculate precision, recall, and F-score for a particular mapping strategy.

Performance on identifying individual mapping bases: After mapping simulated reads, each mapping result is decomposed into individual nucleotides and associated with a location in the genome. Those locations are retained. We keep multiplicities, so when two mapped fragments overlap in the genome we count certain nucleotides twice. These are the predicted positive bases in the reference. The condition positive bases are those known *a priori* from the simulation. The original fragment mapping locations may overlap in the reference genome, leading to multiplicities in the condition positive bases, but with low probability. The true positives are the intersection of the condition positive and the predicted positive bases. When there are multiplicities of mapped fragments and simulated fragments overlapping the same bases in the reference genome, this is determined by taking the smaller of the two values. After removing the true positives bases, the remaining predicted positive bases are false positives, and the remaining condition positive bases are false negatives. These criteria penalize mapping approaches that do not cover the entire simulated SMURF-seq reads, and also penalize approaches that predict fragments that overlap within the read. The true positives, false positives, and false negatives here allow us to assign precision and recall in terms of individual bases and corresponding F-scores. Although the reference bases for both predicted positive and condition positive could involve multisets, since our simulations used relatively low coverage this almost never happened.

3.3.3 Data sets for generating simulated SMURF-seq reads

To generate simulated reads we used the standard long reads from four sequencing runs (Flowcell ID: FAB42704, FAB42810, FAB49914, and FAF01253) in the public dataset available at https://github.com/nanopore-wgs-consortium/NA12878/blob/master/Genome.md [2, 7]. We downloaded the raw data from EBI (Run accession: ERR2184696, ERR2184704, ERR2184712, and ERR2184722)

and base-called these with Guppy (version: 2.3.5).

3.3.4 Initial selection of mapping tools

We tested the following mapping tools: BWA-MEM[8], Minimap2[9], LAST[10], GraphMap[11], BLASR[12], rHAT[13], and LAMSA[14]. These were selected either because they are known to perform well on certain mapping tasks or have unique properties that plausibly could help in mapping SMURF-seq reads. We tested each of these using default parameters on simulated reads (see above) and downsampled real SMURF-seq reads (data not shown). Among these BWA-MEM, Minimap2, and LAST had higher accuracy on simulated data, and the other tools identified at most 15 fragments per read on real data. Thus, we explored performance of BWA-MEM (0.7.17), LAST (963), and Minimap2 (2.15) in more detail, varying parameters to improve performance.

We remark that none of these tools were designed to map SMURF-seq reads; results we report here do not reflect the overall performance of the various mapping tools, only that the three aforementioned tools happened to perform relatively well on a task for which they were not directly designed for.

3.3.5 Detailed evaluation and parameter optimization

The selected mapping tools have variations on the following basic steps:

- Identifying seeds: All tools have a step of identifying seeds, which are short exactly matching parts of the reads. Choices in how seeds are defined and used are often made for mapping speed. The total size of SMURF-seq data sets is currently (relatively) small, so speed is not our primary concern. We favor the most sensitive seed strategy, but depending on implementation too many seed hits could lead to ambiguity later in the mapping process.
- Chaining seeds: The identified seeds are further extended and filtered to avoid aligning potentially false positive seed hits.
- Aligning within the chains: In this stage a Smith-Waterman alignment is performed, typically allowing users to specify a mismatch penalty along with penalties for both gap-open and gap-extend.
- Selecting best alignments: When high-scoring alignments overlap within a read, one of them (or both) could be trimmed or one is selected and the other discarded. The choices made here could lead to dis-

carding entire fragments.

These mapping tools have several parameter options, in general, these are related to: (1) the seeding and chaining algorithm used by the individual tool. (2) The Smith-Waterman alignment scores, i.e. the match score, and the mismatch and indel penalty. The seeding and chaining parameters control the number of proto alignments that are further refined by aligning parts of the read to the reference genome using the specified alignment scores.

The Smith-Waterman alignment score used to align fragments to the reference genome is crucial for determining the optimal fragment length. On one extreme, a match score of 1 with a mismatch and indel penalty of 0 will result in one identified fragment covering the entire read and mapping perfectly, but will always map ambiguously. On the other extreme, a match score of 1 with a mismatch and indel penalty of $-\infty$ will result in any mismatch or indel on the read to be considered as a fragment boundary. Therefore to align SMURF-seq reads, we need to determine optimal alignment scores to use.

In order to determine the optimal alignment score, we kept the seeding related parameters constant, and varied the alignment score combinations to perform a grid search. We varied the mismatch penalty from 1 to 6, gap open penalty from 0 to 4, and gap extend penalty form 1 to 4. The match score was fixed at 1. Thus for each tool we tested 120 ($6 \times 5 \times 4$) combinations of alignment scores.

The seeding and chaining related parameters for each tool was set at follows (along with the four alignment scores):

- BWA-MEM: `-x ont2d -k 12 -W 12 -T 30`
- Minimap2: `-w 1 -m 10 -s 30`
- LAST (NEAR): `lastal -Q0 -e 20` and `last-split -m 1 -s 30`

We set the seeding and chaining parameters in a liberal manner to allow for higher sensitivity than the default parameter of each tool, and the minimum alignment score to output was set at 30.

After aligning the simulated reads, we calculated the average precision and recall, each for the mapped fragment locations and nucleotides, for the four datasets. The F-score was computed for each, and the mean of the F-scores was used to determine the optimal alignment parameter for each tool. Based on these results BWA-MEM outperformed other tools for aligning SMURF-seq reads. BWA-MEM performed best with a mismatch, open, and extension penalty of 2, 1, 1 respectively.

To further refine the optimal alignment parameter for BWA-MEM, we aligned the simulated reads with parameter values around the value described above with a higher resolution. We varied the mismatch penalty from 1.5 to 2.5, and open and extend penalties from 0.5 to 1.5 in increments of 0.25. However, BWA-MEM does not accept floating point values for alignment score parameters. To overcome this, we scaled the alignment score proportionately to have integer values, i.e we varied the mismatch penalty from 6 to 10, open and extend penalties from 2 to 6, and fixed the match score at 4 (125 combinations). Based on these results, the highest accuracy was obtained with the mismatch, open, and extension penalty of 2.5, 1.5, 0.75 respectively (corresponding scaled values are 10, 6 and 3). We used these optimal alignment scores for mapping real SMURF-seq read, and all the CNV profiles presented are based on these.

3.4 Efficient CNV profiling using SMURF-seq

To demonstrate the utility of SMURF-seq, we generated CNV profiles of normal diploid and highly rearranged cancer genomes. The mapped fragments were grouped into variable length "bins" across the genome and bin counts were used to generate CNV profiles as described in [15, 16].

3.4.1 Normalizing restriction site bias

3.4.2 Accurate CNV profiles using SMURF-seq

We sequenced a normal diploid female genome with SMURF-seq, resulting in 270.8k reads (mean read length of 6.75 kb) in a single run. These reads were split into 7.28 million fragments (26.87 mean fragments per read). A CNV profile for this normal diploid genome, with the expected (approximately flat) appearance can be seen in Fig. 3.4a. We verified that the SMURF-seq procedure behaves similarly using the Rapid Sequencing Kit (the 213.38k sequenced reads had a mean read length of 3.9 kb, and were split into 2.81 million fragments). Next we applied SMURF-seq to the breast cancer line SK-BR-3, generating 147.0k reads with mean length of 7.62 kb, which were split into 4.52 million fragments (30.78 mean fragments per read). We then obtained a CNV profile using 5,000 bins, corresponding to an average bin size of approximately 600 kb (Fig. 3.4b).

To provide a quantification of accuracy in terms of individual CNV events we conducted whole-genome

sequencing (WGS) on the same SK-BR-3 using Illumina (5.56 million reads; 130 bp, single-end). We used this to define a ground truth by calling CNV events for each of the pre-defined bins (both amplifications and deletions) based on segmented signal with a cutoff of 1.25/0.8 (Fig. 3.4b) [17, 18]. This resulted in 1,466 events (886 amplifications, 580 deletions) from 4,953 bins. We then called events using the identical procedure with SMURF-seq data from the same SK-BR-3 sample. The precision and recall for SMURF-seq relative to the Illumina calls was 0.982 and 0.988, respectively (Fig. 3.4c). Fig. 3.4d shows a zoom-in of a region with extreme copy number alterations. The bin ratios for the Illumina WGS and the SMURF-seq profiles are highly correlated (Pearson $r = 0.99$; Fig. 3.4e). Replicates for these genomes show a high degree

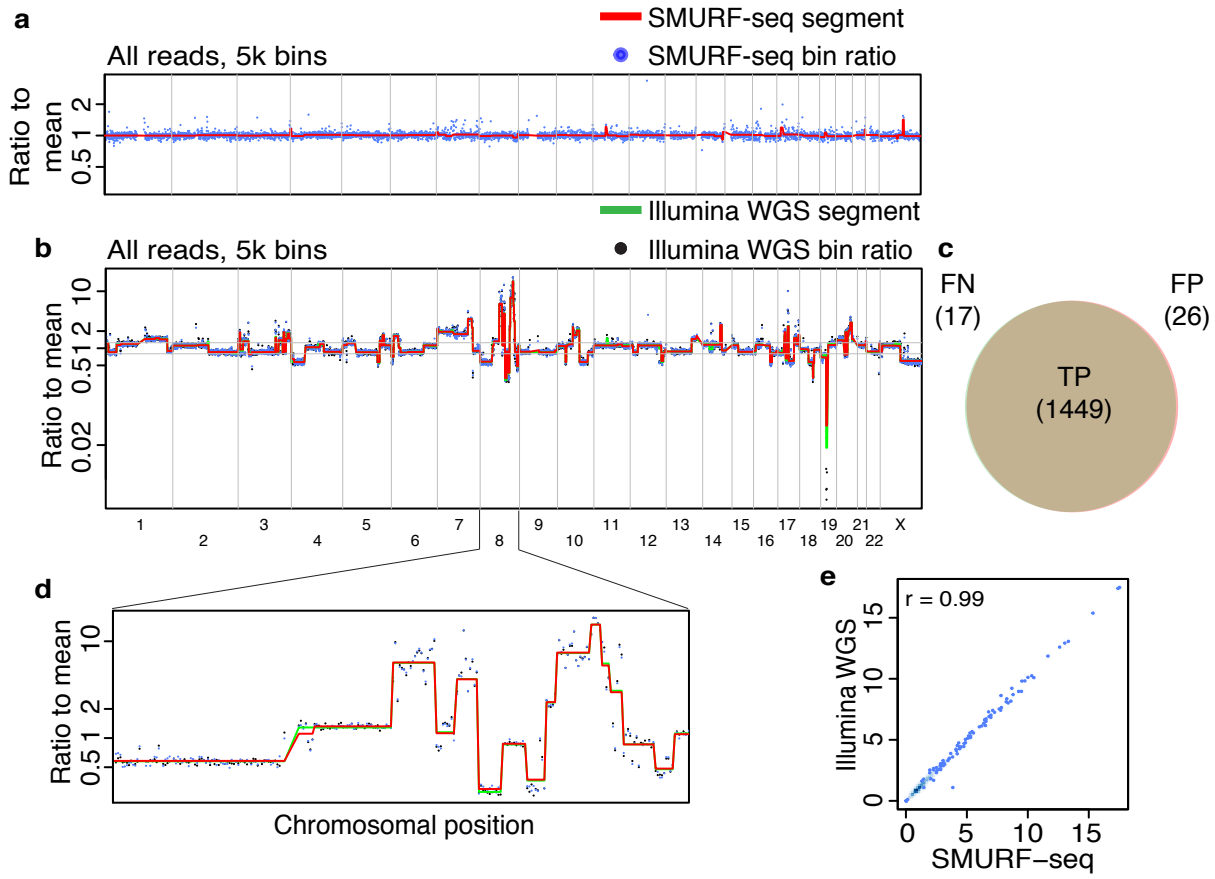

Figure 3.4: Accurate copy number profiles with SMURF-seq. (a) CNV profile of a normal diploid genome. Each blue point is a bin ratio to mean and the red line is the segmented bin ratio. (b) Superimposed CNV profiles of SK-BR-3 genome generated using SMURF-seq and Illumina WGS reads. (c) Venn diagram illustrating the accuracy of event calls using SMURF-seq compared with Illumina WGS. (d) Zoom-in of copy number changes on chromosome 8. (e) Scatter plot of bin ratio of SK-BR-3 genome using SMURF-seq and Illumina WGS reads. Pearson correlation of the data is shown.

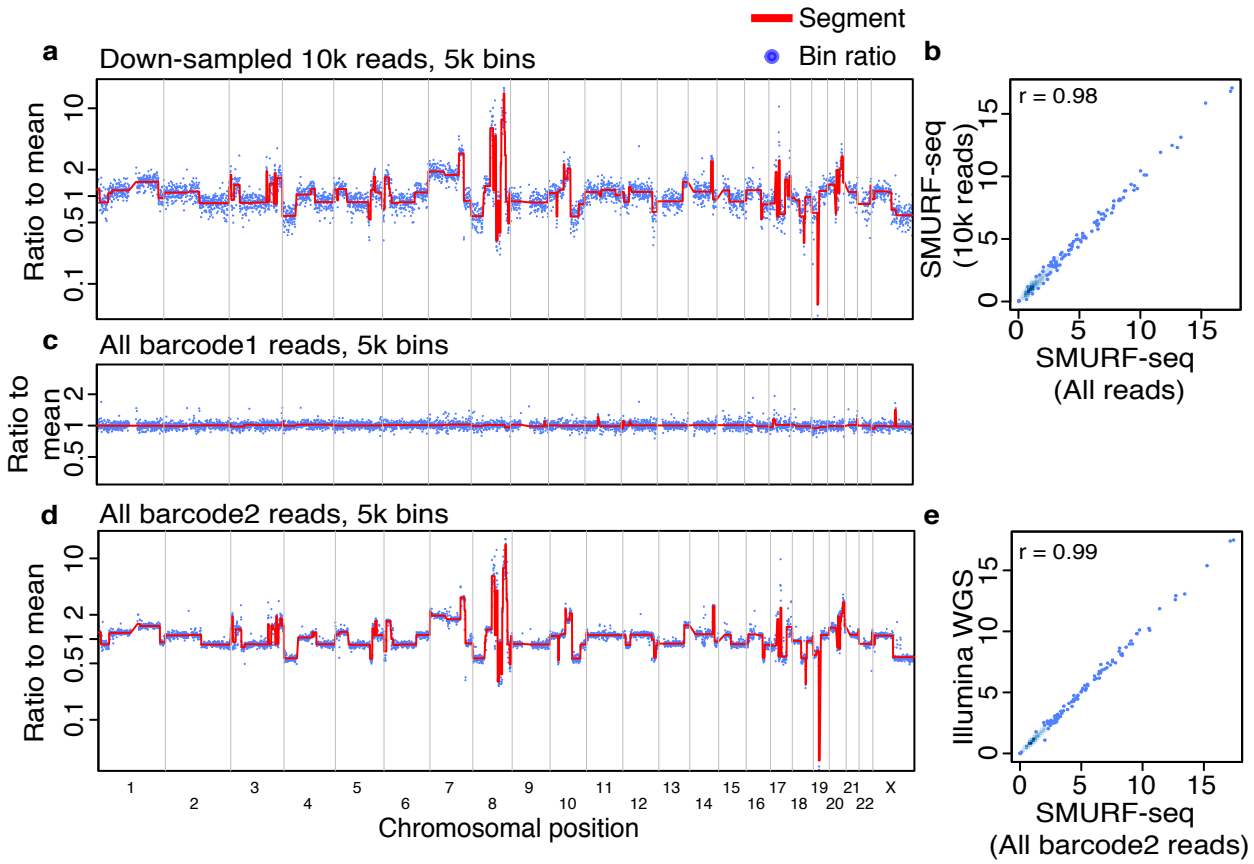

Figure 3.5: Multiple SMURF-seq CNV profiles by multiplexing in a single run. (a) CNV profile of SK-BR-3 genome with down-sampled 10k SMURF-seq reads. (b) Scatter plot of normalized bin counts of the original SMURF-seq data and data down-sampled to 10k SMURF-seq reads. Pearson correlation of the data is shown. (c) CNV profile of barcode01 (Normal diploid genome) reads. (d) CNV profile of barcode02 (SK-BR-3 cancer genome) reads. (e) Scatter plot of bin ratios of SK-BR-3 genome using multiplexed SMURF-seq and Illumina WGS reads.

of reproducibility for these profiles (data not shown).

3.4.3 Concordant profiles from fewer countable fragments

Several cancer-related studies have employed CNV profiling based on low-coverage WGS [19, 20]. It has previously been demonstrated that 250k reads are sufficient for accurate genome-wide CNV profiling of single cells [21]. At the same time, the CNV profiles from a population of cells has been shown to have a high correlation with single-cell profiles [22, 21]. We reasoned that using 250k fragments for CNV profiling using a population of cells would give useful profiles if they remained sufficiently accurate. By

down-sampling our SMURF-seq data, we verified that 10k reads, approximately 250k fragments, result in highly-correlated CNV profiles (Pearson r = 0.98; Fig. 3.5a, b).

Given the total capacity of the MinION instrument, this indicates that multiple samples can effectively be barcoded and multiplexed in a single sequencing run. To verify this we sequenced two DNA samples (normal diploid female and SK-BR-3) in a single run. These samples were processed with SMURF-seq protocol and then barcoded following the standard library construction. After demultiplexing and mapping the reads, the diploid genome had a CNV profile as expected (Fig. 3.5c) and the SK-BR-3 CNV profile was nearly identical to the profile obtained using Illumina WGS (Pearson r = 0.99; Fig. 3.5d, e).

Chapter 4

Identifying fragment boundaries on a SMURF-seq read

4.1 Motivation

New sequencing methods motivate development of new algorithms for mapping and analysis of sequences generated using these methods. A few significant developments include BLAST [23] and FASTA [24] motivated by database searches with the advent of Sanger sequencing, BWA [25] and Bowtie [26] inspired by high-throughput short-read sequencing, and BLASR [12] by single-molecule long-read sequencing. SMURF-seq has enabled efficient short-read sequencing for read-counting applications on portable long-read machines. However, efficient methods tailored for mapping SMURF-seq reads are still lacking; Especially as SMURF-seq protocol evolves and the fragments become shorter, and thus, making the mapping process challenging in terms of identifying accurate fragment locations and boundaries.

As currently implemented, SMURF-seq protocol uses a single restriction enzyme (SaqAI) to fragment DNA molecules to $\sim$150 bp. However, depending on the downstream application, the fragment lengths need to be just long enough to ensure unique mappability to a sufficient fraction of the genome. Fragments could be made shorter using methods discussed in section **??**. As an example, for copy-number profiling (at low resolutions, as used for tumor samples) the fragment lengths could be as short as 40 bp.

We used BWA-MEM [8] to align SMURF-seq reads generated with the current protocol, which consists

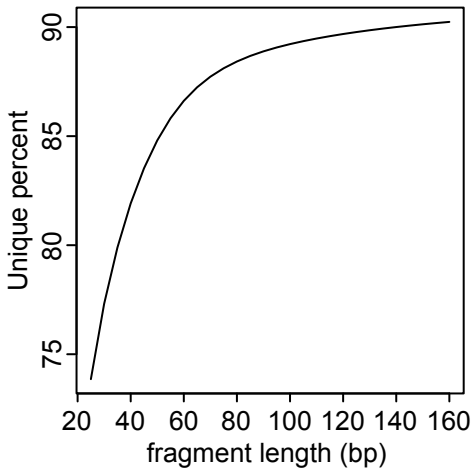

Figure 4.1: The fraction of genome that is uniquely mappable decreases with fragment length.

of fragments that are typically over 100 bp. Though not designed to align SMURF-seq reads, BWA-MEM is designed for split read alignment, and it works sufficiently well at these fragment lengths. SMURF-seq reads can also be aligned with other mapping tools capable of split-read alignment such as Minimap2 [9] and LAST [10]. However, all of these tools are either designed for aligning short reads with low sequencing error or long reads with high sequencing error. Moreover, since these tools are not designed for SMURF-seq reads, they lack certain capabilities; e.g. they do not provide a method to estimate the number of fragments or determine the optimal fragment boundaries in a SMURF-seq reads, and these become increasingly important with shorter fragments.

The significance of having accurate fragment boundaries is understood by looking at the fraction of the genome that is uniquely mappable. For the human reference genome (hg19), when allowing no mismatches or indels, the fraction of the genome that is uniquely mappable reduces by 0.06% when going from 150 to 145 bp, whereas it reduces by 2.02% when going from 40bp to 35bp (Fig. 4.1). Thus, as the fragments get shorter, the probability of a fragment that originated from a unique location on the genome to misalign to an ambiguous location or vice-versa increases. Further, as sequencing errors are considered the difference in unique mappability due to having inaccurate fragment boundaries is likely to increase. Thus, as the fragments become shorter, having accurate fragment boundaries would improve the sensitivity of aligning SMURF-seq reads.

We define the fragment identification problem for identifying the number of fragments and the fragment

boundaries on a SMURF-seq read. We approach the fragment identification problem by defining a score function for aligning a SMURF-seq read, study the null score distribution of aligning reads and reference generated at random, and estimate the number of fragments in a SMURF-seq read by comparing its alignment score with the null distribution for all possible fragmentations of a read. Then we show the accuracy of our method using from simulated genomes and SMURF-seq reads. Further, we empirically show that this method could also be used with a general score function.

4.2 Background

In the early days of DNA sequencing, as the number of nucleotides sequenced grew, comparison of DNA sequences became an indispensable tool to a biologist. DNA sequence comparison can be broadly classified into global alignment [27] and local alignment [28]. A global alignment seeks an optimal alignment between two sequences such that each base of one sequence is aligned to each base of the other sequences. On the other hand, a local alignment seeks an optimal alignment between any subsequences of the sequences being compared.

Comparison of two sequences, even unrelated or random sequences, always produces an optimal alignment. This motivated the development of approaches to differentiate a "meaningful" alignment from alignment of unrelated sequences. These methods determine the significance of an alignment by comparing the alignment score with a null distribution of alignment scores of unrelated sequences. Determining the appropriate null distribution was the subject of an enormous amount of research, some of which are summarized below.

In the context of local alignment, at the time of the initial studies on the score distribution of unrelated sequences, mathematical tools to understand the null distributions were still lacking, and these studies relied on empirical distributions generated from aligning unrelated sequences. In [29], it is shown that the similarity score is proportional to the logarithm of the length of the sequences being compared, and the standard deviation is independent of the sequence length. The significance of an alignment was determined from the number of standard deviations over mean of the alignment score. These studies [30] also highlight that the statistical properties [31], such as nucleotide frequencies or codon usage, of the sequences affect the distribution of the alignment scores. Generating a null distribution from an incorrect model could lead to

an alignment of unrelated sequences being dubiously declared significant. Several methods are available to generate random sequences preserving these statistical properties [32, 33].

Erdos and Renyi [34] presented results for the length of the longest headrun in a the first n tosses of a biased coin. The length of the longest headrun in coin tosses is equivalent to the number of matches between two DNA sequences when shifts in the starting and ending positions of the sequences are not allowed, with the probability of head equal to the probability of match between letters of the DNA alphabet. In [35], this is generalized to matches between DNA sequences, while allowing shifts. These results indicate that allowing shifts doubles the length of the longest headrun. Results for the longest headrun allowing for up to k mismatches and sequences generated from a Markov chain are also considered. In [36, 37] the distribution of the longest matches is shown to a have an extreme value distribution with mean that is proportional to the logarithm of the sequences lengths and variance independent of sequence length. Here, when considering only matches, the asymptotic extreme value distribution is shown by considering a maximum of geometric distributions, and when mismatches are allowed, it is shown by considering a maximum of negative binomial distributions. An alternate approach is a Poisson approximation for the distribution of the longest match [38].

An crucial aspect of in aligning nucleic acid and protein sequences is using the appropriate score function. For example, PAM and Dayoff matrices is used for protein sequences [], and xxx is used for DNA sequences []. The score function used alters the score of the aligned sequences and thus the alignment score distribution of unaligned sequences. However, the approach based on the length of the headruns does not consider the score function used for an alignment. In [], it is shown that the score distribution of aligning unrelated sequences for any score function (that has at least one positive score and the expected score is negative) has the form of an extreme value distribution, and and explicit formulas that its parameters are provided.

4.3 Fragment Identification problem

Let Σ be an alphabet. A string X is a sequence of letters $a_0 a_1 \dots a_{n-1}$, where $a_i \in \Sigma$; $|X|$ denotes the length of the string X; and $X[i \dots j] = a_i \dots a_{j-1}$ is a substring of X.

The reference string T is generated from the DNA alphabet $\Sigma = \{A, T, G, C\}$, with $|T| = n$. A SMURF-seq read S is generated by concatenating substrings (called fragments) of T, with no information

available *a priori* about the number, length, orientation (forward or reverse-complement), and the position on T of these fragments. Further, S contains sequencing errors with a rate ρ. Let $|S| = m$ and $m \ll n$.

A fragment set P is an set of start locations of fragments on S. $P \subset \{0 \dots m-1\}$ and $|P| = k$, with the rule that 0 is in P always. By convention we consider the set P to be ordered such that if $i < j$ then $P_i < P_j$. For a fragment set P, $\sum_{i=1}^{k} P_{i+1} - P_i = m$ and we say that the i^{th} of S is the substring $S[P_i \dots P_{i+1}]$, with $P_{k+1} = m$.

For a given T and S, the fragment identification problem is to determine the elements of the fragment set P such that it corresponds to the start locations of fragments contained in S.

4.4 Approach to the fragment identification problem

We approach the fragment identification problem by defining a score function as follows: For a given fragment set P, we define the score of aligning S to T as:

$$score_T(S, P) = \sum_{i=1}^{k} \max\{score(T[u \dots v], S[P_i \dots P_{i+1}]) : 0 \leq u < v \leq n\}.$$

This allows us to consider the fragment identification problem as two inter-related problems: (1) Determining k, the size of the fragment set, and (2) given k, determining the elements of P such that $score_T(S, P)$ is maximized.

By the score function defined above, to determine the elements of the fragment set P, requires the knowledge of the number of fragments k and this is not known *a priori*. Further, the k that maximizes the score function would almost never correspond to the optimal fragment set. As an example, taking $k = m - 1$ which corresponds to taking each base as a fragment would maximize the score, however, this is a non-sensical alignment.

We propose to estimate the number of fragments k by aligning a read to the reference genome with different k. For each of these fragmentations, we determine the p-value by comparing the alignment score with the null distribution generated from aligning reads generated at random to a reference genome generated at random. Finally, we choose the fragmentation with lowest p-value as the optimal fragmentation.

The fragment identification problem differs from the alignment problems described in section 4.2 in a

crucial manner. For the fragment identification problem we have the reference genome, and it is assumed that the reads always arise from this genome; the score distribution of sequences generated at random is used to determine the optimal number of fragments on a SMURF-seq read. Whereas in the context of local alignment the score distribution of aligning random reads are used to determine a "meaningful" alignment by comparing the alignment score of sequences with the random null distribution.

4.5 Score distribution under a random model

Calculation of p-value for aligning a SMURF-seq read with a given fragmentation requires the null distribution of aligning reads generated at random with the same with the same fragmentation. The problem of finding the null distribution is defined as: consider strings T and S are generated by drawing letters independently from the same distribution from an alphabet $a \in \Sigma$ with probability p_a such that $\sum_{a \in \Sigma} p_a = 1$. For a given fragment set P containing k elements, we need to determine the distribution of $score_T(S, P)$. We use the following score function to obtain the distribution of $score_T(S, k)$:

$$score(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \\ -\infty & \text{otherwise.} \end{cases}$$

To determine the distribution of $score_T(S, P)$, we first consider the score distribution when $k = 1$, i.e. the entire read aligns as one fragment. Then, we consider the score distribution when $k > 1$ as the sum of $k = 1$ distributions. We also empirically show that the form of the null distribution when using a generalized scoring function is similar to the distribution obtained with score function defined above.

4.5.1 Score distribution of one fragment

The score distribution of $score_T(S, 1)$ has similarities to the score distribution of local alignment [] and profile alignment [], but also differs from these. The distribution of $score_T(S, 1)$ differs from the local alignment as we require an end-to-end alignment of S to a substring of T, and also differs from the profile score distribution since the letters of S are generated at random. Based on these results, the distribution of

$score_T(S, 1)$ is likely to follow an extreme value distribution.

Let X_j denote the score of aligning S with $T[j \ldots j + m - 1]$, then

$$X_j = \sum_{i=0}^{m-1} score(S[i], T[j + i]), j = 0, \ldots, n - m + 1.$$

Since the letters of T and S are iid, we have

$$X_j \sim binom(m, p)$$

where $p = \sum_{a=\Sigma} p_a^2$. For a large enough m, X_j can be approximated by a normal distribution as

$$X_j \sim N(mp, mp(1 - p)).$$

$score_T(S, 1)$ is the maximum score over all positions in T,

$$score_T(S, 1) = \max_{0 \leq j \leq n-m+1} X_j.$$

$score_T(S, 1)$ is a maximum of normal distributions, which follows an extreme value distribution (EVD) [39]. Here, we have a dependence between X_j and X_k for $|j - k| < m$.

We verified score distribution of $score_T(S, 1)$ by generating a random genome of length 50kb from the DNA alphabet with equal probabilities, and reads of length 40 bp and 100 bp. For each read length, we determined the score distribution by aligning 10,000 reads generated at random (Fig. 4.2a, b). The parameters for the EVD was estimated using the method of moments. Further, the score distribution for increasing read lengths shows an increasing trend in the mean and standard deviation of the distribution (Fig. 4.2c, d).

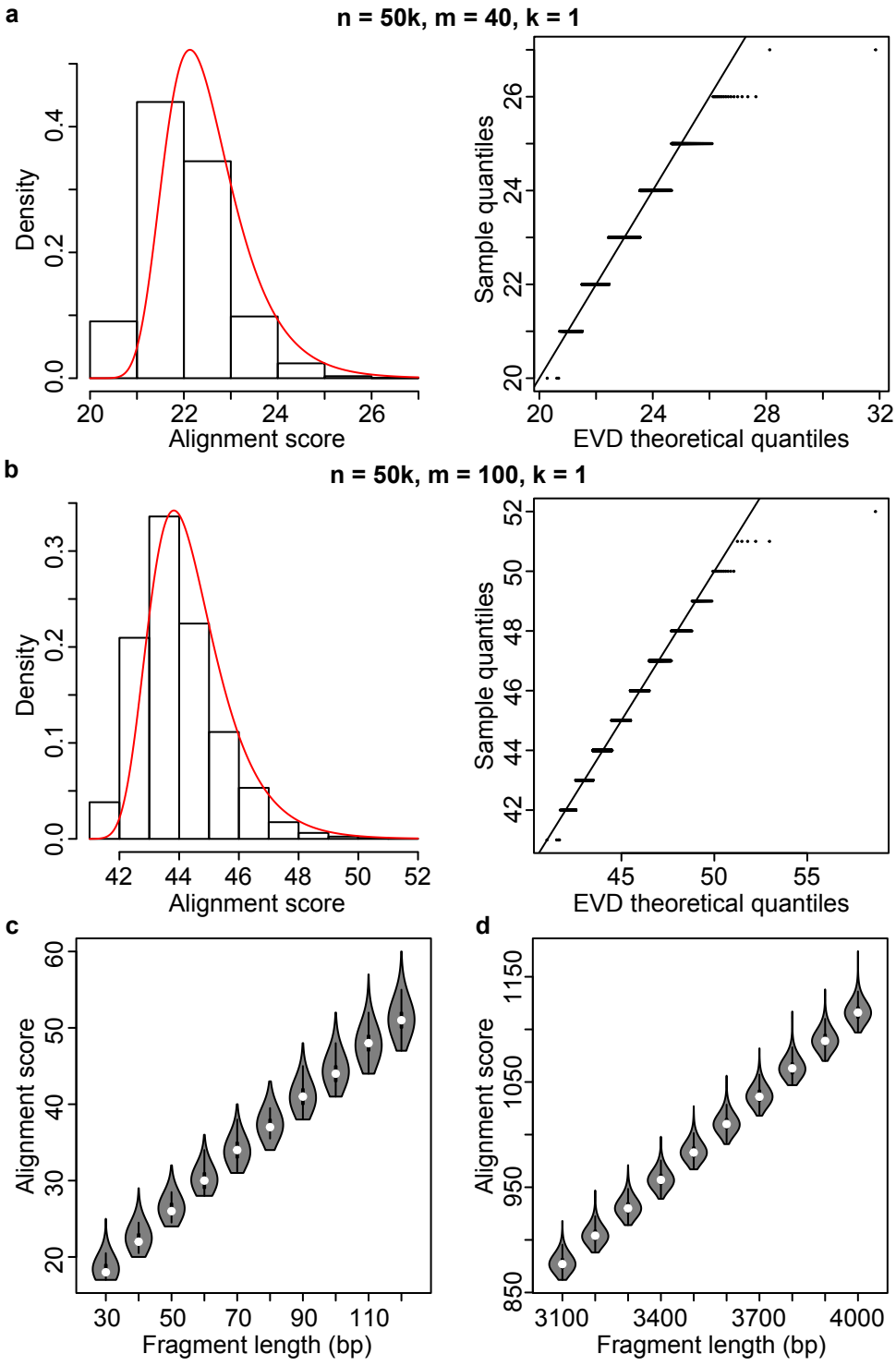

Figure 4.2: Empirical score distribution of $score_T(S, 1)$. (a - b) Empirical distribution for $m = 40$ and $m = 100$ with a fitted EVD using the method of moments estimator. Q-Q plot comparing the theoretical and empirical distributions are shown. (c) Empirical score distribution for m corresponding to shorter fragments. (d) Empirical score distribution form m correspomding to longer fragents.

4.5.2 Score distribution for a given fragment set

The distribution of $score_T(S, k)$ for $k > 1$ and a given P is the sum of k independent distributions of $score_T(S, 1)$, i.e the distribution of $score_T(S, k)$ is the sum of k independent extreme value distributions

$$score_T(S, k) = \sum_{i=1}^{k} score_T(S[P_i \dots P_{i+1}], 1).$$

The independence of the distributions for each fragment is justified because it is required that $n >> m$, and the probability of two fragments to aligning to overlapping location on T is extremely small.

The distribution of the sum and linear combination of extreme value distributions has been studied [40, 41, 42, 43]. In [42] the exact distribution of two independent Gumbel distributions is given and in [43] the exact distribution of the linear combination of Gumbel distributions is given. However, these distributions do not follow a "standard" distribution.

Since each the distribution of score of each fragment is independent, when the fragments are of equal lengths, the distribution of $score_T(S, k)$ is a sum of i.i.d. random variables. Thus, we can apply the central limit theorem to approximate the score distribution to a normal distribution as $k \to \infty$. To test the convergence to $score_T(S, k)$ to normal, we compared the score distribution to the normal distribution for k that are typical for a SMURF-seq read (Fig. 4.3). The fragments lengths for all the comparisons were kept constant at 40 bp, and the parameters for the normal distribution was determined using the method of moments estimator.

In aligning a SMURF-seq read, we cannot expect the fragment lengths to be equal. The distribution of $score_T(S, k)$, when the fragment lengths differ, is a sum of independent, but not identical, random variables. We empirically verified that this distribution converges to normal (Fig. 4.4). For each k, the fragment lengths were generated from at random from a geometric distribution.

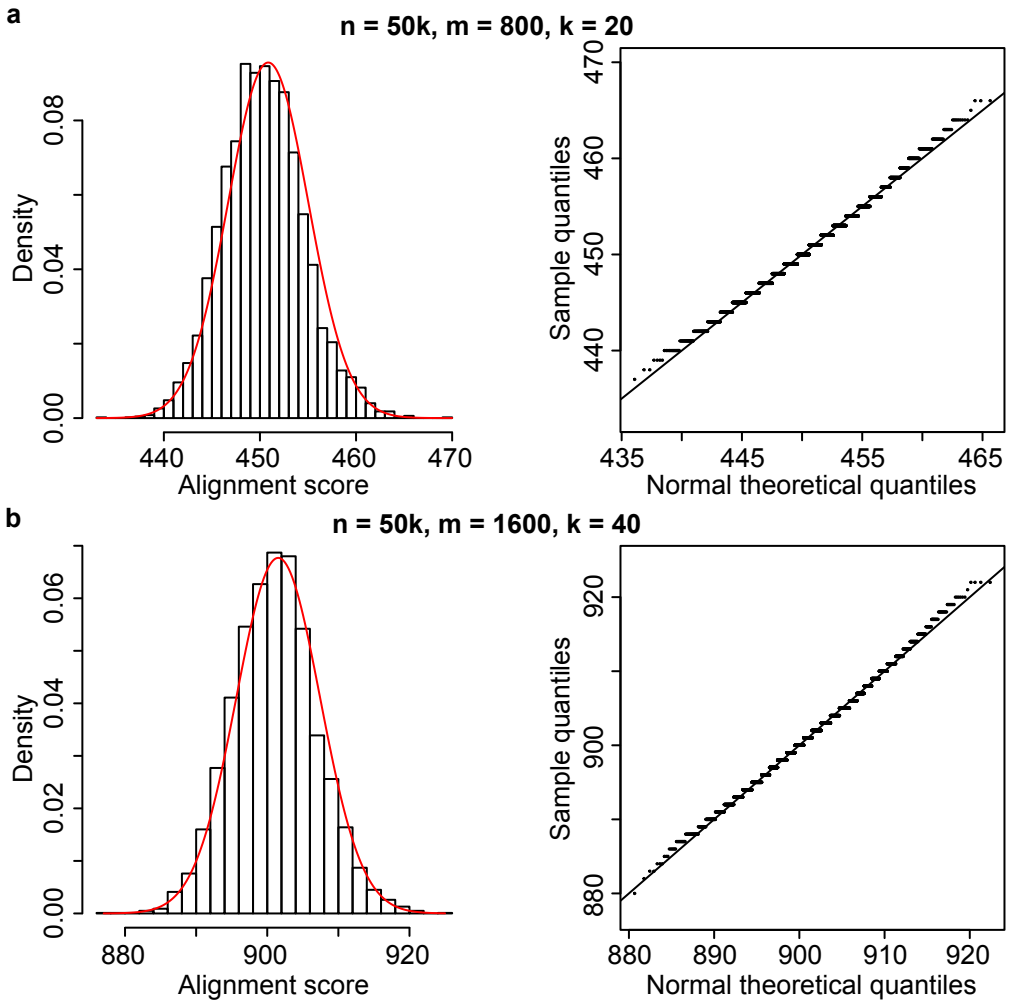

Figure 4.3: Empirical score distribution of $score_T(S, k)$ with a fitted normal using the method of moments estimator. All fragments are 40 bp. Q-Q plot comparing the theoretical and empirical distributions are shown. (a) $m = 800, k = 20$. (b) $m = 1600, k = 40$.

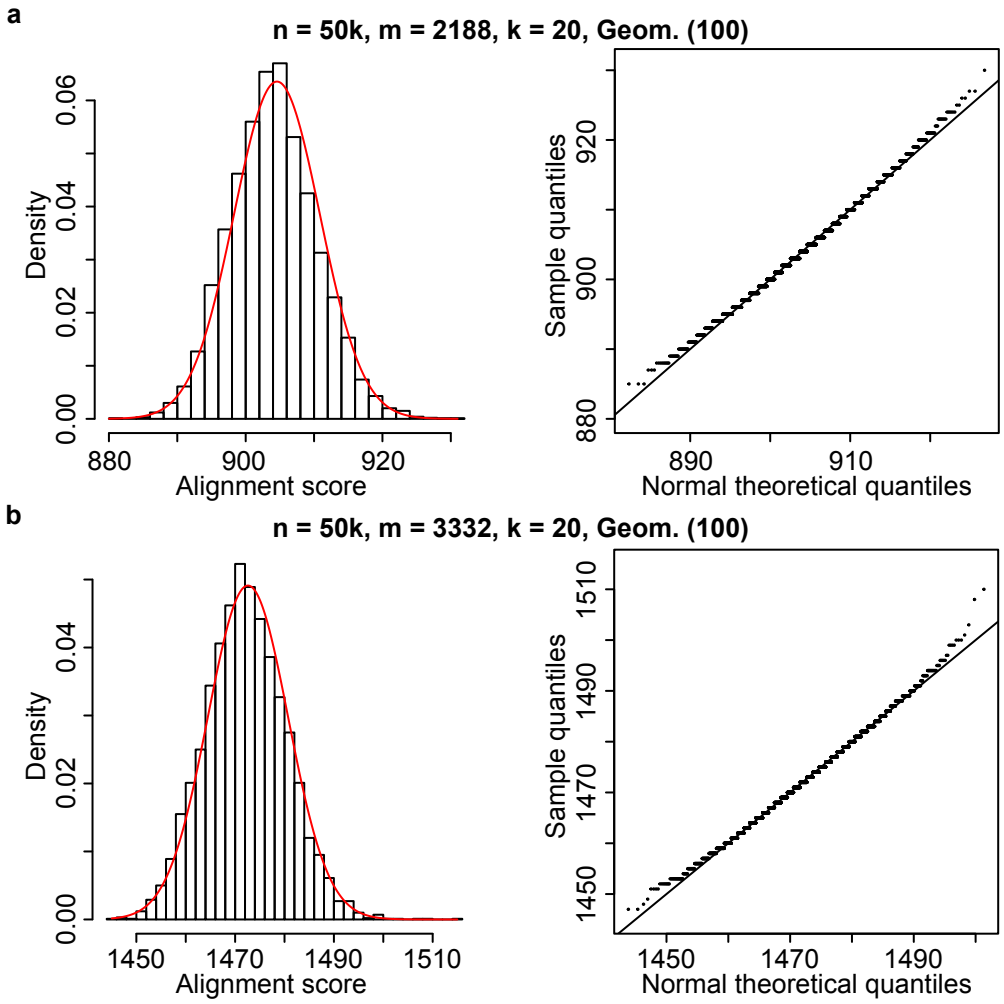

Figure 4.4: Empirical score distribution of $score_T(S, k)$ with a fitted normal using the method of moments estimator. The fragment lengths are genrated from a geometric distribution with mean 100. Q-Q plot comparing the theoretical and empirical distributions are shown. (a) $m = 2188, k = 20$. (b) $m = 3332, k = 40$.

4.6 Identifying optimal fragment boundaries

Having a score function for SMURF-seq reads enables a statistical approach to estimate the number of fragments on a read. The proposed method depends on an algorithm that can identify the optimal fragment boundaries, given the number of fragments, such that the sum of the alignment score of each fragment is maximized.

4.7 Fragment boundary identification under exact matching

We first examine the fragment identification problem assuming the score function requires exact matching

$$score(a,b) = \begin{cases} 1 & \text{if } a = b \\ -\infty & \text{otherwise.} \end{cases}$$

The fragment identification problem then becomes an exact matching problem where the goal is to minimize the number of fragments such that $score_T(P,S)$ is maximized.

A simple linear time solution to this problem can be obtained as follows. First, we assume some data structure for T has been constructed in linear time and allows for longest prefix matches to be computed in time proportional to the length of the query string. The data structure could be a suffix tree [44], or a more space efficient and a modern structure like an FM-index [45]. The principle of the algorithm can be seen by starting at the beginning of S, and identifying the longest prefix match of S in T. Then retain j as the first position of where this longest prefix matches in T, and denote the first mismatching position on S as i. Repeat the procedure solving the subproblem of fragment identification for $S[i \ldots m]$. Repeating these steps, the algorithm iteratively solves the longest prefix match problems, retaining as P_{i+1} the position of mismatch that terminates matching during iteration i. The following pseudocode describes the procedure.

Algorithm 1 ExactFragmentMatching(T, S):

```
1: i ← 0
2: while i < m do
3:       P ← P ∪ {i}
4:       i ← longest-match-length(S, i, T)
5: return P
```

Proof: Consider an optimal solution to this problem, where the identified fragment set P_{opt} has minimal size. To prove the optimality of our algorithm we need to show that it finds the same number of fragments as the optimal solution, i.e. $|P| = |P_{\text{opt}}|$.

The first iteration of the greedy algorithm will find the longest prefix match. If the optimal solution has its first fragmentation ending before P_1, i.e. $P_{\text{opt1}} < P_1$. Then the longest match starting at P_{opt1} will end at or before P_2, the end of the second fragment found by the greedy algorithm. If it ends at P_2 then

the greedy algorithm has the same number of fragments as the optimal solution so far. And it cannot end before P_2, because then the optimal solution will have more fragments than found by the greedy algorithm. Moreover, we cannot have $P_{\text{opt1}} > P_1$ as this would imply a longer prefix than found by the longest prefix match exists. With this reasoning we can say that this greedy approach will find just as little fragments as the optimal solution.

4.8 Fragment boundary identification allowing mismatches

Here, we examine the fragment identification problem assuming the score function that allows matches and mismatches (but not indels)

$$score(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \\ -\infty & \text{otherwise.} \end{cases}$$

For a fixed number of mismatches, if we are attempting to determine if some fragment set attains that number of mismatches, the optimal number of fragments might require fewer mismatches earlier in S and also shorter fragments, so that longer fragments are possible later, requiring more mismatches. Intuitively this means we cannot rely on maximality of prefix matching. Here we explain a dynamic programming algorithm to solve the fragment identification problem with a bounded number of mismatches summed over all the fragments.

Let M denote a table with $k+1$ rows and $d+1$ columns, where k is the maximum number of fragments and d is the maximum number of mismatches allowed. We will use entry $M(i, j)$ to represent the longest prefix of S that matches T with at most i fragments and j mismatches summed over all fragments. The following algorithm computes the entries of M using a subroutine for identifying approximate matches to a pattern within a text.

Algorithm 2 FragMatchingWithMismatches(T, S, k, d)

$M(0, i) \leftarrow 0$ for all $0 \leq i \leq d$
for $i \leftarrow 1$ **to** k **do**
    for $j \leftarrow 0$ **to** d **do**
        for $l \leftarrow 0$ **to** j **do**
            $M(i, j) \leftarrow \max\{p | dist(S[M(i-1, l) \ldots p], T) = j - l)\}$

After computing all the entries in M, if $M(i, j) = m$ for any i and any j then there exists a set of k fragments in S having at most d total mismatches with respect to T.

Time and space complexity: $O(f(m, n) + kd^2 g(m, n))$, where $f(m, n)$ is the time required for pre-possessing S and T in order to allow approximate prefix matching queries, and $g(m, n)$ is the time required for those queries assuming pre-processing had been done. The algorithm uses $O(h(m, n) + kd)$ space, where $h(m, n)$ is the space required for a data structure that allows longest prefix match with a specified number of mismatches.

4.9 Fragment boundary identification allowing mismatches and indels

Let M denote a table with $m + 1$ rows, $n + 1$ columns and $k + 1$ dimensions, where k is the maximum number of fragments ($1 \leq k \leq m$). $\max_{0 \leq j \leq n} M(i, j, l)$ represents the best fragmentation of $S[1 \ldots i]$ with l fragments. The entries of M are computed as follows:

Algorithm 3 FragBoundaryIdentification(T, S, k)

$M(i, j, 0) \leftarrow -\infty$ for all $0 \leq i \leq m, 0 \leq j \leq n$
$M(0, j, 1) \leftarrow 0$ for all $0 \leq j \leq n$
$M(i, 0, 1) \leftarrow M(i-1, 0, 1) + score(S[i], \_)$ for all $0 \leq i \leq m$
$M(l-1, j, l) \leftarrow -\infty$ for all $2 \leq l \leq k, 0 \leq j \leq n$
$M(i, 0, l) \leftarrow -\infty$ for all $2 \leq l \leq k, l \leq i \leq m$
for $l \leftarrow 1$ **to** k **do**
    for $i \leftarrow l$ **to** m **do**
        for $j \leftarrow 1$ **to** n **do**

$$M(i, j, l) \leftarrow \max \begin{cases} M(i-1, j-1, l) + score(S[i], T[j]) \\ M(i-1, j, l) + score(S[i], \_) \\ M(i, j-1, l) + score(\_, T[j]) \\ \max_{0 \leq h \leq n} M(i-1, h, l-1) + score(S[i], T[j]). \end{cases}$$

Time and space complexity: Each entry of M is computed in constant time by storing the value of

$\max_{0 \leq j \leq n} M(i-1, j, l-1)$ for every row of M in a separate array. The algorithm runs in $O(knm)$ time and uses $O(knm + km)$ space, where the additional $O(km)$ is used to store the values of $\max_{0 \leq j \leq n} M(i-1, j, l-1)$.

The optimal alignment and fragment boundaries are determined from the usual traceback procedure starting from $\max_{1 \leq j \leq n} M(m, j, k)$ and ending in $M_{1 \leq j \leq n}(1, j, 1)$, with the exception of storing if a new fragment maximized the score at a cell.

Intuitively, this algorithm is similar to the local alignment algorithm but instead of picking an empty alignment when the score of an extension is negative, this algorithm starts a new fragment when the score of extending a fragment is less than score of starting a new fragment. In terms of an alignment graph, each node has a zero-weight incoming edge from the node corresponding to $\max_{0 \leq j \leq n} M(i-1, j, l-1)$, in addition to the weighted match/mismatch and indel edges (Fig. 4.5).

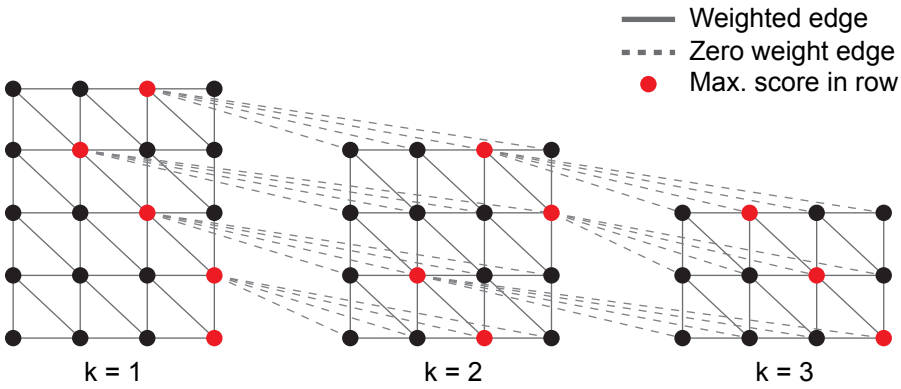

Figure 4.5: Alignment graph for fragment boundary identification algorithm with an arbitrary score function. The direction of arrows are omitted for clarity. The horizontal edges are directed from left to right and all other edges are directed from top to bottom.

4.10 Estimating the optimal fragment set

The score distribution of aligning a random read S_{rand} to a random genome T_{rand} can be used to estimate the optimal k for aligning a SMURF-seq read S_{SMURF} to a reference genome T_{ref}. A procedure to do this is as follows: For a SMURF-seq read, find the best alignment score k_{score} and the fragment set k_{P} for all k from 1 to m using the algorithm given in section **??**. Also, for each k using the fragment set k_{P}, determine

the p-value of finding an alignment with score greater than k_{score} from the random model. The optimal k for a read is the one with the lowest p-value.

Algorithm 4 OptimalK (T, S)

```
k_opt ← 1
Pr_opt ← 1
for k ← 1 to m − 1 do
    k_score, k_P ← FragMatch(T_ref, S_SMURF, k)
    k_Pr ← Pr(score_{T_rand}(S_rand, k_P) > k_score)
    if k_Pr < Pr_opt then
        Pr_opt ← k_Pr
        k_opt ← k
return k_opt
```

Say the optimal k for a read is k_{opt}, as k goes from 1 to $k_{\text{opt}} - 1$, k fragments in the read are likely mapped to its true location, whereas the rest of the bases in the read are going to be mapped to random locations genome adjacent to the mapped fragments. So as k goes toward $k_{\text{opt}} - 1$, the number of bases that are mapped to its true location will increase and the bases mapped to random locations will decrease, but there will be bases mapped to random genome locations. Therefore with each iteration the p-value would decrease. At $k = k_{\text{opt}}$, all the fragments are mapped to its true locations and there would be no bases mapped to random locations, so the p-value should be at its lowest. For $k > k_{\text{opt}}$, there are no random base mappings but the true mappings will be split into shorter fragments. As k increases beyond k_{opt}, the p-value would increase since as the fragments get shorter, the more likely they are to align to a random location on the reference.

4.11 Results

To evaluate the profromace of your method, we simulated a refrence genome, generated simulated SMURF-seq reads from the genome, and introduced sequencing errors to the reference genome. The number of fragments and the starting location of each fragment on the smulated read is known. These reads are then mapped back to the reference genome for values several values of k using algorithm **??**, yielding the fragment set that maximizes the aligment score for each k. These fragment sets were then used to generate the null distribution by simulationg a random reference genome with the same base probabilities as the

reference genome, and aliging random reads with fragment start locations based on the fragment set. The p-value for each fragmentation was determined using an EVD for $k = 1$ and normal distibutions for $k > 1$ with parametes estimated using the method of moments from the simulated reads. The fragmenation with the smallst p-value was considerd as the optimal framgnation. These predicted fragmentations were then compared with the ground truth to determine the accuracy.

4.11.1 Reads with mismatches

4.11.2 Fast computation of p-values

4.11.3 Aligning with a general score function

Bibliography

[1] Philipp Euskirchen, Franck Bielle, Karim Labreche, Wigard P Kloosterman, Shai Rosenberg, Mailys Daniau, Charlotte Schmitt, Julien Masliah-Planchon, Franck Bourdeaut, Caroline Dehais, et al. Same-day genomic and epigenomic diagnosis of brain tumors using real-time nanopore sequencing. *Acta Neuropathologica*, 134(5):691–703, 2017.

[2] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature Biotechnology*, 2018.

[3] John R Tyson, Nigel J O'Neil, Miten Jain, Hugh E Olsen, Philip Hieter, and Terrance P Snutch. Minion-based long-read sequencing and assembly extends the caenorhabditis elegans reference genome. *Genome Research*, 28(2):266–274, 2018.

[4] M Muthukumar. Theory of capture rate in polymer translocation. *The Journal of Chemical Physics*, 132(19):05B605, 2010.

[5] Meni Wanunu, Jason Sutin, Ben McNally, Andrew Chow, and Amit Meller. DNA translocation governed by interactions with solid-state nanopores. *Biophysical Journal*, 95(10):4716–4725, 2008.

[6] Achilles Dugaiczyk, Herbert W Boyer, and Howard M Goodman. Ligation of ecori endonuclease-generated dna fragments into linear and circular structures. *Journal of molecular biology*, 96(1):171–184, 1975.

[7] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads, 2018.

[8] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[9] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

[10] Szymon M Kiełbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin C Frith. Adaptive seeds tame genomic sequence comparison. *Genome research*, 21(3):487–493, 2011.

[11] Ivan Sović, Mile Šikić, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjan Nagarajan. Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nature Communications*, 7:11307, 2016.

[12] Mark J Chaisson and Glenn Tesler. Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory. *BMC bioinformatics*, 13(1):238, 2012.

[13] Bo Liu, Dengfeng Guan, Mingxiang Teng, and Yadong Wang. rHAT: fast alignment of noisy long reads with regional hashing. *Bioinformatics*, 32(11):1625–1631, 2015.

[14] Bo Liu, Yan Gao, and Yadong Wang. LAMSA: fast split read alignment with long approximate matches. *Bioinformatics*, 33(2):192–201, 2017.

[15] Timour Baslan, Jude Kendall, Linda Rodgers, Hilary Cox, Mike Riggs, Asya Stepansky, Jennifer Troge, Kandasamy Ravi, Diane Esposito, B Lakshmi, et al. Genome-wide copy number analysis of single cells. *Nature Protocols*, 7(6):1024, 2012.

[16] Jude Kendall and Alexander Krasnitz. *Computational Methods for DNA Copy-Number Analysis of Tumors*, pages 243–259. Springer New York, New York, NY, 2014.

[17] Angel E Dago, Asya Stepansky, Anders Carlsson, Madelyn Luttgen, Jude Kendall, Timour Baslan, Anand Kolatkar, Michael Wigler, Kelly Bethel, Mitchell E Gross, et al. Rapid phenotypic and genomic change in response to therapeutic pressure in prostate cancer inferred by high content analysis of single circulating tumor cells. *PloS ONE*, 9(8):e101777, 2014.

[18] Jesse L Berry, Liya Xu, A Linn Murphree, Subramanian Krishnan, Kevin Stachelek, Emily Zolfaghari, Kathleen McGovern, Thomas C Lee, Anders Carlsson, Peter Kuhn, et al. Potential of aqueous humor as a surrogate tumor biopsy for retinoblastoma. *JAMA ophthalmology*, 135(11):1221–1230, 2017.

[19] Geoff Macintyre, Teodora E Goranova, Dilrini De Silva, Darren Ennis, Anna M Piskorz, Matthew Eldridge, Daoud Sie, Liz-Anne Lewsley, Aishah Hanif, Cheryl Wilson, et al. Copy number signatures and mutational processes in ovarian carcinoma. *Nature Genetics*, 50(9):1262, 2018.

[20] Tanjina Kader, David L Goode, Stephen Q Wong, Jacquie Connaughton, Simone M Rowley, Lisa Devereux, David Byrne, Stephen B Fox, Gisela Mir Arnau, Richard W Tothill, et al. Copy number analysis by low coverage whole genome sequencing using ultra low-input DNA from formalin-fixed paraffin embedded tumor tissue. *Genome Medicine*, 8(1):121, 2016.

[21] Timour Baslan, Jude Kendall, Brian Ward, Hilary Cox, Anthony Leotta, Linda Rodgers, Michael Riggs, Sean D'Italia, Guoli Sun, Mao Yong, et al. Optimizing sparse sequencing of single cells for highly multiplex copy number profiling. *Genome Research*, 25(5):714–724, 2015.

[22] Nicholas Navin, Jude Kendall, Jennifer Troge, Peter Andrews, Linda Rodgers, Jeanne McIndoo, Kerry Cook, Asya Stepansky, Dan Levy, Diane Esposito, et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90, 2011.

[23] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[24] William R Pearson and David J Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[25] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

[26] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome biology*, 10(3):R25, 2009.

[27] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[28] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[29] Temple F Smith, Michael S Waterman, and Christian Burks. The statistical distribution of nucleic acid similarities. *Nucleic Acids Research*, 13(2):645–656, 1985.

[30] David J. Lipman, W. John Wilbur, Temple F. Smith, and Michael S. Waterman. On the statistical significance of nucleic add similarities. 1984.

[31] TF Smith, MS Waterman, and JR Sadler. Statistical characterization of nucleic acid sequence functional domains. *Nucleic acids research*, 11(7):2205–2220, 1983.

[32] Walter M Fitch. Random sequences. *Journal of Molecular Biology*, 163(2):171–176, 1983.

[33] Stephen F Altschul and Bruce W Erickson. Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Molecular biology and evolution*, 2(6):526–538, 1985.

[34] Paul Erdös and Pál Révész. On the length of the longest head-run. *Topics in information theory*, 16:219–228, 1975.

[35] Richard Arratia and Michael S Waterman. An erdös-rényi law with shifts. *Advances in mathematics*, 55(1):13–23, 1985.

[36] Richard Arratia, Louis Gordon, Michael Waterman, et al. An extreme value theory for sequence matching. *The annals of statistics*, 14(3):971–993, 1986.

[37] Louis Gordon, Mark F Schilling, and Michael S Waterman. An extreme value theory for long head runs. *Probability Theory and Related Fields*, 72(2):279–287, 1986.

[38] Richard Arratia, Michael S Waterman, et al. The erdös-rényi strong law for pattern matching with a given proportion of mismatches. *The Annals of Probability*, 17(3):1152–1169, 1989.

[39] Samuel Kotz and Saralees Nadarajah. *Extreme value distributions: theory and applications*. World Scientific, 2000.

[40] Coskun Cetinkaya, Vikram Kanodia, and Edward W Knightly. Scalable services via egress admission control. *IEEE Transactions on multimedia*, 3(1):69–81, 2001.

[41] Filipe J Marques, Carlos A Coelho, and Miguel De Carvalho. On the distribution of linear combinations of independent gumbel random variables. *Statistics and Computing*, 25(3):683–701, 2015.

[42] HA Loaiciga and RB Leipnik. Analysis of extreme hydrologic events with gumbel distributions: marginal and additive cases. *Stochastic Environmental Research and Risk Assessment*, 13(4):251–259, 1999.

[43] Saralees Nadarajah. Exact distribution of the linear combination of p gumbel random variables. *International Journal of Computer Mathematics*, 85(9):1355–1362, 2008.

[44] Edward M McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM (JACM)*, 23(2):262–272, 1976.

[45] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 390–398. IEEE, 2000.