Information-Theoretic Results for DNA-based Data Storage in the Shotgun Sequencing Channel with Erasures

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computational Natural Sciences
by Research

by

Hrishi Narayanan
2019113022
hrishi.narayanan@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
JUNE 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Information-Theoretic Results for DNA-based Data Storage in the Shotgun Sequencing Channel with Erasures" by Hrishi Narayanan, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____
Date

_____
Adviser: Prof. Prasad Krishnan

_____
Adviser: Prof. Nita Parekh

To all those who stood by me during my most challenging times.

Acknowledgments

I am deeply grateful to all those who have supported and encouraged me throughout the journey of completing this thesis. Without their contributions, whether big or small, this achievement would not have been possible.

First and foremost, I would like to express my heartfelt gratitude to my primary advisor, Prof. Nita Parekh, for her invaluable guidance, continuous support and encouragement. Her expertise and insights have been instrumental in shaping this thesis and enhancing its quality.

I extend my deepest gratitude and indebtedness to my advisor, Prof. Prasad Krishnan. His unwavering support, expert guidance, and genuine interest in my work have been critical in shaping this thesis and enhancing its quality. I am truly grateful for his dedication, constructive feedback, and mentorship, which have significantly contributed to my academic growth and the successful completion of this project.

My heartfelt appreciation goes to my family members, Achchan, Amma, Mottu Chettan and Kannunni, for their constant love, encouragement, and understanding throughout this journey. Their support has been my source of strength and motivation.

I would also like to acknowledge the contributions of my friends and peers, who have been there for me with their moral support, camaraderie, and occasional distractions that provided much-needed relief during challenging times.

I also extend special thanks to iHub Data Foundation, IIIT Hyderabad for extending the research fellowship that provided financial support during the course of this study. Their generosity and support have enabled me to focus on my research and bring this thesis to fruition.

Lastly, I express my gratitude to all those whose names may not be mentioned here but have contributed in any way to the completion of this thesis. Your support, encouragement, and belief in my abilities have been invaluable.

Abstract

In shotgun sequencing, the input string (typically, a long DNA sequence composed of nucleotide bases) is sequenced as multiple overlapping fragments of much shorter lengths (called *reads*). Modelling the shotgun sequencing pipeline as a communication channel for DNA data storage, the capacity of this channel was identified in a recent work, assuming that the reads themselves are noiseless substrings of the original sequence. Modern shotgun sequencers however also output quality scores for each base read, indicating the confidence in its identification. Bases with low quality scores can be considered to be erased. Motivated by this, we consider the *shotgun sequencing channel with erasures*, where each symbol in any read can be independently erased with some probability δ. We identify achievable rates for this channel, using a random code construction and a decoder that uses typicality-like arguments to merge the reads. To do this, we analyse the probability of error of the decoder and establish that the probability of error vanishes, as the length of the code goes to infinity, when the rate of the code is bounded based on the parameters of the channel. Our achievability result subsumes the achievability result obtained in the prior work for the shotgun sequencing channel (without erasures, i.e., with erasure probability $\delta = 0$) [1]. However, the case of non-zero erasure probability has never been considered in the literature before, and hence our achievability results are completely novel in this case. For given parameters of the problem, we give some numerical comparisons of our achievable rate with an 'interpolated' version of the achievable rate from prior work for the $\delta = 0$ case, and show that our result is a non-trivial improvement over such an interpolation.

Table of General Notation used in this thesis

Capital letters ($X, \mathcal{Y}, \Lambda, \Phi$ etc.)	Random quantities
Underline ($\underline{x}, \underline{y}, \underline{\omega}$ etc.)	Strings or tuples
$[a:b]$	Set of integers $a, a+1, ..., b$
$[b]$	Set of integers $[1:b]$
$\mathbb{I}_A$	Indicator random variable associated with event A
$\Pr(A)$	Probability of event A
$\overline{A}$	Complement of event A
$\Pr(A, B)$	Probability $\Pr(A \cap B)$, for two events A, B
$S^\star$	Set of finite length strings with symbols from set S

Table of Select Notations used in this thesis

R	Rate
C	Capacity of a channel
c	Coverage depth
n	Block length
L	Read length
$\bar{L}$	Normalised read length
K	Number of reads
δ	Erasure probability
$\mathsf{SSE}(\delta)$	Shotgun Sequencing Channel with Erasures (with probability δ)
CI	Set of candidate islands

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Promise of DNA as a storage medium

On December 29, 1959, in his talk, titled "*There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics*", at the annual American Physical Society meeting at Caltech, Richard P. Feynman shared his very strong belief that miniaturisation was the key to advancement across various fields. Feynman discussed in detail about potentially more robust synthetic chemistry through direct manipulation of individual atoms, the many advantages of miniaturisation, and the various challenges while working with objects in the nanoscale. [11]

Feynman's lecture did not garner significant popular impact at the time. However, following the development of the scanning tunnelling microscope in 1981 and the subsequent emergence of the field of nanotechnology in the 1990s, Feynman's lecture suddenly gained considerable academic attention, over four decades after it was originally presented. Since then, the "Plenty of Room" lecture has been cited as the motivation behind numerous works in nanotechnology, and has also been cited as an inspiration behind several other works in cutting-edge fields including quantum computing [12], synthetic biology and molecular machines [13, 14], optoelectronics [15] etc.

Among other topics, Feynman remarked on the size of the computers in the lecture, which back then occupied several rooms. He then contrasted this to the size of the brain, which is much smaller and is yet able to compute much more efficiently, and argued that there must be ways to improve upon the computational efficiency through miniaturisation. Similarly, he compared the highly-inefficient digital data storage systems of the day, to the storage of information in biological systems. He noted the tremendous amount of information stored in each cell of the body in the form of DNA and emphasised on the vast potential of building data storage systems inspired by biological systems. Over half a century later, in the 2000s, the idea of storing digital data in DNA molecules, started gaining significant momentum. This was accompanied by practical demonstration of such storage systems, bringing Feynman's vision of a data storage inspired by biology to fruition.

In the past decade, there has been considerable academic and industrial interest in studying and developing DNA as an archival storage medium. Mainly, the motivation for developing DNA-based

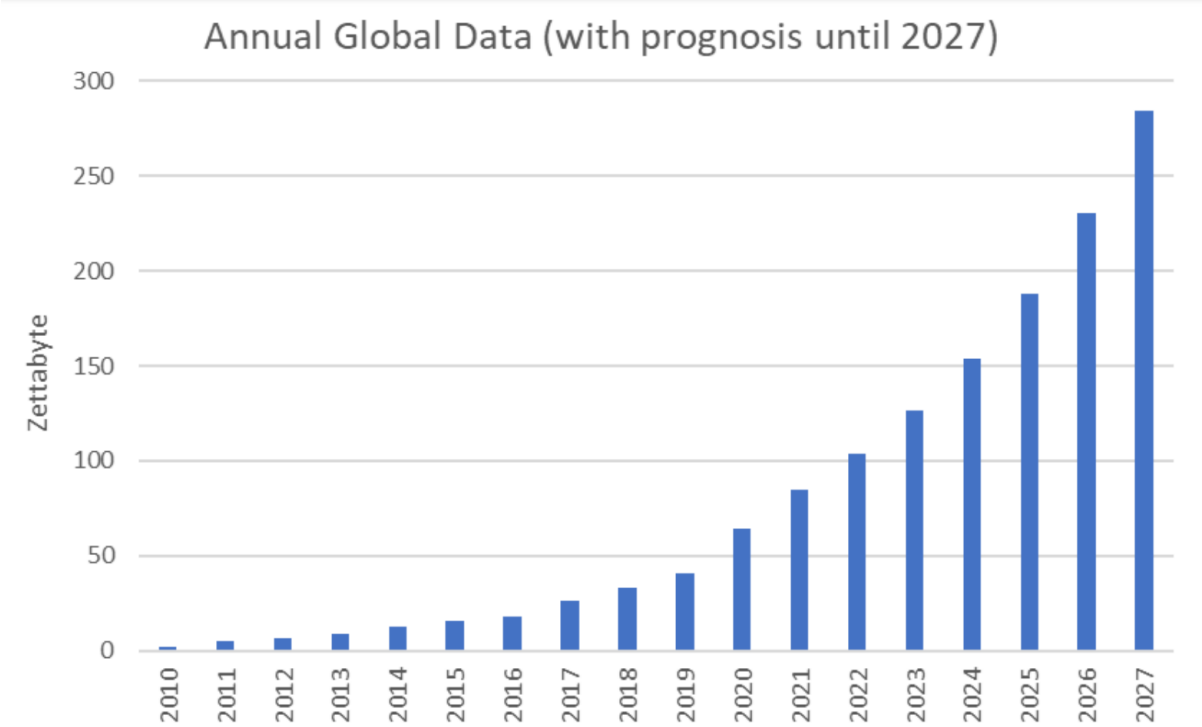

Figure 1.1 Volume of data generated by year (2010-2027) [2]

storage systems stems from two major practical concerns. Firstly, the amount of data generated by the world is increasing rapidly. There has been an exponential growth in the amount of data stored (see Figure 1.1), with total amount of stored data by 2027 projected to be 284 zettabyte [2]. Out of this, around 80% of the data is cold data or archival data, which need to be stored for long durations of time, but is infrequently accessed [2].

Secondly, traditional storage devices have certain disadvantages. Such storage devices, including hard drives, magnetic tapes, solid state drives systems are unstable and prone to data corruption. This is because such devices use the magnetic spin of electrons in the medium to store digital data. Writing of the data to the medium is done through polarisation of the electrons, while reading is done through measuring the magnetic dipole created by the electrons. Thus, factors such as exposure to environmental electromagnetic (EM) radiations, heat etc. can cause changes in the magnetic polarisation, leading to corruption of data, especially over a long period of time. HDD and magnetic tape, two of the most popular traditional candidates for archival storage, have a mean lifetime of only 15 and 20 years respectively [2]. In addition, such storage systems take up a large amount of physical space. For instance, the GitHub Arctic Code Vault required 3,500 feet of tape to store 21 TB of data [16].

In contrast, DNA molecules are very stable. While magnetic storage can last at most a couple of decades, synthetic DNA molecules are estimated to last several centuries to several millennia [17] [18], with some estimates going as high as 400,000 years [9]. Furthermore, the storage density of DNA as

a medium is significantly higher. In particular, the theoretical information density of synthetic DNA is 10^{18}B/mm^2 and nearly 455 billion GB of data per gram, which is 10^7 times and 10^6 times, respectively, more than that of magnetic tapes [18] [9]. Hence, due to its exceptional stability and high data density, DNA molecules show great promise as a medium for long term storage of digital data.

1.2 DNA as a Storage Medium

1.2.1 DNA Storage: Concept, Development, and Information-Theoretic Approaches

The conceptual model of a DNA storage systems, like any other storage pipeline, has distinct read and write steps, as represented in Figure 1.2. In the writing stage, the digital files in the binary format are first encoded into a nucleobase code (i.e., in A, T, G and C). The encoding into nucleobase includes error-correction encoding steps, if any. The nucleobase code so generated is used as a template for the synthesis of DNA strand. The DNA molecule so produced is then stored. In order to retrieve (read) the data, the DNA molecule is first sequenced and translated into nucleobase code. The nucleobase code is subsequently decoded to get binary strings corresponding to the stored data.

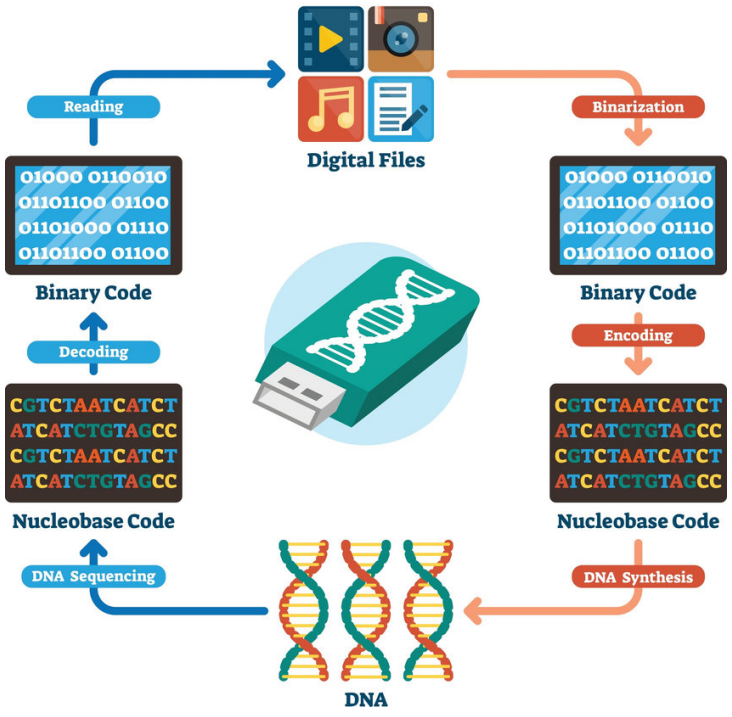

Figure 1.2 Diagrammatic representation of DNA-based storage process (reproduced from [3]).

The first practical demonstration of storing messages in DNA was done in 1988 [17]. Since then, researchers have developed sophisticated storage pipelines for DNA-based data storage, and attempted storing greater amounts of data within synthetic DNA molecules [4, 18]. The trend in the amount of

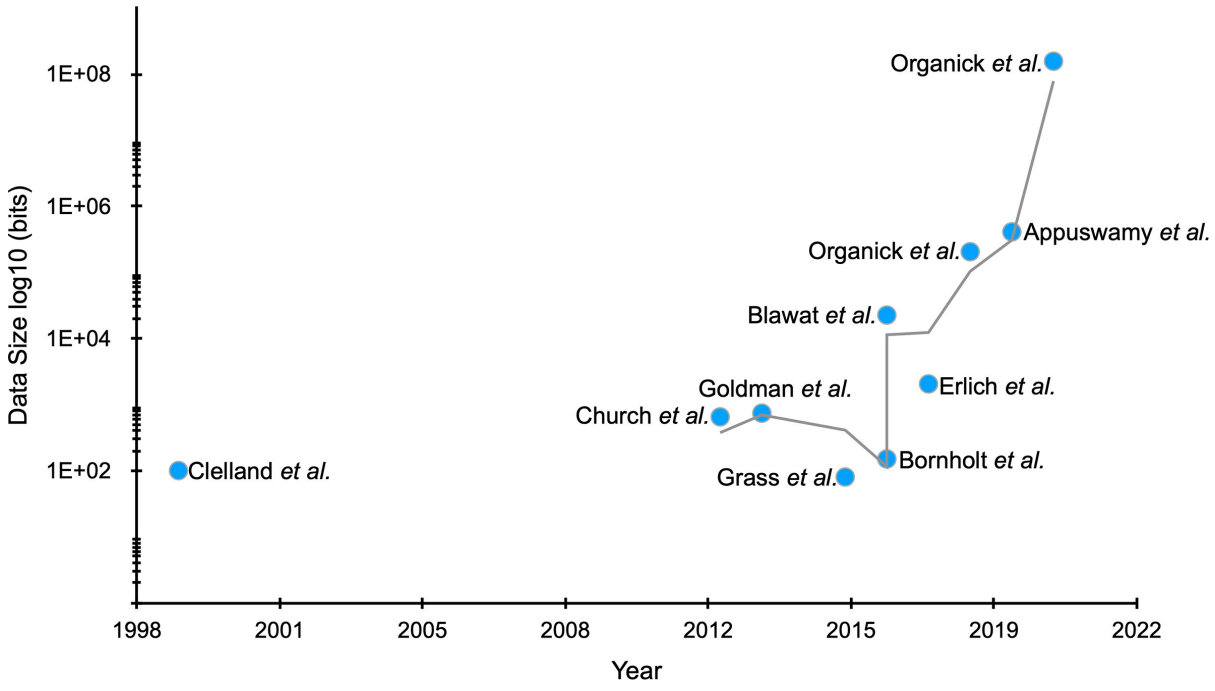

Figure 1.3 Trend in the amount of data stored in DNA (reproduced from [4]).

data stored in DNA molecules by various works across the past years is given in Figure 1.3. In 2019, Takahashi *et al.* constructed the first full end-to-end automated DNA storage device, outlining how various stages of the pipeline can be effectively automated [19].

Characterisation of the errors that occur within the DNA storage pipeline, both the sources of error and the error probability, is also necessary for further analysis and understanding of the storage process. Several such works have been conducted in the recent times. For instance, Heckel *et al.* compared the results obtained experimentally with the data from two other research groups and identified of the synthesis and sequencing steps as major sources of errors in the storage process. In addition to the errors during pipeline, errors may occur due to the decay of DNA molecules over large periods of time. Such errors were characterised by Grass *et al.* through in silica simulation [20]. The authors also performed in silica experiments to demonstrate that error-correction coding can be used to increase the lifetime of the stored data.

Based on the error characterisations, several sequencing and alignment algorithms have been developed for DNA storage. For instance, Bresler *et al.* presented an algorithm for optimal assembly of high throughput shotgun sequencing channel and established that a greedy algorithm is nearly optimal when length of repeats is approximately equal to the length of interleaving section [21]. Shomorony *et al.* presented an algorithm (NOT-SO-GREEDY algorithm) which constructs a sparse read-overlap graph, which is then used to solve the genome assembly. The work also establishes a performance guarantee for the algorithm, by showing that the genome assembly problem reduces into a Eulerian path problem when certain conditions are met [22].

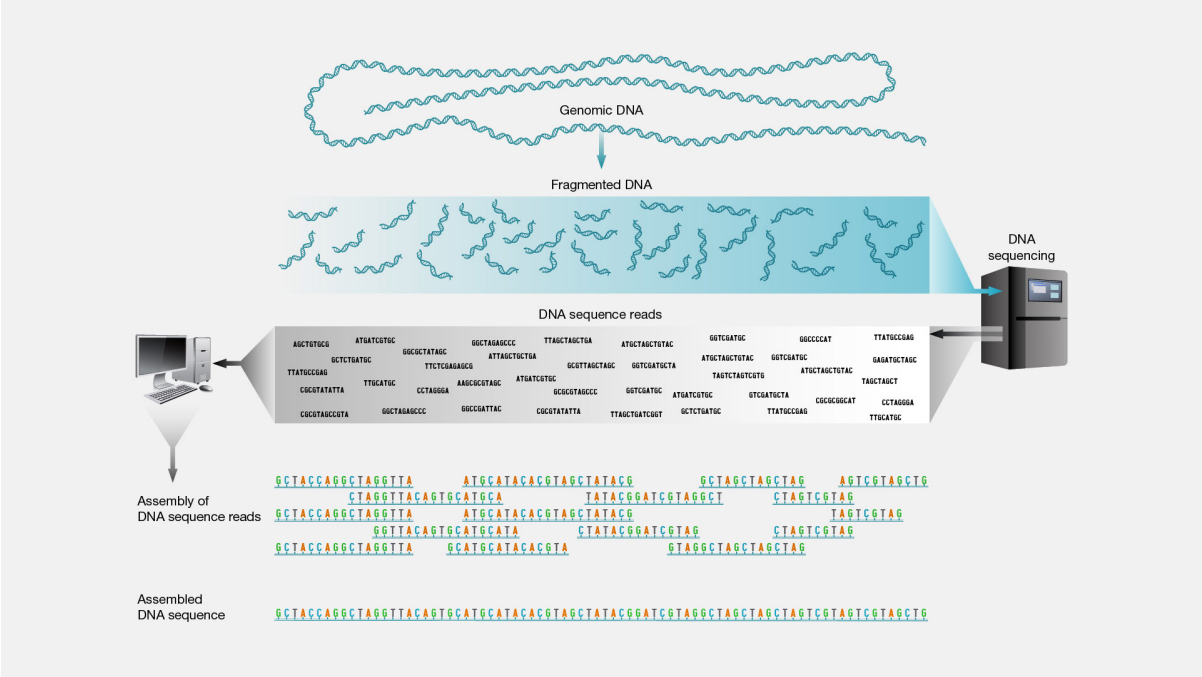

Figure 1.4 Diagrammatic representation of the high-throughput shotgun sequencing pipeline (reproduced from [5]).

To facilitate further analysis, the DNA storage pipeline has been abstracted as various communication channels by different works. This has motivated the development of practical error-correction code schemes [23–25]. Furthermore, it has motivated extensive information-theoretical analysis of the channel, including determining the fundamental limits of the channel [1, 10, 26–34].

1.2.2 Sequencing Techniques

The advancements, as well as the academic and industrial interest, in DNA-based storage systems have been fuelled by the developments in sequencing techniques which reduced costs and increased feasibility of such a technology. One such significant development is that of the *high-throughput shotgun sequencing* pipeline, as represented in Figure 1.4. In such a pipeline, multiple copies of the DNA molecule are first created, and then broken into fragments using restriction enzymes. The fragments are generally much shorter than the original molecule. These fragments are then sequenced, resulting in a collection of *reads*. The reads are then subsequently aligned, by mapping the overlaps between them, to reconstruct the original sequence.

Some of the most popular sequencing techniques include the chain-termination method (Sanger sequencing), the sequencing-by-synthesis (Illumina) and the nanopore sequencing (Oxford Nanopore) techniques. In the Sanger sequencing process, multiple copies of the DNA strand are created through chain-termination PCR, in the presence of fluorescently tagged nucleotides with terminator groups. The

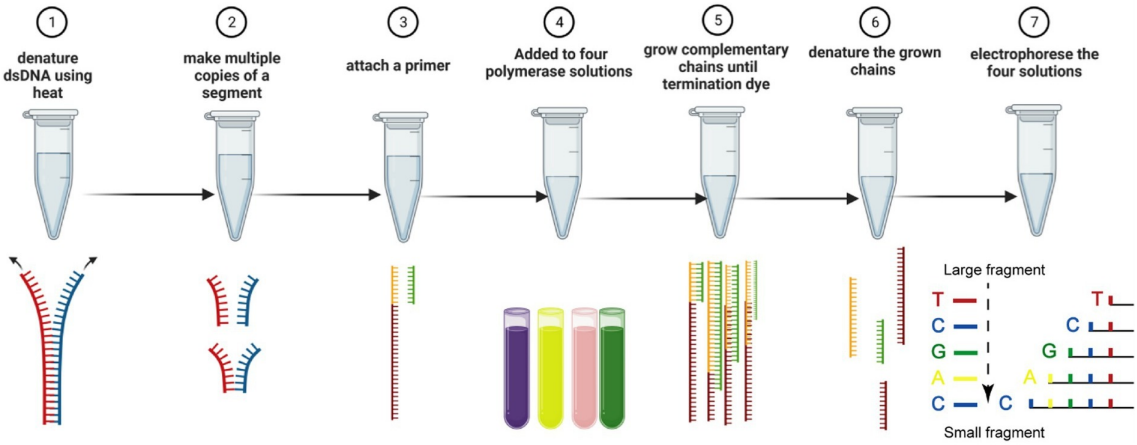

Figure 1.5 Diagrammatic representation of the chain termination method (reproduced from [6]).

copies so generated are of varying length, and are separated from one another through gel electrophoresis. The original strand is read nucleotide-by-nucleotide by ordering the copy strands in increasing order of length and detecting the last fluorescent nucleotide of each strand, in order. A diagrammatic representation of the Sanger sequencing technique is given in Figure 1.5

Meanwhile, in Illumina's sequencing-by-synthesis approach, the sequencing reaction occurs on the surface of a glass slide (referred to as the flow cell), on which the original fragmented DNA is covalently attached by adapters. The sequencing process occurs in cycles by creating a reverse complementary copy of the template strand by adding fluorescent nucleotides one at a time. The copy of the DNA strand is read nucleotide-by-nucleotide by detecting the last fluorescent nucleotide in each cycle, using light as a signal [2]. This process is shown diagrammatically in Figure 1.6.

On the other hand, nanopore sequencing is done by passing the DNA molecule through a nanopore (ion channels) embedded in an artificial membrane. The membrane used in this process typically has a high electrical resistance. On applying electrical potential, a steady flow of ions is maintained through the nanopores. As the DNA molecules pass through the pore, there is a change in electrical potential and which is detected by a sensor. As each nucleotide leads to a specific amplitude of change, this is used to identify the nucleotides and sequence the DNA strand [2]. A diagrammatic representation of nanopore sequencing technique is presented in Figure 1.7.

Due to its high accuracy, Sanger sequencing is considered the gold standard among sequencing methods, with 99.99% base accuracy. However, this sequencing technique does not allow for parallel sequencing and hence is not a high-throughput method. In contrast, next generation sequencers (NGS) such as Illumina (second generation) and Oxford Nanopore (third generation) are massively parallel and allow for high-throughput sequencing. A comparison between the Illumina and Oxford Nanopore sequencers, as reported in [35], is given in Table 1.2.2.

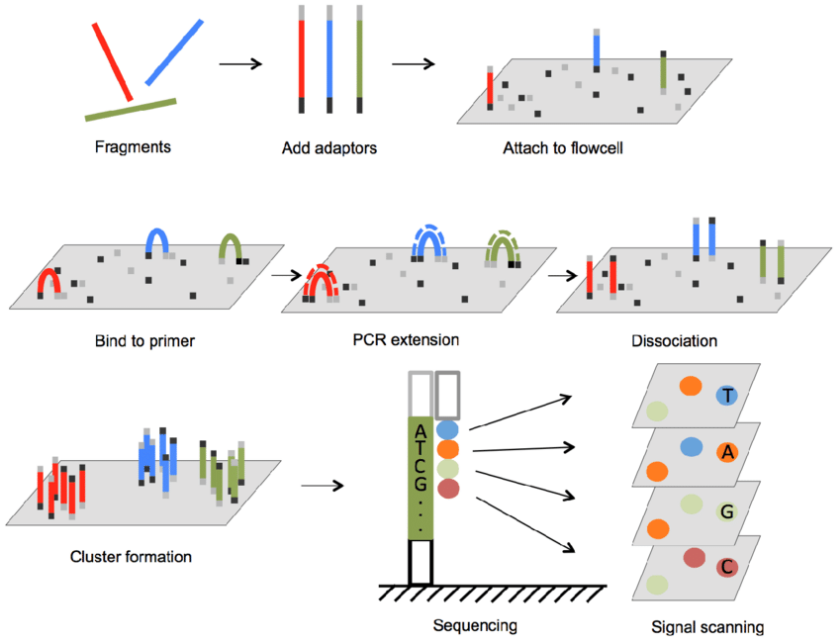

Figure 1.6 Diagrammatic representation of the sequencing-by-synthesis method (reproduced from [7]).

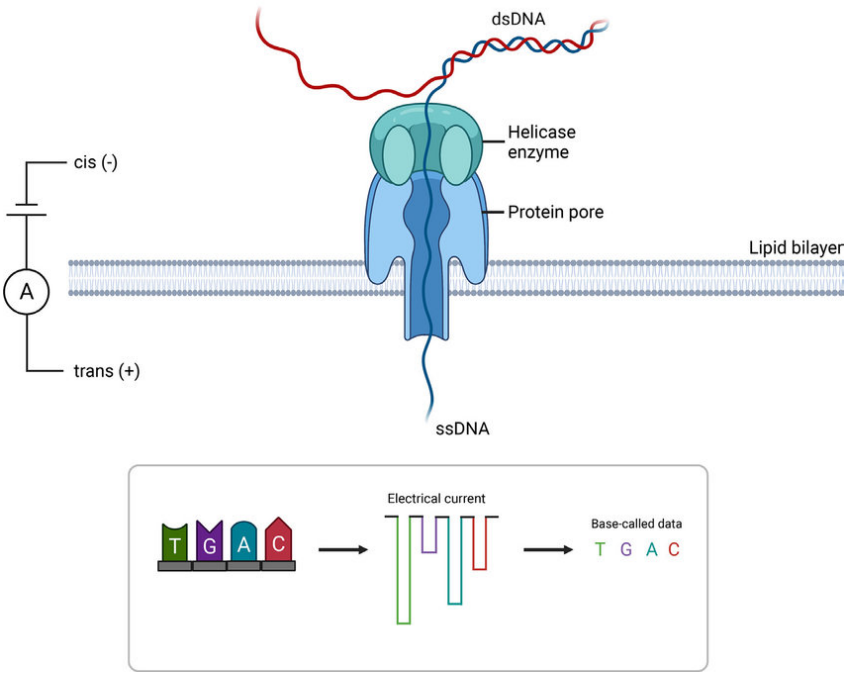

Figure 1.7 Diagrammatic representation of the nanopore sequencing technique (reproduced from [8]).

Sequencer	MiSeq	NovaSeq 600	GridIon	MinIon
Manufacturer	Illumina	Illumina	ONT	ONT
Sequencing Technique	Sequencing-by-Synthesis	Sequencing-by-Synthesis	Nanopore Sequencing	Nanopore Sequencing
Data Output	13-15 Gb	4.8-6 Gb	2.8-50 Gb	$\sim$ 5 Gb
Accuracy	$\sim$ 99.9%	$\sim$ 99.9%	$\sim$ 99%	$\sim$ 95%
Read length	2x300 bp	2x150 bp	>4 Mb	>4 Mb
Time per run	$\sim$ 48 hours	16-44 hours	1 min - 72 hours	0.5-72 hours

Table 1.1 Comparison between various Illumina and Oxford Nanopore Technologies (ONT) sequencers.

1.2.3 Noise in DNA storage: Erasures and Errors

Both storage (write) and retrieval (read) steps of the DNA-based storage pipeline are susceptible to noise. The nature and sources of noise depends on the specificities of the implementation as well as the use-cases [9, 36] (see Figure 1.8). Specifically, the synthesis (write) and sequencing (read) steps are the most significant sources of noise are mostly in the form of *erasures*, and *insertion*, *deletion* and *substitution* (IDS errors) [9, 36].

In many modern sequencers, each nucleotide that is sequenced would be accompanied by a quality score, for instance Phred scores [37], corresponding to the probability that the base call for that nucleotide is erroneous. The Phred Score or Q-score, denoted by Q, is given as

$$Q = -\log_{10} P,$$

where P is the probability of making an erroneous base call. Considering the availability of such quality scores, recent works on the information theoretic characterization of the DNA storage channel (for instance, [38, 39]) have considered the low-quality base calls, below a certain threshold, as *erasures* in the reads. Insertion, and deletion respectively, implies that a nucleotide base is added, and respectively removed, to some position in the strand (thereby increasing, and respectively decreasing, its length), while substitution refers to replacement of a nucleotide base with another.

Several works have analysed the IDS errors and derived results on how to correct such errors for the DNA sequencing problem. For instance, in their work, Sabary *et al.* mathematically analyse the *trace reconstruction* problem (reconstruction of DNA under deletion errors) and generalised reconstruction problem (reconstruction of DNA under all the IDS errors) [36]. They also propose algorithms for effective reconstruction of the original DNA strand under both regimes. Srinivasavardhan *et al.*, meanwhile, presented results on *coded trace reconstruction*, i.e., reconstruction when DNA strand has some redundant bits [40]. The work also introduced a reconstruction algorithm, the Trellis BMA, which has a linear

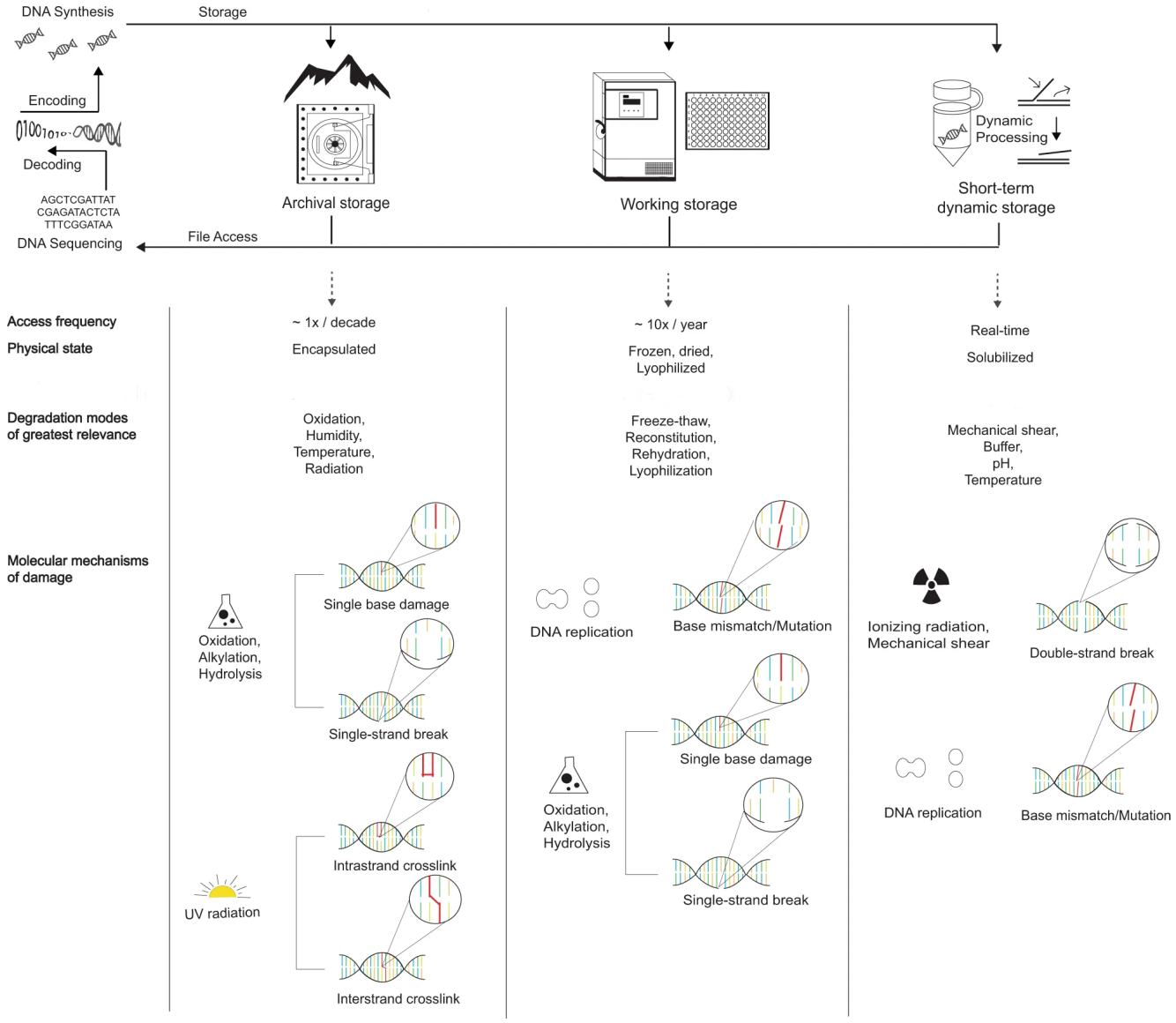

Figure 1.8 Access frequencies, longevities, functional characteristics, and degradation modes for different categories of DNA-based storage systems (reproduced from [9]).

complexity in the number of reads and establishes its effectiveness in correcting IDS errors using both simulated and experimental data.

Error-correction coding for DNA storage under IDS errors was studied in [25]. Here, the model involved the stored data being represented as an unordered set of sequences of equal length. The authors derived Gilbert-Varshamov lower bounds, as well as the sphere-packing upper bounds on achievable cardinalities for error-correction codes within the storage model. Another work [41] describes the HEDGES (Hash Encoded, Decoded by Greedy Exhaustive Search) error-correcting code. The outer-code employed in this method is a modification of the Reed-Solomon code, wherein the code has been diagonalised to include bits from across all the rows to prevent burst errors. The HEDGES encoding process is a hashing function where each bit is made to go through a predefined hashing process. The decoding process involves the generation of a weighted tree of all possibilities, with weights assigned to occurrence of symbols or certain patterns. The algorithm involves searching, through a greedy approach, over this weighted tree to find the least expensive path to the sink node. This coding technique provides significant advantages when used to the DNA storage use-case, as it takes into account several specificities of the current DNA synthesis and sequencing techniques.

All the aforementioned works consider the errors for simpler DNA storage models, in which the inputs strands are read completely or reads are of fixed length with non-overlapping segments, with reads subjected to erasures or IDS errors in either case. To the best of our knowledge, there is no literature that considers errors for cases wherein there are overlapping reads, as in the shotgun sequencing channel.

1.3 Summary of Results

In the present thesis, we consider the shotgun sequencing channel *with erasures*, motivated by the need to incorporate the availability of quality scores of the bases sequenced. We provide a summary of the channel model we consider and the main result of this work, in this section. We elaborate on the channel model and the proof of our main result in subsequent chapters. The model is similar to the Shotgun Sequencing Channel presented in [1], with the addition being that each symbol in each read is assumed to be erased with probability δ. We denote this channel as $\mathsf{SSE}(\delta)$ (therefore, $\mathsf{SSE}(0)$ represents the channel considered in [1]). The essential parameters of the $\mathsf{SSE}(\delta)$ are the *coverage depth*, denoted by c, and the *normalised read length*, denoted by $\bar{L}$. The coverage depth is the expected number of times any given position of the transmitted DNA strand occurs in the collection of reads, while $\bar{L}$ is linked to the read length L as $L = \bar{L} \log n$. A code $\mathcal{C}$ of block-length n for the $\mathsf{SSE}(\delta)$ is a subset of input sequences of length n. The rate of such a code is then written as $R = \log_2 |\mathcal{C}|/n$ bits per channel use (bpcu). We say that a rate R is achievable, if we can decode the transmitted sequence correctly with high probability. The largest possible rate is then defined as the capacity of $\mathsf{SSE}(\delta)$.

This thesis is devoted to obtaining a lower bound for capacity of shotgun sequencing channel with erasures $C_{\mathsf{SSE}(\delta)}$. The main result in this work is the following.

Theorem 1. *Let c and $\bar{L}$ be the parameters of $\mathsf{SSE}(\delta)$ such that $c > 0$ and $\bar{L}(1-\delta) > 1$. Let $\alpha = c/(\bar{L}(1-\delta))$. The rate R is achievable on $\mathsf{SSE}(\delta)$ if*

$$R < \left(1 - e^{-c(1-\delta)}\right) - (1-\delta)\left(e^{-c\left(1-\frac{1}{\bar{L}(1-\delta)}\right)} - e^{-c}\right). \tag{1.1}$$

Observe that at $\delta = 0$, Theorem 1 shows that the rate R is achievable on the channel $\mathsf{SSE}(0)$, if

$$R < (1 - e^{-c\left(1-\frac{1}{\bar{L}}\right)}). \tag{1.2}$$

Indeed, the expression in the R.H.S. of (1.2) is identical to the capacity of the Shotgun Sequencing Channel (without erasures), which was presented in [1]. The proof of Theorem 1 involves demonstrating an achievability scheme via a random code construction and a decoder which uses typicality-like techniques for estimating the true message.

We recount a brief history of the 'random coding' technique, in order to place our result in context. The random coding technique is an idea that originates in Claude Shannon's original work [42], which arguably introduced the area of Information and Coding Theory, in the modern sense. In this work, Shannon introduced multiple fundamental ideas, including modelling information sources and noisy communication channels using probability distributions, the notion of using a 'code' (subsets of input sequences) as a tool to make communication feasible on noisy channels, etc. Further, Shannon proposed the notion of the *capacity of a noisy channel*, obtained for the probabilistic characteristics of the channel-noise, and remarkably showed that the largest rate of any code which can be decoded correctly with high probability is exactly the capacity of the channel. Shannon proposed an extraordinarily simple technique to construct a code called the *random coding* technique, to prove the achievability of rates arbitrarily close to the channel capacity, for a large class of channels known as *memoryless channels* (which is the foundational model upon which modern channels are modelled). In the random coding technique, a subset of 2^{nR} input sequences of length n are chosen at random from the set of all possible input sequences according to some probability distribution. Note that, by definition, the *rate* of this code is exactly R. Using a careful probabilistic analysis, Shannon showed that any input sequence (a *codeword*) transmitted from such a randomly constructed code can be decoded correctly with high probability (indeed, with probability going to 1, as n goes to infinity), as long as the rate R of this code is less than the capacity (precisely, if R is smaller than the capacity by any arbitrary positive constant). Remarkably, Shannon also showed *a converse argument* to this result: that is, if the rate R of a code is larger than the capacity of a channel, then the probability of error in using this code will necessarily be large.

While the case of memoryless channels was masterfully completed by Shannon, other channels which have considerably different models, such as channels *with memory*, multi-look channels (in which more than one 'look' or output sequence is generated from the same input sequence), are much more difficult to handle. The DNA sequencing channels, specifically the Shotgun sequencing channel, is one such channel. In the previous work [1], the authors adapted the random coding argument, along with a novel careful analysis of the probability of error, to show the achievability of rates arbitrarily

close to the R.H.S. expression of (1.2). Further, they were also able to prove a 'matching' converse for the same, thus establishing the same as the capacity of the shotgun sequencing channel. However, the scenario becomes more difficult when we consider erasures in the output reads from this channel, which is the subject of this thesis. While we are able to show the achievability of rates arbitrarily close to the expression in the R.H.S. of (1.1), a matching converse for the same remains a work in progress, at the time of writing this thesis. The mathematical techniques adopted to show the achievability result generally follow those in [1]. However, there are differences that arise. In some parts, we are able to simplify the analysis as compared to [1]. In others, the analysis is more complicated, owing to the fact that we have to account for the erasures in the reads.

1.3.1 Numerical Comparisons of Theorem 1 with prior work

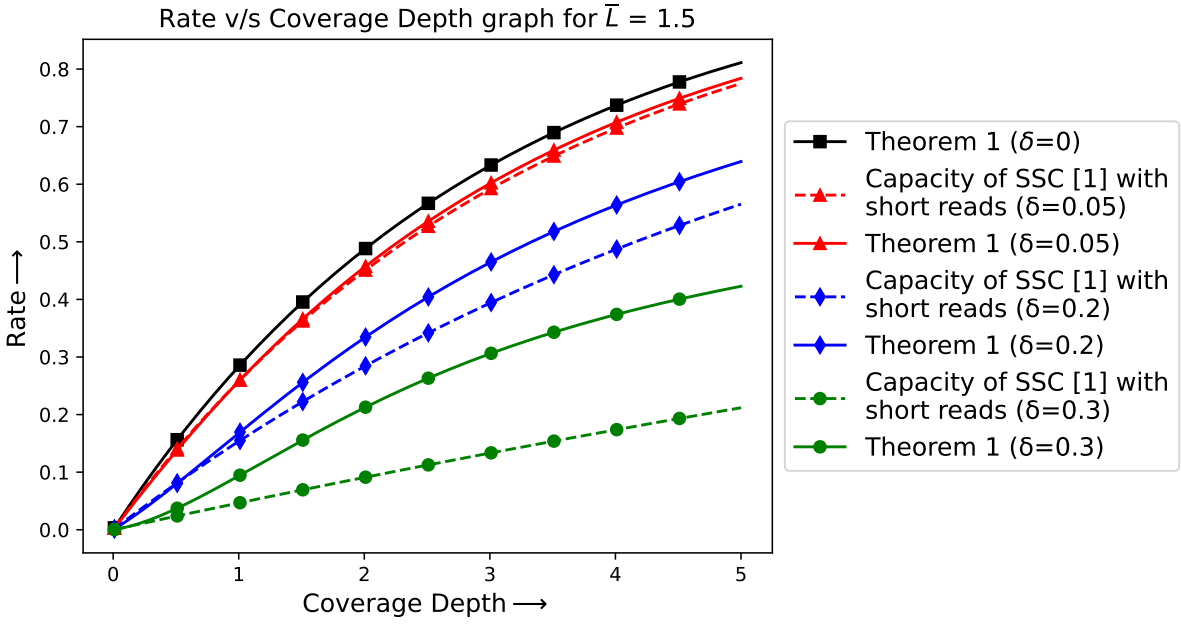

Figure 1.9 The plot shows comparison of the rate from Theorem 1, with $\bar{L} = 1.5$, as the coverage depth c varies, for $\delta = 0, 0.05, 0.2$, and 0.3. These are compared with results from [1].

Figure 1.9, Figure 1.10 and Figure 1.11 plot the upper bound for the achievable rate for $\mathsf{SSE}(\delta)$ from Theorem 1, for $\delta \in \{0, 0.05, 0.2, 0.3\}$, against varying values for the coverage depth c. The parameter $\bar{L}$ is fixed as 1.5, 1.75 and 2 respectively (thus satisfying the requirement $\bar{L}(1-\delta) > 1$, for all chosen δ). We observe that as $\bar{L}$ increases (for a given c, this means K decreases), the capacity increases. Further, the difference in capacity for different δ values decreases, as $\bar{L}$ increases. This might be the case because, for larger values of read-length, the expected overlap between the reads would be greater

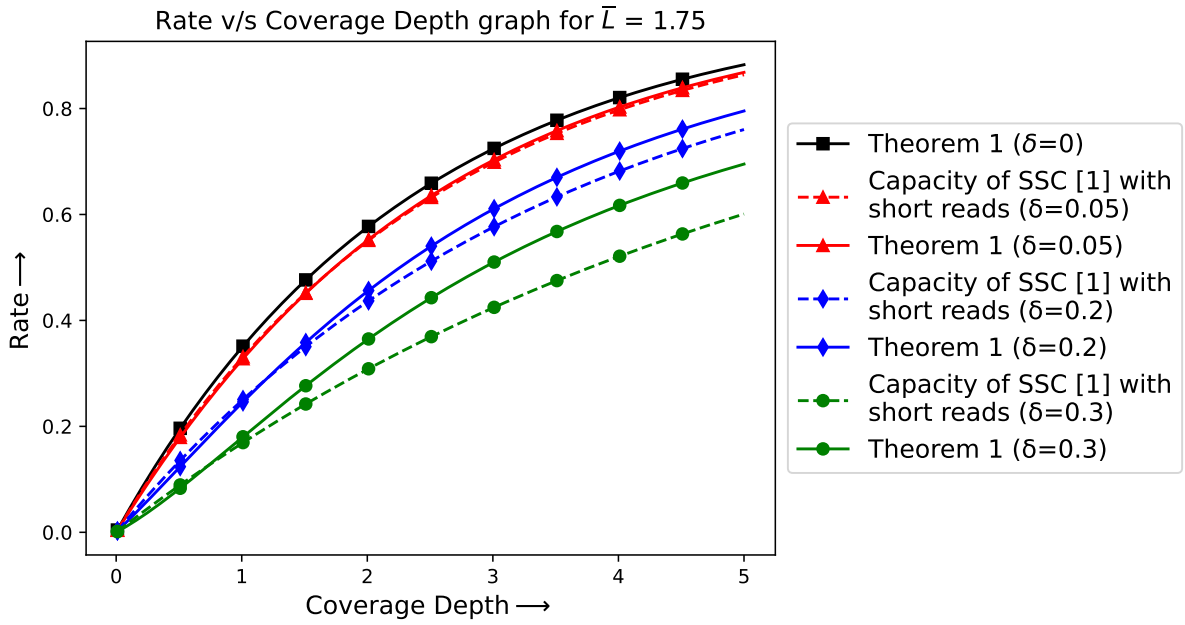

Figure 1.10 The plot shows comparison of the rate from Theorem 1, with $\bar{L} = 1.75$, as the coverage depth c varies, for $\delta = 0, 0.05, 0.2$, and 0.3. These are compared with results from [1].

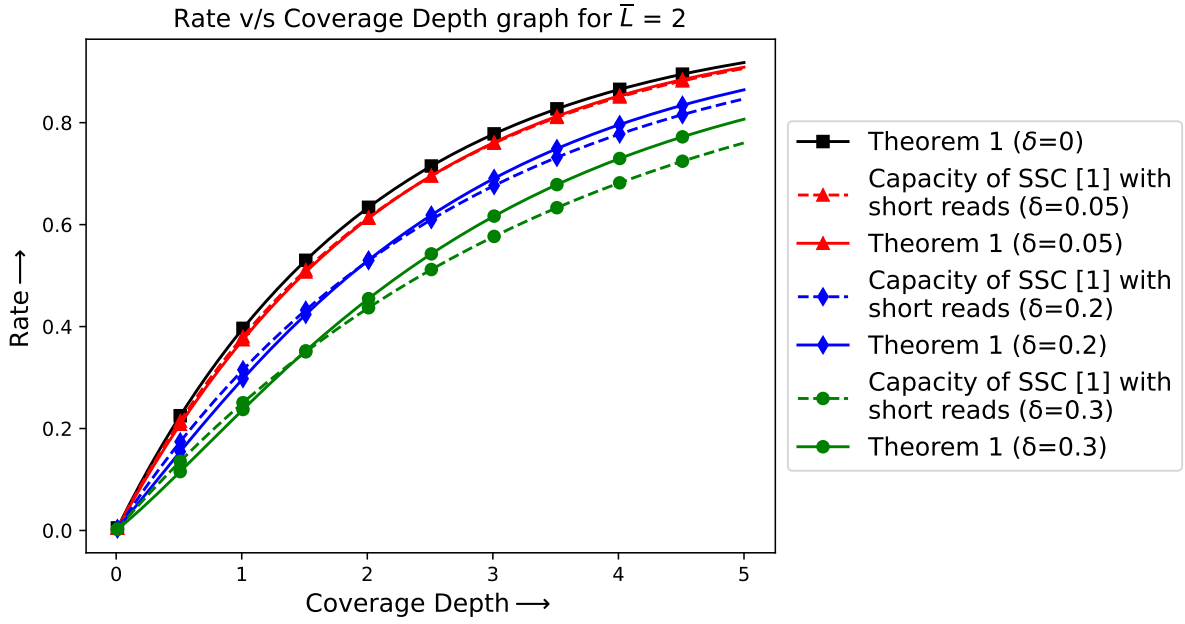

Figure 1.11 The plot shows comparison of the rate from Theorem 1, with $\bar{L} = 2$, as the coverage depth c varies, for $\delta = 0, 0.05, 0.2$, and 0.3. These are compared with results from [1].

and more information about the relative positions of the bits in the transmitted sequence is likely to be conserved.

As a note of comparison, we plot the SSC capacity from [1], with shortened reads of size $\bar{L}(1-\delta)\log n$ (note that the read length is $\bar{L}\log n$ in $\mathsf{SSE}(\delta)$). We observe that this short-read SSC capacity is larger than our bound from Theorem 1, when c is small (roughly, $c < 1$), whereas it is progressively smaller compared to our bound, as c increases (for given L, this means K increases). We will now remark on why this may be the case. Firstly, we observe that, due to the length being shorter, the number of reads K for the SSC is larger than K for the $\mathsf{SSE}(\delta)$, for any specific c. In spite of this, for larger values of c, some information about the relative positions of the bits in the transmitted sequence is likely lost by the SSC, as the read length is shorter. As a result, in the case of $\mathsf{SSE}(\delta)$, each contiguous string obtained after merging the reads as per the overlaps tends to be longer, and the number of such strings will be smaller, in comparison to SSC with shorter reads. Hence, the $\mathsf{SSE}(\delta)$ channel is probably able to preserve the information about the relative positions better, due to the longer reads, in spite of the erasures and lesser K. In the small c regime, there are likely too few reads in $\mathsf{SSE}(\delta)$ to see this advantage. Instead, due to K being less, the reconstructed sequence in $\mathsf{SSE}(\delta)$ likely has many unrecoverable bits, compared to the SSC channel (in spite of its shortened reads). This arguably leads to the behaviour seen in Figure 1.9, Figure 1.10 and Figure 1.11.

1.3.2 Organisation of this thesis

This thesis is organised as follows. Chapter 2 sets the background for our analysis. Related information-theoretical works are presented in Sections 2.1 and 2.2, and Section 2.3 formally states the problem statement and provides the formal description of the channel model in our work (in subsection 2.3.1), The proofs leading to the main result (i.e., Theorem 1) of this work are presented in Chapter 3. This work concludes with some remarks in Section 4, along with a discussion of possible future directions for related research.

Notation used in this thesis

In this work, ordered tuples or strings are denoted with underlines, such as $\underline{x}$. We denote the set of integers $a, a+1, \ldots, b$ as $[a:b]$. The set of integers $[1:b]$ is denoted as $[b]$. For an event A, the indicator random variable associated with the event is denoted by $\mathbb{I}_A$. The probability of an event A is denoted by $\Pr(A)$. The complement of an event A is denoted by $\overline{A}$. For two events A, B, we write $\Pr(A, B)$ for the probability $\Pr(A \cap B)$. For a set S, the set of finite length strings with symbols from S is denoted by $S^\star$. All logarithms are in base 2.

Chapter 2

The Shotgun Sequencing Channel

2.1 Information-Theoretical Approaches to Shotgun Sequencing

One of the early works which analytically modelled the problem of DNA sequence assembly was the work by Lander and Waterman [43]. In their work, Lander and Waterman consider the task of creating a physical map of the genomes in microbial DNA. Typically, the approach to such tasks is to first "fingerprint" many randomly generated clones (smaller fragments of DNA strands) from a recombinant library and subsequently inferring the overlaps between the generated clones using similarity in fingerprints[1]. There are many possible choices of fingerprints that may be used for this task. Lander and Waterman explore the problem of selecting an appropriate fingerprinting scheme and analyse the theoretical considerations that govern the selection of such schemes. To show this, they establish the expected distribution of islands and compare theoretical results with the experimental data from other such physical mapping projects. They also derived various limits on the parameters that govern the sequencing process, which were shown to be necessary for reliable reconstruction. This included read length and coverage in the sequencing process.

Building on the model developed in [43], Motahari *et al.* [10] studied the shotgun sequencing from an information theoretic perspective. In this work, they considered the case where the input sequence $\underline{x}$ is generated uniformly at random from all possible quaternary sequences. In this scenario, length of reads scales as $\bar{L}\log n$, where $\bar{L}$ is a fixed constant and n is the length of the input sequence $\underline{x}$. The work demonstrated some necessary and sufficient conditions on $\bar{L}$, as well as a parameter known as the *coverage depth* c (which captures the average number of times any position in $\underline{x}$ occurs in the collection of reads), for the reconstruction of $\underline{x}$ in the asymptotic regime, i.e., when $n \to \infty$. In particular, the main result of their work established a *critical phenomenon*: reliable sequencing is impossible if $\bar{L} < \frac{2}{H_2(p)}$[2], where p is the probability distribution over the quaternary language, and that if $\bar{L} \geq \frac{2}{H_2(p)}$, then even a minimum normalised coverage depth of $c_{\min} = 1$ is sufficient for reliable sequencing. In other words,

[1] In this context, fingerprint for a DNA strand refers to some properties or patterns present in it, which can be used for comparison with other strands.

[2] $H_2(p)$ denotes the Renyi entropy over the quaternary alphabet given by the expression $H_2(p) \triangleq -\log\left(\sum_i p_i^2\right)$

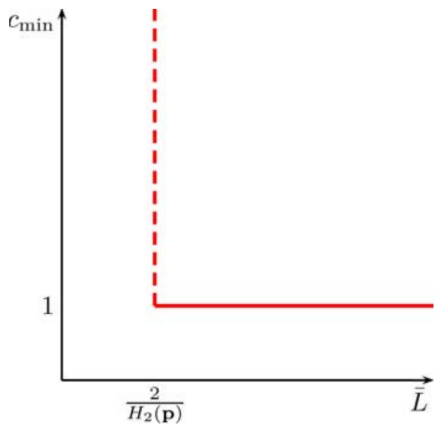

Figure 2.1 Critical phenomenon in read length established by Motahari *et al* [10]. The x-axis and y-axis represent the normalised read length $\bar{L}$, and the minimum coverage depth $c_{\min}$ required for reliable reconstruction, respectively.

$\bar{L} < \frac{2}{H_2(p)}$ and $\bar{L} \geq \frac{2}{H_2(p)}$ represent *repeat-limited* and *coverage-limited* regimes, respectively. The graph in Figure 2.1 represents this result.

The approach taken in [10] also proved useful in studying the fundamental limits of DNA data storage, where the goal is to find the *capacity* of the DNA sequencing channel, i.e., the largest normalized size of any collection of input sequences which can be decoded with vanishing error probability when transmitted through the DNA sequencing channel. For instance, the works [28, 34] model and analyse the capacity of the *Sampling-Shuffling Channel*. In this channel, data is stored as a set of M short DNA strands of equal length $L = \bar{L} \log n$. In the retrieval process, a set of K reads are obtained by sampling (with replacement) over this set. Thus, there can be several or no copies of each of the strands that were stored. Further, reads obtained after the sampling stage are unordered. The works establish the capacity of the sampling-shuffling channel as $C = (1 - e^c)\big(1 - \frac{1}{\bar{L}}\big)$. They also show that a simple index-based coding scheme, which assigns indexes to each of the stored strands, achieves the optimal rate. The diagrammatic representation of this channel can be found in Figure 2.2. The capacities of noisy versions of sampling-shuffling channel, which considered various types of errors including erasures, substitution, noisy sampling etc., were also presented in [34].

Another closely related work is [30], which studies the so-called *Torn Paper Channel*. Unlike the sampling-shuffling channel, in this scenario, the data is stored in the form of a large DNA sequence. In this channel model, during the retrieval process, reads are obtained by fragmenting this large sequence into non-overlapping pieces. This is done as per a geometric distribution. Thus, unlike the sampling-shuffling channel, the reads in the torn-paper channel have a variable length. The work [30] analyses the channel model and establishes the fundamental limits for this channel.

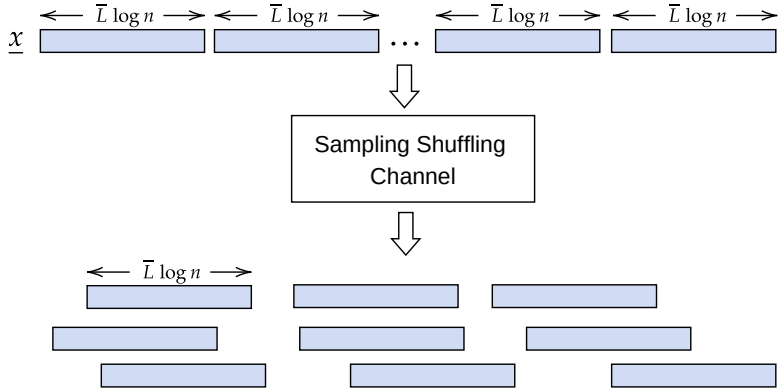

Figure 2.2 Diagrammatic representation of the Sampling-Shuffling Channel.

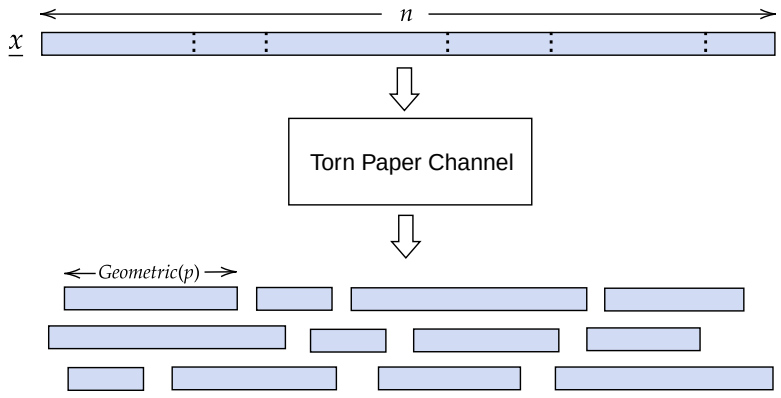

Figure 2.3 Diagrammatic representation of the Torn-Paper Channel.

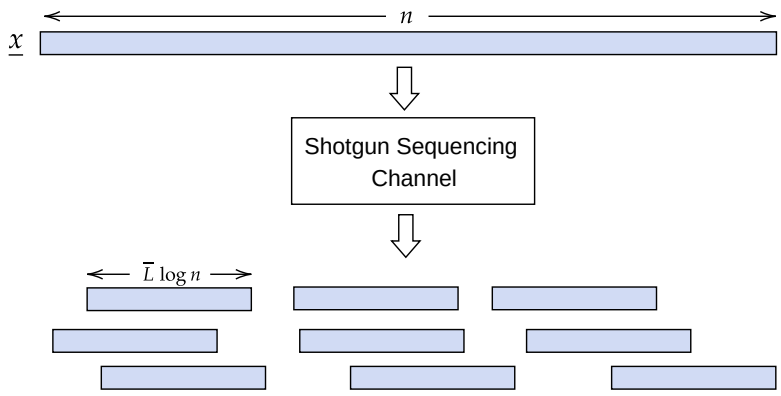

Figure 2.4 Diagrammatic representation of the Shotgun Sequencing Channel.

More specifically, [30] showed that if the "tearing" of the long DNA strand is done as per the distribution $Geometric(p)$, then the capacity of the corresponding channel is given by $C = e^{-\alpha}$, where $\alpha = \lim_{n\to\infty} p \log n$. A diagrammatic representation of the torn-paper channel is given in Figure 2.3. A more recent work [31] on the torn-paper channel analysed the case where certain fragments are "lost", i.e., deleted. It also generalised the result for the torn-paper and sampling-shuffling channels and showed that the capacity of such channels can be expressed in the form $C = (\text{coverage}) - (\text{reordering cost})$, where coverage is the expected number of bits from the transmitted sequence that appear in the collection of reads, and reordering cost is the number of redundant bits remaining in the fragments [31].

2.2 Capacity of Shotgun Sequencing Channel

The capacity of the Shotgun Sequencing Channel (with binary-valued inputs) was presented in [1]. In this work, during the retrieval process, K reads of fixed length $L = \bar{L}\log n$ are generated by sampling starting positions uniformly at random from the indices of the input string $\underline{x}$. The main result of this work is the establishment of the capacity of the channel,

$$C_{\mathsf{SSE}} = 1 - e^{-c\left(1-\frac{1}{L}\right)},$$

where $c = \frac{KL}{n}$ denotes the *coverage depth* (the expected number of times any position of $\underline{x}$ occurs in the collection of reads). Figure 2.4 shows the diagrammatic representation of the Shotgun-Sequencing Channel. Note that, unlike the sampling-shuffling and torn paper channels, which had non-overlapping reads, shotgun sequencing deals with reads which might overlap with one another. The result in [1] is shown through two distinct results: the achievability result that establishes a lower-bound of the capacity, and the converse result that establishes an upper-bound of the capacity.

The achievability part utilises a *random coding argument*, i.e., each symbol in each codeword in the codebook is generated randomly by picking from $\{0, 1\}$ with probability $\frac{1}{2}$. In this part, centrality

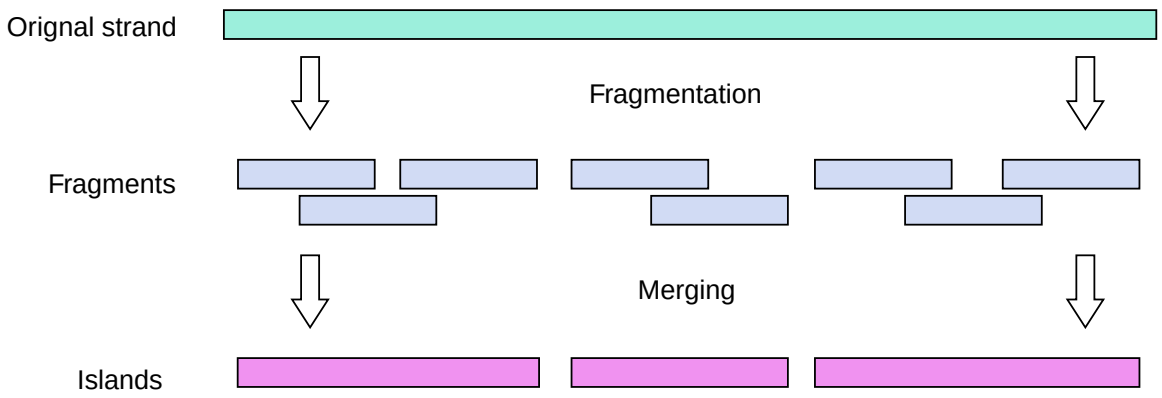

Figure 2.5 Diagrammatic representation of the islands. As shown, islands are formed when obtained when subsequent reads do not overlap with one another.

results corresponding to several important quantities such as the coverage, the number of real islands, the number of reads with given overlap etc. are established. Further, it is shown that these quantities do not deviate from their expected values with high probability.

The work [1] presents a decoding algorithm called the *Partition-Merge Algorithm* (PMA). This algorithm starts by considering all possible tuples that can describe the overlap between the reads in a given ordering. Such vectors are referred to as *partition vectors* in the paper. Using the centrality results that the authors establish, the decoder eliminates partition vectors which deviate greatly from the established typical behaviour. The decoder then brute-forces over all possible ordering of reads and checks whether they can be merged as per the partition vector. If this check goes through, then the reads are merged *as per the partition vector*. The result is a collection of *islands* (each island is a maximal collection of merged reads, which is no longer merge-able with other islands, as shown in Figure 2.5). The islands so created are subsequently added to the set of *candidate islands*. Finally, the decoder compares the codewords to the set of islands. Decoding is said to be successful if and only if there is a unique codeword which is a superstring of all islands in each set of islands in the set of candidate islands. In such a case, the unique codeword identified is returned as output by the decoder. However, if there are more than one or no such codewords, then decoding failure is declared.

The achievability concludes by considering the probability of error in the decoding process and by establishing that probability of error goes to 0, as $n \to \infty$ if

$$R \leq 1 - e^{-c\left(1-\frac{1}{L}\right)}.$$

The converse part starts by establishing a constraint on the read length, i.e., by showing that reconstruction is impossible if $\bar{L} < 1$. Subsequently, it uses *genie-aided arguments* to show the converse. More specifically, [1] first considers an *omniscient genie* which can merge all reads as per their real

overlap. However, the upper-bound so obtained is much higher than the achievability result. Hence, the authors subsequently consider a *constrained genie*. This genie can only merge islands under certain conditions. Through this, they obtain a converse result

$$R \leq 1 - e^{-c\left(1-\frac{1}{L}\right)}.$$

In both the genie-aided arguments, results from an earlier work [31] on the torn-paper channel were used.

2.3 Our Work: Modelling the Shotgun Sequencing Channel with Erasure Noise

We consider the shotgun sequencing channel *with erasures*, motivated by the need to incorporate the availability of quality scores of the bases sequenced. The model is similar to that in [1], with the addition being that each symbol in each read is assumed to be erased with probability δ. We denote this channel as $\mathsf{SSE}(\delta)$ (thus, $\mathsf{SSE}(0)$ is the channel considered in [1]).

In this work, we obtain an achievability result for the channel $\mathsf{SSE}(\delta)$, thus showing a lower bound on its capacity. The mathematical techniques adopted to show the achievability and the converse results generally follow those in [1]. However, there are differences that arise. In some parts, we are able to simplify the analysis as compared to [1]. In others, the analysis is more complicated, owing to the fact that we have to account for the erasures in the reads.

2.3.1 Channel Description for the Shotgun Sequencing Channel with Erasures

We follow the description and terminology similar to those in [1], as the present work essentially extends the achievability result in [1] to the erasure scenario.

The channel takes a n-length binary[3] string $\underline{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ as input, corresponding to a message $W \in [2^{nR}]$ chosen at random. The output of the channel can be envisioned as a concatenation of two stages, as shown in Figure 2.6. Firstly, the channel samples K substrings of length L, from $\underline{x}$. We denote these by a multiset $\tilde{\mathcal{Y}} = \{\underline{\tilde{y}}_1, \ldots, \underline{\tilde{y}}_K\}$. Each read $\underline{\tilde{y}}_i$ is obtained by first selecting a position $\mathsf{S}(\underline{\tilde{y}}_i)$ uniformly at random from $[n]$, and then taking the L-length (contiguous) substring from the position $\mathsf{S}(\underline{\tilde{y}}_i)$ onwards, i.e., $\underline{\tilde{y}}_i = (x_{\mathsf{S}(\underline{\tilde{y}}_i)}, \ldots, x_{\mathsf{S}(\underline{\tilde{y}}_i)+L-1}) \in \{0,1\}^L$. When $\mathsf{S}(\underline{\tilde{y}}_i) > n - L + 1$, similar to the circular DNA model in [10], we assume that the substring is obtained in a cyclic wrap-around fashion, for ease of analysis. Thus, $\tilde{\mathcal{Y}}$ can be thought of as the output of a noise-free shotgun sequencing channel (as in [1]), when the input is $\underline{x}$.

In the second stage, each read $\underline{\tilde{y}}_i$ is assumed to pass through a binary erasure channel with erasure probability δ (denoted by $\mathsf{BEC}(\delta)$), thus erasing each position in $\underline{\tilde{y}}_i$ with probability δ independently, to

[3]The symbols in a DNA sequence take values in a quaternary set, but for simplicity we assume the symbols to be binary. Our results can be easily extended to the quarternary case.

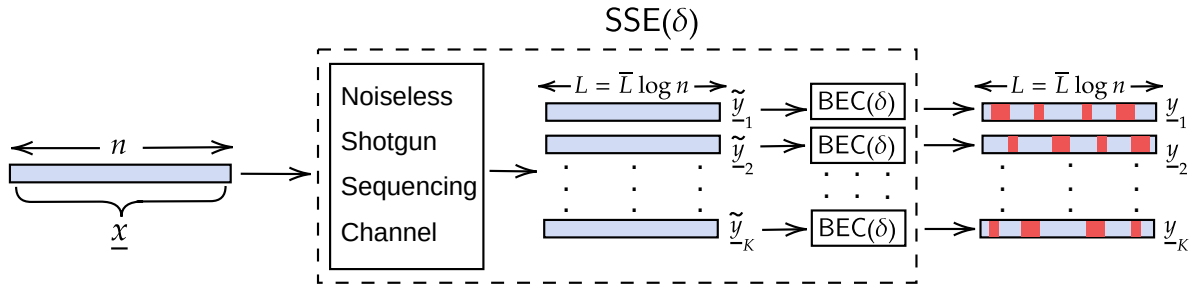

Figure 2.6 The Shotgun Sequencing Channel with Erasures ($\mathsf{SSE}(\delta)$). The collection $\tilde{\mathcal{Y}} = \{\underline{\tilde{y}}_1, \underline{\tilde{y}}_2, \cdots, \underline{\tilde{y}}_K\}$ may be visualized as the output of the Shotgun Sequencing Channel [1], and $\mathcal{Y} = \{\underline{y}_1, \underline{y}_2, \cdots, \underline{y}_K\}$ is the output of $\mathsf{SSE}(\delta)$, after bits in each read are erased (indicated in bold/red) with probability δ.

obtain $\underline{y}_i \in \{0, 1, \perp\}^L$, where $\perp$ denotes an erasure. The multiset of these reads, denoted as $\mathcal{Y} = \{\underline{y}_i : i \in [K]\}$, is the output of the channel. Note that the start positions are unaltered, i.e., $\mathsf{S}(\underline{y}_i) = \mathsf{S}(\underline{\tilde{y}}_i), \forall i$. We denote this shotgun sequencing channel with erasures as $\mathsf{SSE}(\delta)$.

A rate R is said to be *achievable* on $\mathsf{SSE}(\delta)$ if the message W can be reconstructed from $\mathcal{Y}$ using some decoding algorithm with a probability of error that is vanishing as n grows large. The capacity of $\mathsf{SSE}(\delta)$ is then defined as $C_{\mathsf{SSE}(\delta)} \triangleq \lim_{n\to\infty} \sup\{R : R \text{ is achievable}\}$.

The expected number of times a coordinate of $\underline{x}$ (say the j^{th} coordinate) is sequenced in the first stage is called the coverage depth, denoted as c. Thus, $c \triangleq \mathbb{E}(\sum_{i=1}^{K} \mathbb{I}_{\{j \in [\mathsf{S}(\underline{y}_i) : \mathsf{S}(\underline{y}_i)+L-1]\}})$. A simple calculation reveals that

$$c = \frac{KL}{n} \tag{2.1}$$

We assume that the length of each read is $L = \Theta(\log n) \triangleq \bar{L} \log n$. for some positive $\bar{L}$. As in [1], we study the regime where c and $\bar{L}$ are some positive constants. Thus in our regime, $K = \frac{cn}{\bar{L} \log n} = \Theta\left(\frac{n}{\log n}\right)$.

As mentioned in Chapter 1, our goal in this thesis is to determine a lower bound for capacity of shotgun sequencing channel with erasures $C_{\mathsf{SSE}(\delta)}$. In particular, we show Theorem 1, which shows that, for $\mathsf{SSE}(\delta)$, the rates that satisfy $R < \left(1 - e^{-c(1-\delta)}\right) - (1-\delta)\left(e^{-c\left(1-\frac{1}{\bar{L}(1-\delta)}\right)} - e^{-c}\right)$ are achievable, where c and $\bar{L}$ are the coverage depth and normalised read length respectively.

Chapter 3

Achievable Rates for Shotgun Sequencing Channel with Erasures: Proof of Theorem 1

We use a random coding argument to show the achievability of the rate as in Theorem 1. We outline the main components of our code design below. While these share similarities to the techniques in [1], the decoding algorithm and the proof arguments are more complex, owing to consideration of the reads with erasures.

3.1 Outline of the Coding Scheme

- Codebook: A codebook with 2^{nR} codewords, denoted as $\mathcal{C} = \{\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_{2^{nR}}\}$, is generated by picking each symbol of $\underline{x}_j$ independently and uniformly at random from $\{0, 1\}$, for each $j \in [2^{nR}]$.

- Encoder: To communicate the message W (chosen uniformly at random from $[2^{nR}]$) through the channel $\mathsf{SSE}(\delta)$, the encoder communicates the codeword $\underline{x}_W \in \mathcal{C}$ through the channel. The output $\mathcal{Y}$, a set of reads as described in Section 2.3.1, is generated post-sequencing.

- Decoder: The decoding algorithm we propose takes as input the collection of reads $\mathcal{Y}$ and generates an estimate $\hat{W}$ of the transmitted message, or a failure. We briefly describe the process of obtaining the estimate $\hat{W}$ from $\mathcal{Y}$. The decoder proceeds in three phases. In the first phase, which we call the **merge phase**, the decoder first implements a merging process of the reads. Such a merging process will be run for all possible orderings of the reads, considering multiple possible 'typical' ways to merge the reads, where the typicality will be defined based on the concentration properties of some quantities we will subsequently define.

  For each such typical merge process, we get a set of *islands*, where an island refers to a string of maximal length obtained in the merging process (formal definitions follow in subsequent sections). Ultimately, upon going through all possible orderings, several such island sets may be generated. In the second phase, which we call the **filtering phase**, these island sets are then filtered based on further typicality constraints. The filtered island sets which pass the final typicality conditions are referred to

as *candidate island sets*. The third and final phase is called the **compatibility check phase**. In this phase, for each candidate island set, the decoder checks if all the islands of that candidate island set occur as compatible substrings of any codeword. If there is precisely one codeword $\underline{x}_{\hat{w}}$ in $\mathcal{C}$ that passes this check, across all the candidate island sets, then the estimate is declared as $\hat{W} = \hat{w}$. Otherwise, a decoding failure is declared. We show that the decoding algorithm results in the correct estimate, i.e., $W = \hat{W}$ with high probability, as n grows large. A more precise description and analysis of the decoding is provided in section 3.4 and 3.6. Section 3.2 and section 3.3 describe the various quantities required for the description and analysis of the decoder, and the concentration results on some of these quantities, respectively.

3.2 Merging and Coverage: Definitions and Terminology

We now give the formal definitions and terminology for various quantities. Again, these quantities are either identical or parallel to those defined in [1].

Definition 1 (Length and Size of string)**.** *For any $\underline{u} \in \{0, 1, \perp\}^\star$, the length of $\underline{u}$ is denoted by $\ell(\underline{u})$. The size of $\underline{u}$ is the number of unerased bits in $\underline{u}$ and is denoted by $\ell_{ue}(\underline{u})$.*

Definition 2 (Prefix and Suffix)**.** *For a string $\underline{v} \in \{0, 1, \perp\}^l$ and any positive integer $l' \leq l$, a string $\underline{z} \in \{0, 1, \perp\}^{l'}$ is said to be a l'-suffix of $\underline{v}$, if $(v_{l-l'+1}, \ldots, v_l) = \underline{z}$, and is denoted by $\texttt{suffix}(\underline{v}, l')$. Similarly, if $(v_1, \ldots, v_{l'}) = \underline{z}$, then $\underline{z}$ is said to be a l'-prefix of $\underline{v}$ and is denoted by $\texttt{prefix}(\underline{v}, l')$.*

Definition 3 (Compatibility, l-compatible strings and substring compatibility)**.** *Let $\underline{u}$ and $\underline{v}$ be any two strings in $\{0, 1, \perp\}^l$. We say that $\underline{v}$ and $\underline{u}$ are compatible, if*

$$u_i = \perp \text{ or } v_i = \perp, \quad \text{for any } i \in [l] \text{ s.t. } u_i \neq v_i.$$

For any $\underline{u}, \underline{v} \in \{0, 1, \perp\}^\star$ (not necessarily of same length), the string $\underline{v}$ is said to be a compatible substring of $\underline{u}$, if $\underline{v}$ is compatible with any substring of $\underline{u}$. Finally, for any $\underline{u}, \underline{v} \in \{0, 1, \perp\}^\star$, we say $\underline{v}$ is l-compatible *with $\underline{u}$, if $\texttt{suffix}(\underline{u}, l)$ and $\texttt{prefix}(\underline{v}, l)$ are compatible.*

We define the merging of two reads in the following manner.

Definition 4 (Merge of two strings)**.** *Let $\underline{u}$ and $\underline{v}$ be any two strings in $\{0, 1, \perp\}^\star$ such that $\underline{v}$ is l-compatible with $\underline{u}$. Let $\texttt{suffix}(\underline{u}, l) = \underline{u}'$ and $\texttt{prefix}(\underline{v}, l) = \underline{v}'$. Suppose $\ell_{ue}(\underline{u}') \neq 0$. Then, we say that $\underline{u}$ and $\underline{v}$ are* mergeable with overlap l*. The output of the merge operation is defined as the string $(\underline{u}'' \mid \underline{z} \mid \underline{v}'')$ obtained by the concatenation of three substrings: $\underline{u}''$, $\underline{z}$ and $\underline{v}''$, where $\underline{u}'' =$*

$\texttt{prefix}(\underline{u}, \ell(\underline{u}) - l)$, $\underline{v}'' = \texttt{suffix}(\underline{v}, \ell(\underline{v}) - l)$, *and* $\underline{z}$ *is defined as follows.*

$$z_i = \begin{cases} 0 & \text{if } u'_i = 0 \text{ or } v'_i = 0, \\ 1 & \text{if } u'_i = 1 \text{ or } v'_i = 1, \\ \perp & \text{if } u'_i = \perp \text{ and } v'_i = \perp \end{cases} .$$

With respect to the merge defined above, we term the substring $\underline{u}'$ *as the* merging suffix*, or simply the* suffix*. Fig. 3.1 shows an illustration of this merge operation.*

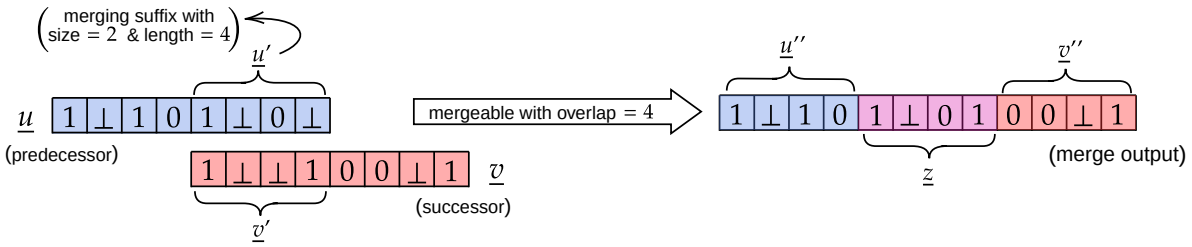

Figure 3.1 Diagrammatic representation of the merging process between two reads $\underline{u}$ and $\underline{v}$. The reads here are mergeable with overlap 4. The predecessor and successor reads are $\underline{u}$ and $\underline{v}$ respectively. The merging suffix is $\underline{u}'$ and size of the merging suffix is $\ell_{ue}(\underline{u}') = 2$, corresponding to the unerased positions in the merging suffix. The merge output is given by the concatenation of $\underline{u}''$, $\underline{z}$ and $\underline{v}''$. Note that $\underline{z}$ has erasures at a given position if and only if there is an erasure in the corresponding position in $\underline{u}'$ and $\underline{v}'$.

We also recall that, during the sequencing process, each of the read $\underline{y} \in \mathcal{Y}$ has a certain starting position $\mathsf{S}(\underline{y})$. The following terminologies are regarding the ground truth of $\mathcal{Y}$.

Definition 5 (True Successors, Ordering, and Overlaps)**.** *The true successor of a read* $\underline{y}_1 \in \mathcal{Y}$ *is another read* $\underline{y}_2 \in \mathcal{Y}$*, such that* $\mathsf{S}(\underline{y}_2) \geq \mathsf{S}(\underline{y}_1)$ *(in cyclic wrap around fashion) and* $(\mathsf{S}(\underline{y}_2) - \mathsf{S}(\underline{y}_1))$ *is smallest among all reads* $\underline{y}_2 \in \mathcal{Y} \setminus \{\underline{y}_1\}$*. Thus, the true ordering, is an ordering of the* K *reads such that each read is succeeded by its true successor. The true overlap between any read* $\underline{y}_1$ *and its true successor* $\underline{y}_2$ *is defined to be* 0*, if* $\mathsf{S}(\underline{y}_2) > \mathsf{S}(\underline{y}_1) + L - 1$ *(in cyclic wrap around manner). If* $\mathsf{S}(\underline{y}_1) \leq \mathsf{S}(\underline{y}_2) \leq \mathsf{S}(\underline{y}_1) + L - 1$*, then the true overlap of* $\underline{y}_1$ *with* $\underline{y}_2$ *is* $\mathsf{S}(\underline{y}_1) + L - \mathsf{S}(\underline{y}_2)$*.*

As mentioned before, our algorithm merges the reads corresponding to various orderings and typical overlaps. We now formally define the notion of an island arising out of a merging process of the reads, following a given ordering and a tuple prescribing the sizes of the merging suffixes.

Definition 6 (Orderings, Islands, and True Islands)**.** *Let ζ denote a permutation of $[K]$. Consider the ordering of the K reads defined by ζ. With respect to this ordering, the read $\underline{y}_{\zeta(i)}$ is called the predecessor of the read $\underline{y}_{\zeta(i+1)}$, while $\underline{y}_{\zeta(i+1)}$ is the successor of $\underline{y}_{\zeta(i)}$. Consider $\underline{\omega} = (\omega_1, \ldots, \omega_K) \in [0 : L]^K$. For some $j \in [K]$, for some positive integer l, suppose that the following conditions hold.*

- *$\omega_{j'} > 0$ and $\underline{y}_{\zeta(j')}$ is mergeable with $\underline{y}_{\zeta(j'+1)}$ with size of the merging suffix being $\omega_{j'}$, for all $j' \in [j : j + l - 1]$ (in cyclic wrap around fashion).*
- *$\omega_{j-1} = \omega_{j+l} = 0$.*

Then the string obtained by the merging of the reads $\underline{y}_{\zeta(j')} : j' \in [j : j + l - 1]$ successively with their respective successors, is called an island. If ζ^t is the true ordering of the reads, and each read $\underline{y}$ is merged with its true successor $\underline{y}'$ (as per ζ^t) based only on the true overlap, i.e., if $\mathsf{S}(\underline{y}) + L - \mathsf{S}(\underline{y}') > 0$, then the islands so obtained are called true islands.

Another quantity that will use to check the goodness of our islands is the expected number of unerased bits in them. Towards that end, we have the following two definitions.

Definition 7. *(A bit being covered and visibly covered): The i^{th} bit of $\underline{x}_W$, denoted by x_i, is said to be* covered *by a read $\underline{y} \in \mathcal{Y}$ if $\mathsf{S}(\underline{y}) \in [i - L + 1 : i]$. Further, x_i is said to be visibly covered by $\underline{y}$ if it is covered by $\underline{y}$ and further unerased in $\underline{y}$. The bit x_i is said to be covered (visibly covered) by the collection of reads $\mathcal{Y}$ if it is covered (visibly covered, respectively) by at least one read in $\mathcal{Y}$.*

Definition 8 (Visible Coverage)**.** *The visible coverage denoted by Φ_v of the collection $\mathcal{Y}$ is defined as the fraction of the bits which are visibly covered, by the reads in $\mathcal{Y}$. Thus, $\Phi_v \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\{x_i \text{ is visibly covered by reads in } \mathcal{Y}\}}$.*

Remark 1. *The notion of visible coverage is parallel to coverage in [1], where coverage (denoted by Φ) for a collection $\mathcal{Y}$ is the fraction of bits which covered by reads in $\mathcal{Y}$, i.e, $\Phi \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\{x_i \text{ is covered by reads in } \mathcal{Y}\}}$.*

To analyse the errors the decoder can make while merging, we need to bound the different possible ways a read can be merged with other reads in the set $\mathcal{Y}$. To capture this, we define the quantity $M_{\underline{z}}$.

Definition 9 (The quantity $M_{\underline{z}}$)**.** *For a string $\underline{z} = \{0, 1, \perp\}^l, l \in [L]$, the random variable $M_{\underline{z}}$ is defined as the number of reads in $\mathcal{Y}$, which are l-compatible with $\underline{z}$ (i.e., which have a l-length prefix that is compatible with $\underline{z}$). Thus,*

$$M_{\underline{z}} \triangleq \sum_{\underline{y} \in \mathcal{Y}} \mathbb{I}_{\{\underline{y} \text{ is } l\text{-compatible with } \underline{z}\}}$$

Towards assessing the goodness of some overlap tuples, we need the following quantity, denoted by $G(\tau)$.

Definition 10 (The quantity $G(\tau)$)**.** *We define*[1] $G(\tau)$ *to be the number of reads in* $\mathcal{Y}$*, such that for each such read, the size of the merging suffix with its true successor is* $\tau \log n$*. Thus,*

$$G(\tau) \triangleq \sum_{i\in[K]} \mathbb{I}_{\omega_i^t = \tau \log n}, \tag{3.1}$$

where $\underline{\omega}^t = (\omega_1^t, \ldots, \omega_K^t)$ *is the sequence of sizes of the true merging suffixes.*

3.3 Concentration Results and Bounds on Quantities

In order to show achievability, we will first prove concentration results for some of the quantities that we have defined. Our first concentration result is on the number of true islands. Note that by definition, erasures do not affect the existence of any true islands. Hence, the proof for the following lemma is identical to the proof of Lemma 2 in [1].

Lemma 1 (Concentration of Number of True Islands)**.** *Let the number of true islands be* K'*. Thus, for any* $\epsilon > 0$

$$\lim_{n\to\infty} \Pr\left(\left|K' - Ke^{-c}\right| \geq \epsilon K e^{-c}\right) = 0.$$

For the sake of completeness, proof of this lemma given in Appendix B.

We now show the concentration of the visible coverage.

Lemma 2 (Concentration of Visible Coverage)**.** *For any* $\epsilon > 0$*, the visible coverage* Φ_v *satisfies*

$$\lim_{n\to\infty} \Pr\Big(|\Phi_v - (1 - e^{-c(1-\delta)})| > \epsilon(1 - e^{-c(1-\delta)})\Big) = 0. \tag{3.2}$$

Proof. We have the following equalities,

$$\begin{aligned}
\mathbb{E}[\Phi_v] &= \frac{1}{n} \cdot \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{I}_{\{x_i \text{ is visibly covered by in reads in } \mathcal{Y}\}}\right] \\
&= \Pr(x_1 \text{ is visibly covered by in reads in } \mathcal{Y}) \\
&= 1 - \Pr(x_1 \text{ is not visibly covered by in reads in } \mathcal{Y}) \\
&= 1 - \Pr(x_1 \text{ is not visibly covered by } \underline{y}_j, \forall j \in \{1, \cdots, K\}) \\
&= 1 - \Pr(x_1 \text{ is not visibly covered by read } \underline{y}_1)^K \\
&= 1 - (1 - \Pr(x_1 \text{ is visibly covered by read } \underline{y}_1))^K.
\end{aligned}$$

[1] *In this work, the* τ *in* $G(\tau)$ *represents the size of the merging suffix (i.e., the number of unerased bits in the overlapping portion of the predecessor) normalised by* $\log n$*, rather than the normalised length of overlap itself, as defined in [1].*

Now, $\Pr(x_1 \text{ is visibly covered by read } \underline{y}_1) = \frac{L}{n}(1-\delta)$. Hence,

$$\begin{aligned}
\mathbb{E}[\Phi_v] &= 1 - \left(1 - \left(\frac{L}{n}(1-\delta)\right)\right)^{K} \\
&= 1 - \left(1 - \left(\frac{L}{n}(1-\delta)\right)\right)^{\left(\frac{cn}{L}\right)} \\
&= 1 - \left(1 - \left(\frac{\bar{L}\log n}{n}(1-\delta)\right)\right)^{\left(\frac{cn}{\bar{L}\log n}\right)}.
\end{aligned}$$

Let $t = \left(\frac{n}{\bar{L}\log n(1-\delta)}\right)$ Hence, we get

$$\begin{aligned}
\lim_{n\to\infty} \mathbb{E}[\Phi_v] &= \lim_{t\to\infty} \mathbb{E}[\Phi_v] \\
&= \lim_{t\to\infty} \left(1 - \left(1 - \frac{1}{t}\right)^{(c(1-\delta)t)}\right) \\
&= 1 - \lim_{t\to\infty} \left(\left(1 - \frac{1}{t}\right)^{(c(1-\delta)t)}\right) \qquad (3.3) \\
&= (1 - e^{-c(1-\delta)}). \qquad (3.4)
\end{aligned}$$

The rest of this proof follows almost identical arguments as in the Lemma 1 of [1], essentially bounding the variance of Φ_v and using the Chebyshev inequality to complete the result. We therefore omit the details. □

Remark 2. *Note that the concentration of the coverage Φ (i.e., Lemma 1 in [1]) also follows from Lemma 2, by substituting $\delta = 0$.*

Lemma 3 shows the concentration of the parameter $G(\tau)$ in some regime around its mean. Lemma 3 is similar to Lemma 4 in [1], and the proof follows similar arguments. However, a key difference with [1] is that the deviation around the mean in Lemma 4 of [1] is a function of $\bar{G}(\tau)$, whereas for our purpose here, the deviation $\epsilon n/\log^2 n$ suffices.

Lemma 3. *For any $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left(\bigcup_{\tau\in\mathcal{T}} \left\{ \left|G(\tau) - \bar{G}(\tau)\right| \geq \frac{\epsilon n}{\log^2 n} \right\}\right) = 0,$$

where $\mathcal{T} = \{\frac{1}{\log n}, \frac{2}{\log n}, \cdots, \bar{L}\}$ and $\bar{G}(\tau) \triangleq \mathbb{E}[G(\tau)]$.

Proof. We know that,

$$\Pr\left(\bigcup_{\tau\in\mathcal{T}} \left\{ \left|G(\tau) - \bar{G}(\tau)\right| \geq \epsilon \frac{n}{\log^2 n} \right\}\right) \leq \frac{\mathrm{Var}(G(\tau))}{\epsilon^2 \cdot \left(\frac{n}{\log^2 n}\right)^2}. \qquad (3.5)$$

Let A_i denote the event that the i^{th} read $\underline{y}_i$ overlaps with its true successor with size of merging suffix as $\tau \log n$. Thus, $G(\tau) = \sum_{i=1}^{K} \mathbb{I}_{A_i}$. As the random variables $A_i : i \in [K]$ are identically distributed, we have $\bar{G}(\tau) = K \Pr(A_i)$, for any i.

As $G(\tau)$ is the sum of indicator random variables, the following is true (for a proof, see Chapter 4 in [44], for instance),

$$\mathrm{Var}(G(\tau)) \leq \bar{G}(\tau) + \sum_{i,j \in [K]: i \neq j} \mathrm{Cov}(\mathbb{I}_{A_i}, \mathbb{I}_{A_j}), \tag{3.6}$$

where

$$\mathrm{Cov}(\mathbb{I}_{A_i}, \mathbb{I}_{A_j}) = \mathbb{E}[\mathbb{I}_{A_i}\mathbb{I}_{A_j}] - \mathbb{E}[\mathbb{I}_{A_i}]\mathbb{E}[\mathbb{I}_{A_j}], \tag{3.7}$$

is the covariance between the random variables $\mathbb{I}_{A_i}$ and $\mathbb{I}_{A_j}$.

Now,

$$\mathbb{E}[\mathbb{I}_{A_i}] = \Pr(A_i) = \frac{\bar{G}(\tau)}{K} = \mathbb{E}[\mathbb{I}_{A_j}]. \tag{3.8}$$

Further, A_i and A_j are independent if reads $\underline{y}_i$ and $\underline{y}_j$ do not overlap, i.e., $|\mathsf{S}(\underline{y}_i) - \mathsf{S}(\underline{y}_j)| \geq L$. Thus, we have

$$\begin{aligned}
\mathbb{E}[\mathbb{I}_{A_i}\mathbb{I}_{A_j}] &= \Pr(A_i, A_j) \\
&\leq \Pr(A_i, A_j | \{|\mathsf{S}(\underline{y}_i) - \mathsf{S}(\underline{y}_j)| \geq L\}) \\
&\quad + \Pr(\{|\mathsf{S}(\underline{y}_i) - \mathsf{S}(\underline{y}_j)| < L\}) \\
&\leq \Pr(A_i) \Pr(A_j) + \frac{L}{n} \\
&= \frac{\bar{G}(\tau)^2}{K^2} + \frac{L}{n}.
\end{aligned} \tag{3.9}$$

Using (3.7), (3.8) and (3.9) in (3.6), we get

$$\begin{aligned}
\mathrm{Var}(G(\tau)) &\leq \bar{G}(\tau) + \sum_{i,j \in [K]: i \neq j} \frac{L}{n} \\
&\leq \bar{G}(\tau) + K^2 \frac{L}{n} &(3.10) \\
&\overset{(a)}{\leq} K + cK = (1+c)K, &(3.11)
\end{aligned}$$

where (a) holds as $G(\tau) \leq K$, by definition. Using (3.11) in (3.5),

$$\Pr\left(\bigcup_{\tau \in \mathcal{T}} \left\{ \left| G(\tau) - \bar{G}(\tau) \right| \geq \epsilon \frac{n}{\log^2 n} \right\} \right) \leq \frac{\mathrm{Var}(G(\tau))}{\epsilon^2 \cdot \left(\frac{n}{\log^2 n} \right)^2} \leq \frac{(1+c)K}{\epsilon^2 \cdot \left(\frac{n}{\log^2 n} \right)^2} = \Theta\left(\frac{\log^3 n}{n} \right) \to 0,$$

as $n \to \infty$. □

Lemma 4 gives us concentration results for the parameter $M_{\underline{z}}$ in various regimes. This is similar to Lemma 3 in [1], except that the proof is simpler than in [1].

Lemma 4. *For $\underline{z} \in \cup_{l\in[L]}\{0,1,\perp\}^l$, let $\tau_{ue}(\underline{z}) = \ell_{ue}(\underline{z})/\log n$. The following are then true for any $\epsilon > 0$,*

a) $\lim_{n\to\infty} \Pr\Big(\bigcup_{\underline{z}:\ \tau_{ue}(\underline{z})\le 1-\epsilon}\big\{\big|M_{\underline{z}} - Kn^{-\tau_{ue}(\underline{z})}\big| \ge \epsilon K n^{-\tau_{ue}(\underline{z})}\big\}\Big) = 0.$

b) $\lim\limits_{n\to\infty} \Pr\Big(\bigcup_{\underline{z}:\ \tau_{ue}(\underline{z})>1-\epsilon}\{M_{\underline{z}} \ge n^{\epsilon}\}\Big) = 0.$

Proof. Consider vectors with erasures of the form $\underline{z} = \{0,1,\perp\}^l$, $l \le L$. For the transmitted codeword $\underline{x} \in \{0,1\}^n$, let $D \triangleq \{i \in [n]\colon (x_i,\cdots,x_{i+l-1})$ is l-compatible with $\underline{z}\}$. Therefore, $M_{\underline{z}} = \sum_{i=1}^{K} \mathbb{I}_{\{\mathsf{S}(\underline{y}_i)\in D\}}$. Observe that $\Pr(\mathbb{I}_{\{\mathsf{S}(\underline{y}_i)\in D\}} = 1) = 1/2^{\tau_{ue}(\underline{z})\log n} = n^{-\tau_{ue}(\underline{z})}$. Further, the random variables $\mathbb{I}_{\{\mathsf{S}(\underline{y}_i)\in D\}} : i \in [K]$ are independent, as $\mathsf{S}(\underline{y}_i) : i \in [K]$ are independent random variables.

We first prove part a). Consider $\tau_{ue}(\underline{z}) \le 1-\epsilon$.

$$
\begin{aligned}
&\Pr\left(\bigcup_{\underline{z}:\tau_{ue}(\underline{z})\le 1-\epsilon}\left\{\left|M_{\underline{z}} - Kn^{-\tau_{ue}(\underline{z})}\right| \ge \epsilon K n^{-\tau_{ue}(\underline{z})}\right\}\right)\\
&\le \sum_{\underline{z}:\tau_{ue}(\underline{z})\le 1-\epsilon} \Pr\Big(\Big\{\Big|M_{\underline{z}} - Kn^{-\tau_{ue}(\underline{z})}\Big| \ge \epsilon K n^{-\tau_{ue}(\underline{z})}\Big\}\Big)\\
&\overset{(a)}{\le} \bar{L}\log n \cdot n^{\bar{L}\log 3}\cdot\\
&\qquad \max_{\underline{z}:\tau_{ue}(\underline{z})\le 1-\epsilon} \Pr\Big(\Big\{\Big|M_{\underline{z}} - Kn^{-\tau_{ue}(\underline{z})}\Big| \ge \epsilon K n^{-\tau_{ue}(\underline{z})}\Big\}\Big)\\
&\overset{(b)}{\le} \bar{L}\log n \cdot n^{\bar{L}\log 3}\cdot 2e^{-\left(\frac{K\cdot n^{-\tau_{ue}(\underline{z})}\epsilon^2}{(1-n^{-\tau_{ue}(\underline{z})})}\right)}\\
&\le \bar{L}\log n \cdot n^{\bar{L}\log 3}\cdot 2e^{-\left(\frac{c\cdot n^{1-\tau_{ue}(\underline{z})}\epsilon^2}{2L}\right)}\\
&\overset{(c)}{\le} \bar{L}\log n \cdot n^{\bar{L}\log 3}\cdot 2e^{-\left(\frac{c\cdot n^{\epsilon}\epsilon^2}{2L}\right)}.
\end{aligned}
$$

Here (a) holds as $|\{\underline{z} : \tau_{ue}(\underline{z}) \le 1-\epsilon\}| \le \sum_{l=1}^{L} 3^l \le \sum_{l=1}^{L} 3^L \le L\cdot 3^L = \bar{L}\log n \cdot n^{\bar{L}\log 3}$. The inequality (b) is from Hoeffding's inequality in Lemma 5 (which we use with the parameters $X_i = \mathbb{I}_{\{\mathsf{S}(\underline{y}_i)\in D\}}$, $N = K$, and $p = n^{-\tau_{ue}(\underline{z})}$), and (c) follows as $\tau_{ue}(\underline{z}) \le 1-\epsilon$. Thus, part a) of the lemma can be seen to be true, from the R.H.S. of the inequality (c).

Similarly, for $\tau_{ue}(\underline{z}) > 1-\epsilon$, we have,

$$\Pr\left(\bigcup_{\underline{z}:\tau_{ue}(\underline{z})>1-\epsilon}\{M_{\underline{z}}\geq n^{\epsilon}\}\right)\leq \bar{L}\log n\cdot n^{\bar{L}\log 3}\max_{\underline{z}:\tau_{ue}(\underline{z})>1-\epsilon}\Pr\big(\{M_{\underline{z}}\geq n^{\epsilon}\}\big)$$

$$\overset{(a)}{\leq} \bar{L}\log n\cdot n^{\bar{L}\log 3}\cdot e^{-\left(\frac{\left(n^{\epsilon}-Kn^{-\tau_{ue}(\underline{z})}\right)^2}{2Kn^{-\tau_{ue}(\underline{z})}(1-n^{-\tau_{ue}(\underline{z})})}\right)}$$

$$\leq \bar{L}\log n\cdot n^{\bar{L}\log 3}\cdot e^{-\left(\frac{\left(n^{\epsilon}-(c/L)n^{1-\tau_{ue}(\underline{z})}\right)^2}{2(c/L)n^{1-\tau_{ue}(\underline{z})}}\right)}$$

$$\overset{(b)}{\leq} \bar{L}\log n\cdot n^{\bar{L}\log 3}\cdot e^{-\left(\frac{(n^{\epsilon}-(c/L)n^{\epsilon})^2}{2(c/L)n^{\epsilon}}\right)}$$

$$\leq \bar{L}\log n\cdot n^{\bar{L}\log 3}\cdot e^{-\left(\frac{n^{\epsilon}(\bar{L}\log n-c)^2}{2c\bar{L}\log n}\right)},$$

where (a) is due to Hoeffding's inequality in Lemma 6 (with parameters $X_i=\mathbb{I}_{\{\mathsf{S}(\underline{y}_i)\in D\}}$, $p=n^{-\tau_{ue}(\underline{z})}$, and $x=n^{\epsilon}-Kn^{-\tau_{ue}(\underline{z})}$) and (b) is because $\tau_{ue}(\underline{z})>1-\epsilon$. From the above R.H.S. expression, it is easy to see that part b) of the lemma holds. □

Remark 3. *Observe that the bounds on $M_{\underline{z}}$ obtained from Lemma 4 are independent on the vector $\underline{z}$ itself, depending instead only on $\ell_{ue}(\underline{z})=\tau_{ue}(\underline{z})\log n$.*

3.4 Decoding Algorithm

We are now ready to present the decoding algorithm, Algorithm 1. Following the outline presented in section 3.1, we can understand the decoding algorithm in three phases. The first phase attempts merging of the reads $\mathcal{Y}$, using the suffix-sizes from a special subset of $[0:L]^K$, defined as follows.

Definition 11 (Typical Suffix-size tuples)**.** *For a tuple $\underline{\omega}\in[0,L]^K$ and integer $b\in[0:L]$, let $\texttt{count}(\underline{\omega},b)$ be the number of times b appears in $\underline{\omega}$. For any $\epsilon>0$, we define the set Ω of typical suffix-size tuples as the set of $\underline{\omega}\in[0,L]^K$ satisfying the following conditions.*

- $|\texttt{count}(\underline{\omega},0)-Ke^{-c}|\leq \epsilon Ke^{-c}$, *and*
- $|\texttt{count}(\underline{\omega},\tau)-\bar{G}(\tau)|\leq \frac{\epsilon n}{\log^2 n},\forall\tau\in\mathcal{T}$, *where* $\mathcal{T}=\{\frac{1}{\log n},\frac{2}{\log n},\cdots,\bar{L}\}$.

For each typical suffix-size tuple $\underline{\omega}=(\omega_1,\ldots,\omega_K)\in\Omega$, for each permutation ζ of $[K]$ (which we view as an ordering of the K reads), Algorithm 1 attempts to merge the reads such that each value ω_i is the size of the merging suffix of each read $\underline{y}_{\zeta(i)}$The intuition for the first condition in Definition 11 follows from the following observation. As mentioned in Definition 6, in the process of merging, an

island is created, with the last read of the island being read $\underline{y}_{\zeta(i)}$ whenever $\underline{\omega}_i = 0$. Thus, $\texttt{count}(\underline{\omega}, 0)$ denotes the number of islands generated in the process of merging the reads, if it is successful. This is the merge phase of Algorithm 1, which is performed for each ordering ζ and each typical suffix-size tuple $\underline{\omega} \in \Omega$, corresponding to the steps 4-8.

In the filtering phase, the set $\mathcal{I}$ of islands, obtained from a successful merge process, is filtered based on its visible coverage. That is, the total number of unerased bits in the islands obtained, denoted by $\phi(\mathcal{I})$, is checked (as per step 9) for the following condition (designed based on Lemma 2).

$$|\phi(\mathcal{I}) - (1 - e^{-c(1-\delta)})| \leq \epsilon(1 - e^{-c(1-\delta)}).$$

If the above check is passed, then the set $\mathcal{I}$ of islands obtained is added to a collection CI of *candidate island sets* (step 10).

The third phase is the compatibility check phase, which is done in steps 15-18. Any codewords which are compatible with all the islands of any set $\mathcal{I}$ of islands is added into a set $\hat{\mathcal{X}}$ of candidate codewords. Finally, in steps 19-21, the estimated message index $\hat{w} \in [2^{nR}]$ is returned, corresponding to the only codeword in the candidate set $\hat{\mathcal{X}}$, if that is the case. Else, a failure is declared (steps 22-23).

3.5 Brief overview of the proof of achievability

We first show that the true ordering ζ^t is surely picked by step 5, and the true suffix-size tuple $\underline{\omega}^t$ belongs to Ω (and thus considered in step 4) with high probability for large n, following the concentration properties shown in Lemma 1 and Lemma 3. The set of islands resulting from these will be the set of true islands, which will have visible coverage close to the expected visible coverage, following Lemma 2. Thus, the true set of islands, with visible coverage close to the expected visible coverage, will pass the check in step 9, and thus will be in the candidate island set with high probability. Thus, the true transmitted codeword $\underline{x}_w$ belongs to the set of candidate codewords $\hat{\mathcal{X}}$, with high probability. Finally, using the concentration lemmas shown in section 3.3, we show that $|\hat{\mathcal{X}}| = 1$ (therefore containing only the true codeword) with high probability, for large n, provided R satisfies (1.1) in Theorem 1. The precise arguments of the proof follow in section 3.6.

3.6 Detailed Proof of Achievability

Our analysis of the decoder's probability of error follows that in [1]. We define the following undesirable events.

$$B_1 = \{K' > b_1\}, \qquad B_2 = \{\Phi_v < b_2\},$$
$$B_3 = \bigcup_{\underline{z} \in \mathcal{Z}} \{M_{\underline{z}} > b_3(\tau)\}, \qquad B_4 = \bigcup_{\tau \in \mathcal{T}} \{G(\tau) > b_4(\tau)\},$$

Algorithm 1: Decoding Algorithm

Input: Codebook $\mathcal{C} = \{\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_{2^{nR}}\}$, Reads $\mathcal{Y}$, Typical suffix-size tuples Ω.

Output: Estimate $\hat{w}$ of the input codeword, or Failure.

Initialize: Collection of Candidate Islands $\mathsf{CI} \leftarrow \emptyset$.

for *each suffix-size tuple* $\underline{\omega} \in \Omega$ **do**

  for *permutation* ζ *of* $[K]$ **do**

    if $\underline{y}_{\zeta(i)}$ *and* $\underline{y}_{\zeta(i+1)}$ *are mergeable with size of merging suffix as* ω_i, $\forall i \in [K]$ **then**

      Merge reads according to the suffix-size tuple $\underline{\omega}$ to form set of islands $\mathcal{I}$.

      $\phi(\mathcal{I}) \leftarrow$ number of unerased bits in resulting islands.

      if $|\phi(\mathcal{I}) - (1 - e^{-c(1-\delta)})| \leq \epsilon(1 - e^{-c(1-\delta)})$ **then**

        Add $\mathcal{I}$ to CI.

      end

    end

  end

end

Candidate codewords $\hat{\mathcal{X}} \leftarrow \emptyset$.

for *each set of islands* $\mathcal{I} \in \mathsf{CI}$ **do**

  Insert into $\hat{\mathcal{X}}$, all the $\underline{x} \in \mathcal{C}$ such that all islands in $\mathcal{I}$ are compatible substrings of $\underline{x}$.

end

if $|\hat{\mathcal{X}}| = 1$ **then**

  Estimate $\hat{w} \leftarrow$ message index corresponding to $\underline{x} \in \hat{\mathcal{X}}$.

  return $\hat{w}$

else

  return FAIL (decoding failure)

end

where $\mathcal{Z} = \cup_{l \in [L]} \{0, 1, \perp\}^l$, and the constants $b_1, b_2, b_3(\tau)$, and $b_4(\tau)$ are defined as follows.

$$b_1 \triangleq (1+\epsilon)Ke^{-c}, \quad b_2 \triangleq (1-\epsilon)(1 - e^{-c(1-\delta)}),$$

$$b_3(\tau) \triangleq \begin{cases} (1+\epsilon)n^{1-\tau} & \text{if } \tau \leq 1-\epsilon, \\ n^{\epsilon} & \text{if } \tau > 1-\epsilon, \end{cases},$$

$$b_4(\tau) \triangleq \bar{G}(\tau) + \frac{\epsilon n}{\log^2 n}.$$

Let $B = \bigcup_{i=1}^{4} B_i$. Recall that the transmitted message W is chosen uniformly at random. We thus get the following expression for the probability of decoding error.

$$\Pr(W \neq \hat{W}) = \Pr(W \neq \hat{W} | W = 1) \leq \Pr(W \neq \hat{W} | W = 1, \bar{B}) + \Pr(B), \tag{3.12}$$

by the law of total probability.

Now, for some island set $\mathcal{I} \in \mathsf{CI}$, if each island in $\mathcal{I}$ is a compatible substring of some codeword $\underline{x}_i$, then we say that *island set $\mathcal{I}$ is compatible with i*. The event $\{\hat{W} \neq 1\}$ can occur in one the following ways: (a) no island set in CI is compatible with $i = 1$ (event E_1), or (b) some island set in CI is compatible with $i \in [2 : 2^{nR}]$ (event E_i). Thus, we can write,

$$\Pr(W \neq \hat{W} | W = 1, \bar{B}) + \Pr(B) \leq \Pr(E_1 | W = 1, \bar{B}) + \sum_{i=2}^{2^{nR}} \Pr(E_i | W = 1, \bar{B}) + \Pr(B). \tag{3.13}$$

Now, from Lemmas 1-4, we then have $\lim_{n\to\infty} \Pr(B) = 0$. As argued in section 3.5, the event $\bar{B}_1 \cap \bar{B}_2 \cap \bar{B}_4$ ensures the occurrence of the true island set in the CI set with high probability as $n \to \infty$, following Definition 11 and steps 4 to 9 of Algorithm 1. This true island set is surely compatible with $i = 1$, as $W = 1$ is the true message in our case. Thus, $\lim_{n\to\infty} \Pr(E_1 | W = 1, \bar{B}) = 0$.

Let the collection of candidate islands $\mathsf{CI} = \{\mathcal{I}_1, \ldots, \mathcal{I}_{|\mathsf{CI}|}\}$. Hence, we have, for $i \in [2 : 2^{nR}]$,

$$\begin{aligned}
&\Pr(E_i | W = 1, \bar{B}) \\
&= \Pr(\exists \mathcal{I} \in \mathsf{CI} \text{ s.t. } \mathcal{I} \text{ is compatible with } i \mid W = 1, \bar{B}) \\
&\leq \sum_{s=1}^{|\mathsf{CI}|} Pr\big(\mathcal{I}_s \text{ is compatible with } i \mid W = 1, \bar{B}\big).
\end{aligned} \tag{3.14}$$

Now, recall that the number of islands in $\mathcal{I}_s$ is at most b_1, for any typical suffix-size tuple $\underline{\omega} \in \Omega$. Further, from the condition on $\phi(\mathcal{I}_s)$ in the filtering phase, the visible coverage of $\mathcal{I}_s$ must be at least b_2. Thus, the islands in $\mathcal{I}_s$ can be arranged in one of at most n^{b_1} orderings, when checking for compatibility with message i. Further, for any such ordering, the probability of compatibility is at most 2^{-nb_2}, as the bits in the codewords $\underline{x}_1$ and $\underline{x}_i$ are generated independently and uniformly at random (since $i \neq 1$). Thus, the event that $\mathcal{I}_s$ is compatible with i can be bounded as

$$Pr\big(\mathcal{I}_s \text{ is compatible with } i \mid W = 1, \bar{B}\big) \leq \frac{n^{b_1}}{2^{nb_2}}. \tag{3.15}$$

Using (3.15), (3.14), (3.13) and (3.12), we have

$$\begin{aligned}
\Pr(W \neq \hat{W}) &\leq 2^{nR} \cdot |\mathsf{CI}| \cdot n^{b_1} \cdot \frac{1}{2^{nb_2}} + o(1) \\
&= 2^{nR + \log|\mathsf{CI}| + b_1 \log n - nb_2} + o(1) \\
&= 2^{nR + \log|\mathsf{CI}| + (1+\epsilon)Ke^{-c}\log n - n(1-\epsilon)\left(1 - e^{-c(1-\delta)}\right)} + o(1).
\end{aligned}$$

Using the fact that $K \log n = cn/\bar{L}$, we see that $\Pr(W \neq \hat{W}) \to 0$ as $n \to \infty$, if

$$\begin{aligned} R &\leq \lim_{n\to\infty} \Big((1-\epsilon)\Big(1 - e^{-c(1-\delta)}\Big) - (1+\epsilon)\frac{ce^{-c}}{\bar{L}} - \frac{1}{n}\log|\mathsf{CI}|\Big) \\ &\leq (1-\epsilon)\Big(1 - e^{-c(1-\delta)}\Big) - (1+\epsilon)\frac{ce^{-c}}{\bar{L}} - \lim_{n\to\infty}\frac{1}{n}\log|\mathsf{CI}|. \end{aligned} \tag{3.16}$$

In Appendix C, the term $\frac{1}{n}\log|\mathsf{CI}|$ is shown to be upper bounded as shown below, for any $p > 0$ and $d > 0$.

$$\frac{1}{n}\log|\mathsf{CI}| \leq \frac{(1+2p)d}{(1-\delta)} \cdot \Big(\frac{c}{\bar{L}}\Big)^2 \cdot e^{\alpha p d - c} \cdot \left(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)} - \frac{e^{2\alpha d}\big(\big(e^{\alpha(1+d)} - e^{\alpha}\big) - d\big(e^{\alpha(1+d)}-1\big)\big)}{(e^{\alpha d}-1)^2}\right), \tag{3.17}$$

where $\alpha = \frac{c}{\bar{L}(1-\delta)}$.

Using (3.17) in (3.16) and letting $p \to 0$, we get

$$R < \Big(1 - e^{-c(1-\delta)}\Big) - \frac{ce^{-c}}{\bar{L}} - \beta(d), \tag{3.18}$$

where

$$\beta(d) = \frac{d}{(1-\delta)} \cdot \Big(\frac{c}{\bar{L}}\Big)^2 \cdot e^{-c} \cdot \left(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)} - \frac{e^{2\alpha d}\big(\big(e^{\alpha(1+d)} - e^{\alpha}\big) - d\big(e^{\alpha(1+d)}-1\big)\big)}{(e^{\alpha d}-1)^2}\right).$$

Now, as $d \to 0$,

$$\lim_{d\to 0}\beta(d) = (1-\delta)(e^{-c\left(1-\frac{1}{\bar{L}(1-\delta)}\right)} - e^{-c}) + \frac{c}{\bar{L}}e^{-c}. \tag{3.19}$$

The proof of (3.19) is provided in Appendix D.

Thus, using (3.19) in (3.18), we have proven our result, which is that if

$$R < \Big(1 - e^{-c(1-\delta)}\Big) - (1-\delta)\left(e^{-c\left(1-\frac{1}{\bar{L}(1-\delta)}\right)} - e^{-c}\right),$$

then $\Pr(W \neq \hat{W}) \to 0$ as $n \to \infty$.

Chapter 4

Conclusion and Future Work

In this thesis, we identified achievable rates for the shotgun sequencing channel $\mathsf{SSE}(\delta)$ with erasure probability δ, using techniques inspired from [1]. The expression in Theorem 1 is identical to the capacity of the erasure-free shotgun sequencing channel, derived in [1]. As expected, we see that the obtained achievable rate for $\mathsf{SSE}(\delta)$ reduces progressively as δ increases. For the shotgun sequencing channel, the converse result obtained in [1] depends on results from prior work on the torn-paper channel [30, 31]. A converse result for $\mathsf{SSE}(\delta)$ can possibly be obtained by generalizing these results to torn-paper channels with erasures; however, this appears not straightforward, a fact that has been noticed before (see [31, Section VII]).

In particular, the converse result in [1] works is established by using the fact that the capacity of such channels can be expressed in the form $C = (\text{coverage}) - (\text{reordering cost})$, where reordering cost is the number of redundant bits remaining in the fragments. However, due to the added complexity of erasures, the calculation of this quantity is not trivial and requires further analysis.

Apart from erasures due to substitution, the Shotgun Sequencing pipeline is also prone to other types of error. The sensors in the sequencing process may misidentify the nucleotide base, which could lead to substitution errors. In practice, we are likely to see both erasure (due to low quality scores) and substitution errors (quality score above threshold, but nucleotide is misidentified[1]) in the output of the sequencing process. Further, the PCR amplification process that precedes the sequencing step is also prone to IDS errors.

Hence, the analysis of the shotgun sequencing pipeline is still a work in progress. Some directions for further works are as below:

1. Converse result for $\mathsf{SSE}(\delta)$, giving the upper-bound for the capacity of the channel.

2. Development of practical error-correction codes for $\mathsf{SSE}(\delta)$, which realises the bound on achievable rates.

[1] While high quality score implies high probability that the nucleotide is identified correctly, there is still a small probability that the nucleotide is misidentified. This becomes significant when the total number of nucleotide in the sequence becomes extremely large.

3. Examination of the shotgun sequencing pipeline in the context of other errors (insertion, deletion, substitution etc.).

Appendix A

Concentration inequalities used in this work

The following Hoeffding-type concentration inequalities are used in this work (see [45], for instance).

Lemma 5. *For i.i.d. Bernoulli random variables $X_1, X_2 \cdots X_N$ with parameter p,*

$$\Pr\left(\left|\frac{1}{N}\sum_{i=1}^{N} X_i - p\right| \geq \epsilon p\right) \leq 2e^{-\frac{Np\epsilon^2}{1-p}}.$$

Lemma 6. *For i.i.d. Bernoulli random variables $X_1, X_2 \cdots X_N$ with parameter p,*

$$\Pr\left(\sum_{i=1}^{N} X_i - Np \geq x\right) \leq e^{-\frac{x^2}{2Np(1-p)}}.$$

Appendix B

Proof of Lemma 1

We start by noticing that if the suffix of a read does not overlap with any other read, then the read must be the last read of a real island. Hence, the number of real islands can be obtained by counting the total number of such reads. Therefore,

$$\begin{aligned}\mathbb{E}[K'] &= \sum_{i=1}^{K} \mathbb{E}[\mathbb{I}_{\underline{y}_i \text{ does not overlap with any other read}}] \\ &= K \Pr(\underline{y}_i \text{ does not overlap with any other read}) \\ &= K\left(1 - \frac{L}{n}\right)^{K-1}.\end{aligned}$$

Hence,

$$\begin{aligned}\lim_{n\to\infty} \frac{\log n}{n}\mathbb{E}[K'] &= \lim_{n\to\infty} \frac{\log n}{n} K\left(1 - \frac{L}{n}\right)^{K-1} \\ &= \lim_{n\to\infty} \frac{\log n}{n} \frac{cn}{\bar{L}\log n}\left(1 - \frac{\bar{L}\log n}{n}\right)^{\frac{cn}{\bar{L}\log n}-1} \\ &= \frac{c}{\bar{L}} e^{-c}.\end{aligned}$$

Therefore, from the definition of limit, for large enough n we have $|\mathbb{E}[K'] - Ke^{-c}| < \left(\frac{\epsilon}{2}\right)Ke^{-c}$, for any $\left(\frac{\epsilon}{2}\right) > 0$. Thus, by triangle inequality

$$|K' - Ke^{-c}| \leq |K' - \mathbb{E}[K']| + |\mathbb{E}[K'] - Ke^{-c}| \leq \epsilon Ke^{-c},$$

if $|K' - \mathbb{E}[K']| \leq \left(\frac{\epsilon}{2}\right)Ke^{-c}$.

Hence,

$$\Pr\Big(\big|K' - Ke^{-c}\big| \geq \epsilon Ke^{-c}\Big) \leq \Pr\Big(\big|K' - \mathbb{E}[K']\big| \geq \left(\frac{\epsilon}{2}\right)Ke^{-c}\Big) \tag{B.1}$$

Further, for large enough n, we have,

$$\Pr\Big(\big|K' - \mathbb{E}[K']\big| \geq \Big(\frac{\epsilon}{2}\Big)Ke^{-c}\Big) \leq \frac{\mathrm{Var}(K')}{\big(\frac{\epsilon}{2}\big)^2(Ke^{-c})^2}. \tag{B.2}$$

Let A_i denote the event that the $i^{\mathtt{th}}$ read $\underline{y}_i$ does not overlap with another read. Thus, $K' = \sum_{i=1}^{K} \mathbb{I}_{A_i}$. As the random variables $A_i : i \in [K]$ are identically distributed, we have $\mathbb{E}[K'] = K\Pr(A_i)$, for any i.

As K' is the sum of indicator random variables, the following is true (for a proof, see Chapter 4 in [44], for instance),

$$\mathrm{Var}(K') \leq \mathbb{E}[K'] + \sum_{i,j\in[K]:i\neq j} \mathrm{Cov}(\mathbb{I}_{A_i}, \mathbb{I}_{A_j}), \tag{B.3}$$

where

$$\mathrm{Cov}(\mathbb{I}_{A_i}, \mathbb{I}_{A_j}) = \mathbb{E}[\mathbb{I}_{A_i}\mathbb{I}_{A_j}] - \mathbb{E}[\mathbb{I}_{A_i}]\mathbb{E}[\mathbb{I}_{A_j}], \tag{B.4}$$

is the covariance between the random variables $\mathbb{I}_{A_i}$ and $\mathbb{I}_{A_j}$.

Now,

$$\mathbb{E}[\mathbb{I}_{A_i}] = \Pr(A_i) = \left(1 - \frac{L}{n}\right)^{K-1} = \mathbb{E}[\mathbb{I}_{A_j}]. \tag{B.5}$$

Further, A_i and A_j are independent if reads $\underline{y}_i$ and $\underline{y}_j$ do not overlap, i.e., $|\mathsf{S}(\underline{y}_i) - \mathsf{S}(\underline{y}_j)| \geq L$. Thus, we have

$$\begin{aligned}
\mathbb{E}[\mathbb{I}_{A_i}\mathbb{I}_{A_j}] &= \Pr(A_i, A_j) \\
&\leq \Pr(A_i, A_j | \{|\mathsf{S}(\underline{y}_i) - \mathsf{S}(\underline{y}_j)| \geq L\}) \\
&\quad + \Pr(\{|\mathsf{S}(\underline{y}_i) - \mathsf{S}(\underline{y}_j)| < L\}) \\
&\leq \Pr(A_i)\Pr(A_j) + \frac{L}{n} \\
&= \left(1 - \frac{L}{n}\right)^{2(K-1)} + \frac{L}{n}.
\end{aligned} \tag{B.6}$$

Using (B.4), (B.5) and (B.6) in (B.3), we get

$$\begin{aligned}
\mathrm{Var}(K') &\leq \mathbb{E}[K'] + \sum_{i,j\in[K]:i\neq j} \frac{L}{n} \\
&\leq \mathbb{E}[K'] + K^2\frac{L}{n} &\text{(B.7)} \\
&\overset{(a)}{\leq} K + cK = (1+c)K, &\text{(B.8)}
\end{aligned}$$

where (a) holds as $K' \leq K$, by definition. Using (B.2) and (B.8) in (B.1),

$$\begin{aligned}\Pr\big(\big|K' - Ke^{-c}\big| \geq \epsilon K e^{-c}\big) &\leq \Pr\Big(\big|K' - \mathbb{E}[K']\big| \geq \Big(\frac{\epsilon}{2}\Big) K e^{-c}\Big) \\ &\leq \frac{\mathrm{Var}(K')}{\big(\frac{\epsilon}{2}\big)^2 (Ke^{-c})^2} \leq \frac{(1+c)K}{\big(\frac{\epsilon}{2}\big)^2 (Ke^{-c})^2} \\ &= \Theta\bigg(\frac{\log n}{n}\bigg) \to 0,\end{aligned}$$

as $n \to \infty$.

Appendix C

Proof of (3.17) (bound for $\frac{1}{n}\log|\mathsf{CI}|$

We start with a simple upper bound on $|\mathsf{CI}|$, following steps 4-11 of Algorithm 1.

$$|\mathsf{CI}| \leq \sum_{\underline{\omega}\in\Omega} (\text{number of read-orderings } \zeta \text{ compatible with } \underline{\omega}),$$

where an ordering ζ is said to be compatible with a suffix-size tuple $\underline{\omega} = (\omega_1, \ldots, \omega_K)$ if each read $\underline{y}_{\zeta(i)}$ is mergeable with its successor $\underline{y}_{\zeta(i+1)}$ with the specified merging suffix-size ω_i, $\forall i \in [K]$.

For any $\underline{\omega} \in \Omega$, we now provide the intuition for counting the number of compatible orderings. Consider that an arbitrary read $\underline{y}$ is selected as the first read. Note that, due to the presence of erasures, there may be multiple potential merging suffixes with size ω_1 in this specific read $\underline{y}$. A trivial upper bound for the number of such possible merging suffixes is $\ell(\underline{y}) = \bar{L}\log n$. Now, suppose we pick a particular merging suffix, $\underline{z}$, such that $\ell_{ue}(\underline{z}) = \omega_1 = \tau \log n$, where $\tau = \omega_1/\log n$.

We know that, for a given $\underline{z}$, $M_{\underline{z}}$ represents the number of reads which are $\ell(\underline{z})$-compatible with $\underline{z}$. In other words, $M_{\underline{z}}$ gives the number reads which are mergeable with the read $\underline{y}$ with $\underline{z}$ as merging suffix. Thus, there are at most $M_{\underline{z}}$ possible successors for $\underline{y}$, such that the compatibility with $\underline{\omega}$ is maintained. Note that $M_{\underline{z}} \leq b_3(\tau)$ (due to the assumption that the event $\bar{B}$ occurs). Once the size of the merging suffix and the successor to the first read are fixed, similar counting arguments hold the second read's merge with its successor. Note that the expected number of times $\tau \log n$ appears in $\underline{\omega} \in \Omega$ is exactly $G(\tau)$, and $G(\tau) \leq b_4(\tau)$ by the definition of Ω. Also, we observe that, since the ordering and merging process are cyclical, only those orderings where the last read is a valid predecessor of the first read, as per the merge given by the suffix-size tuple $\underline{\omega}$, are allowed. Thus, every pick where the successor of the last read is not the first read is not considered in the counting.

To summarise, for a fixed τ, the number of possible ways of merging a read, choosing some successor and some suffix with size $\tau \log n$, is upper bounded by $\bar{L}\log n \cdot b_3(\tau)$. Such mergings can occur for $G(\tau)$ reads among the K reads. Using this, we get

$$|\mathsf{CI}| \leq \sum_{\underline{\omega}\in\Omega} \prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{b_4(\tau)}$$

$$
\begin{aligned}
&\leq (L+1)^K \cdot \prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{b_4(\tau)} \\
&= (L+1)^K \cdot \prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{\bar{G}(\tau)+\epsilon\frac{n}{\log^2 n}} \\
&= (L+1)^K \cdot \prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{\bar{G}(\tau)} \\
&\qquad\qquad \cdot \prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{\epsilon\frac{n}{\log^2 n}}.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}\log|\mathsf{CI}| \\
&\quad\leq \lim_{n\to\infty}\frac{1}{n}\log\left((L+1)^K \cdot \prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{\bar{G}(\tau)}\right) \\
&\quad+ \lim_{n\to\infty}\frac{1}{n}\log\left(\prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{\epsilon\frac{n}{\log^2 n}}\right).
\end{aligned}
$$

Now,

$$
\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}\log\left(\prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))^{\epsilon\frac{n}{\log^2 n}}\right) \\
&\quad= \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\log\left(\prod_{\tau\in\mathcal{T}} (\bar{L}\log n \cdot b_3(\tau))\right) \\
&\quad= \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\log\left(\prod_{\tau\in\mathcal{T}} \bar{L}\log n\right) \\
&\qquad+ \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\log\left(\prod_{\tau\in\mathcal{T}} b_3(\tau)\right) \\
&\quad= \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\sum_{\tau\in\mathcal{T}}\log\left(\bar{L}\log n\right) \\
&\qquad+ \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\sum_{\tau\in\mathcal{T}}\log b_3(\tau) \\
&\quad\overset{(a)}{\leq} \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\sum_{\tau\in\mathcal{T}}\log\left(\bar{L}\log n\right) \\
&\qquad+ \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}\sum_{\tau\in\mathcal{T}}\log n \\
&\quad\overset{(b)}{\leq} \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}(\bar{L}\log n+1)\log\left(\bar{L}\log n\right) \\
&\qquad+ \lim_{n\to\infty}\frac{\epsilon}{\log^2 n}(\bar{L}\log n+1)\log n \\
&\quad= 0+\epsilon\bar{L}.
\end{aligned}
$$

Here, (a) holds as $b_3(\tau) \leq n, \forall \tau \in \mathcal{T}$ and (b) is due to $|\mathcal{T}| = \bar{L} \log n + 1$. Thus, as $\epsilon \to 0$, the value of this term goes to 0.

Hence, we have,

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{1}{n} \log |\mathsf{CI}| \\
&\quad \leq \lim_{n\to\infty} \frac{1}{n} \log \left((L+1)^K \cdot \prod_{\tau \in \mathcal{T}} (\bar{L} \log n \cdot b_3(\tau))^{\bar{G}(\tau)} \right) \\
&\quad = \lim_{n\to\infty} \frac{K}{n} \log (L+1) + \lim_{n\to\infty} \frac{1}{n} \log \left(\prod_{\tau \in \mathcal{T}} (\bar{L} \log n)^{\bar{G}(\tau)} \right) \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log \left(\prod_{\tau \in \mathcal{T}} b_3(\tau)^{\bar{G}(\tau)} \right) \\
&\quad = \lim_{n\to\infty} \frac{c}{\bar{L} \log n} \log \left(\bar{L} \log n + 1 \right) \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log (\bar{L} \log n) \sum_{\tau \in \mathcal{T}} \bar{G}(\tau) \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log \left(\prod_{\tau > (1-\epsilon)} (n^{\epsilon})^{\bar{G}(\tau)} \right) \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log \left(\prod_{\tau \leq (1-\epsilon)} \left(n^{1-\tau}\right)^{\bar{G}(\tau)} \right) \\
&\quad = 0 + \lim_{n\to\infty} \frac{K}{n} \log (\bar{L} \log n) \\
&\qquad + \lim_{n\to\infty} \frac{\epsilon}{n} \log n \sum_{\tau > 1-\epsilon} \bar{G}(\tau) \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log n \sum_{\tau \leq 1-\epsilon} (1-\tau) \bar{G}(\tau) \\
&\quad \leq \lim_{n\to\infty} \frac{c}{\bar{L} \log n} \log (\bar{L} \log n) \\
&\qquad + \lim_{n\to\infty} \frac{\epsilon}{n} \log n \cdot K \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log n \sum_{\tau \leq 1-\epsilon} (1-\tau) \bar{G}(\tau) \\
&\quad = 0 + \frac{\epsilon c}{\bar{L}} \\
&\qquad + \lim_{n\to\infty} \frac{1}{n} \log n \sum_{\tau \leq 1-\epsilon} (1-\tau) \bar{G}(\tau),
\end{aligned}
$$

Thus, as $\epsilon \to 0$,

$$
\lim_{n\to\infty} \frac{1}{n} \log |\mathsf{CI}| \leq \lim_{n\to\infty} \frac{1}{n} \log n \sum_{\tau \leq 1-\epsilon} (1-\tau) \bar{G}(\tau)
$$

The following claim gives an upper bound for the quantity $\lim_{n\to\infty} \frac{\log n}{n} \sum_{\tau\leq 1-\epsilon}(1-\tau)G(\tau)$. Using this completes the proof.

Claim 1. *Let $\bar{G}(\tau)$ denotes the expectation of $G(\tau)$, and $\alpha = c/(\bar{L}(1-\delta))$. The following statement is true.*

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\log n}{n} \sum_{\tau=0}^{1} (1-\tau)\bar{G}(\tau) \\
&\quad \leq \frac{(1+2p)d}{(1-\delta)} \cdot \Big(\frac{c}{\bar{L}}\Big)^2 \cdot e^{\alpha p d - c} \cdot \\
&\qquad \Bigg(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)} \\
&\qquad - \frac{e^{2\alpha d}\big(\big(e^{\alpha(1+d)} - e^{\alpha}\big) - d\big(e^{\alpha(1+d)}-1\big)\big)}{(e^{\alpha d}-1)^2} \Bigg),
\end{aligned} \tag{C.1}$$

for any $p > 0$ and any $d > 0$.

Proof. Recall that $\underline{\omega}^t = (\omega_1^t, \ldots, \omega_K^t)$ denotes the true sizes of the merging suffixes of the reads, taken in the true ordering ζ^t. Thus, ω_i^t refers to the true suffix-size of the i^{th} read $\underline{y}_{\zeta^t(i)}$. Note that $\omega_i^t : i \in [K]$ are random variables, depending on the length of the overlapping region, as well as the erasure pattern in the overlapping region. Let L_i denote the length of the overlapping region of read $\underline{y}_{\zeta^t(i)}$ with read $\underline{y}_{\zeta^t(i+1)}$. Let $\bar{\omega}_i^t$ denote the expectation of ω_i^t conditioned on L_i, over the randomness of the erasures. Observe that $\bar{\omega}_i^t = L_i(1-\delta)$. Also, we recall that $L_i \leq L$.

We split the interval $[0,1)$ into subintervals of size d for some $d > 0$. Thus, there are $1/d$ intervals. For $k \in \{0, 1, \ldots, 1/d - 1\}$, the k^{th} interval is denoted as $[kd, (k+1)d)$. Let $\bar{G}([kd,(k+1)d))$ denote the expectation of the number of reads which have suffix-size between kd and $(k+1)d$, i.e., $\bar{G}([kd,(k+1)d)) \triangleq \mathbb{E}[\sum_{i\in[K]} \mathbb{I}_{\omega_i^t \in [kd\log n, (k+1)d\log n)}]$. Thus, $\bar{G}([kd,(k+1)d)) \leq \sum_{\tau=kd}^{(k+1)d} \bar{G}(\tau)$. Thus, we can write

$$\sum_{\tau=0}^{1}(1-\tau)\bar{G}(\tau) \leq \sum_{k=0}^{1/d-1} (1-kd)\bar{G}([kd,(k+1)d)). \tag{C.2}$$

We now consider $\bar{G}([kd,(k+1)d))$. We can write the following inequalities. Without loss of generality, we assume that the true ordering starts with read $\underline{y}_1$ (i.e., $\zeta^t(1) = 1$), and $\underline{y}_1$ starts at the first position, i.e., $\mathsf{S}(\underline{y}_1) = 1$. Let A_k denote the event that the $\underline{y}_1$ is mergeable with read $\underline{y}_2$ as its successor, with suffix size $\omega_1^t \in [kd, (k+1)d)$. Let D_j denote the event that $\mathsf{S}(\underline{y}_j) \geq \mathsf{S}(\underline{y}_2)$. Thus, by the definition

of $\bar{G}([kd,(k+1)d))$ and because $\omega_i^t : i \in [K]$ are identically distributed, we have

$$\begin{aligned}
&\bar{G}([kd,(k+1)d)) \\
&\quad = K \Pr\Big(\underline{y}_1 \text{ is mergeable with } \underline{y}_{j'} \text{ with suffix-size} \\
&\qquad \omega_1^t \in [kd,(k+1)d), \text{ for some } j' \in [K]\setminus\{1\}, \\
&\qquad \text{s.t } \Big\{\mathsf{S}(\underline{y}_{j'}) \le \mathsf{S}(\underline{y}_j), \forall j \in [K]\setminus\{1,j'\}\Big\}\Big) \\
&\quad = K(K-1)\Pr(A_k, \{D_j : j \ge 3\}). \qquad \text{(C.3)}
\end{aligned}$$

We recall that $\Pr(\mathsf{S}(\underline{y}_j) = s) = 1/n$, for any $s \in [n]$. Since we assumed that $\mathsf{S}(\underline{y}_1) = 1$, thus we see that $\mathsf{S}(\underline{y}_2) = L - L_1 + 1 = L - \bar{\omega}_1^t/(1-\delta) + 1$. Thus, for any $1 \le \mathsf{w} \le L$, then $\Pr(\bar{\omega}_1^t = \mathsf{w}) = \Pr(\mathsf{S}(\underline{y}_2) = L - \mathsf{w}/(1-\delta) + 1) = 1/n$. We now intend to bound $\Pr(A_k, \{D_j : j \ge 3\})$, as $n \to \infty$, using the fact that if $\bar{\omega}_1^t$ is concentrated in a small interval, then so is ω_1^t.

Consider some small $p > 0$ such that $(1+pd)\log n \le L(1-\delta)$ (such a p exists, as $\bar{L} > 1/(1-\delta)$). We define the following intervals, $1/d$ of them.

$$I_k = \begin{cases} [(k-p)d, (k+1+p)d], & \text{if } 1 \le k \le 1/d - 1 \\ [0,(1+p)d], & \text{if } k = 0. \end{cases} \qquad \text{(C.4)}$$

Let C_k denote the event that $\bar{\omega}_1^t \in I_k$. We have that,

$$\begin{aligned}
\Pr(A_k, \{D_j : j \ge 3\}) = &\Pr(A_k, \{D_j : j \ge 3\}, C_k) \\
&+ \Pr(A_k, \{D_j : j \ge 3\}, \overline{C_k}). \qquad \text{(C.5)}
\end{aligned}$$

We now show that the term $\Pr(A_k, \{D_j : j \ge 3\}, \overline{C_k})$ is $\mathsf{O}(\log n/n^2)$. We do this in two parts.

For $k \ge 1$, we can write

$$\begin{aligned}
&\Pr(A_k, \{D_j : j \ge 3\}, \{\bar{\omega}_1^t < (k-p)d\log n\}) \\
&= \sum_{\mathsf{w} < (k-p)d\log n} \Pr(A_k, \{D_j : j \ge 3\}, \bar{\omega}_1^t = \mathsf{w}) \\
&\le \sum_{\mathsf{w} < (k-p)d\log n} \Pr(A_k | \bar{\omega}_1^t = \mathsf{w}) \Pr(\bar{\omega}_1^t = \mathsf{w}). \qquad \text{(C.6)}
\end{aligned}$$

Now, we can use Hoeffding's inequality[1] to bound the quantity $\Pr(A_k|\bar{\omega}_1^t = \mathsf{w})$. To see this, observe that when $\bar{\omega}_1^t = \mathsf{w}$, the suffix-size ω_1^t of the merge of $\underline{y}_1$ and $\underline{y}_2$ is the sum of $L_1 = \mathsf{w}/(1-\delta)$ independent

[1]See [46]. The inequality is as follows. For S being the sum of n independent Boolean random variables and any $t > 0$, $\Pr(S - \mathbb{E}(S) \ge t) \le 2\mathsf{e}^{-2t^2/n}$

Boolean indicator random variables (1 indicating erasure, 0 indicating no-erasure). Therefore, we get, for $\mathsf{w} < (k-p)d\log n$

$$\begin{aligned}
&\Pr(A_k|\bar{\omega}_1^t = \mathsf{w}) \\
&\leq \Pr(\omega_1^t \geq kd\log n|\bar{\omega}_1^t = \mathsf{w}) \\
&= \Pr(\omega_1^t \geq \bar{\omega}_1^t + (kd\log n - \bar{\omega}_1^t)|\bar{\omega}_1^t = \mathsf{w}) \\
&\overset{(a)}{\leq} 2\mathsf{e}^{-2(kd\log n - \mathsf{w})^2(1-\delta)/\mathsf{w}} \\
&\overset{(b)}{\leq} 2\mathsf{e}^{-2(pd\log n)^2(1-\delta)/((k-p)d\log n)} = 2\mathsf{e}^{-2p^2(1-\delta)d\log n/(k-p)} \\
&= \Theta(1/n),
\end{aligned}$$

where (a) holds by the Hoeffding's inequality, and (b) holds as $\mathsf{w} < (k-p)d\log n$. Using this in (C.6), we get

$$\begin{aligned}
&\Pr(A_k, \{D_j : j \geq 3\}, \{\bar{\omega}_1^t < (k-p)d\log n\}) \\
&\quad\overset{(a)}{\leq} \frac{(k-p)d\log n}{(1-\delta)} \cdot \frac{1}{n} \cdot 2\mathsf{e}^{-p^2(1-\delta)d\log n/(k-p)} \\
&\quad= \Theta\bigg(\frac{\log n}{n^2}\bigg),
\end{aligned} \tag{C.7}$$

where (a) holds because $\bar{\omega}_1^t$ takes values in steps of $(1-\delta)$, (as $\bar{\omega}_1^t = L_1(1-\delta)$ where L_1 takes values in unit steps).

Using similar arguments, we can show the following for all $k \in \{0, \ldots, 1/d - 1\}$.

$$\begin{aligned}
&\Pr(A_k, \{D_j : j \geq 3\}, \{\bar{\omega}_1^t \geq (k+1+p)d\log n\}) \\
&\quad= \sum_{\mathsf{w} \geq (k+1+p)d\log n} \Pr(A_k, \{D_j : j \geq 3\}, \bar{\omega}_1^t = \mathsf{w}) \\
&\quad= \sum_{\mathsf{w} \geq (k+1+p)d\log n} \Pr(A_k, \{D_j : j \geq 3\}|\bar{\omega}_1^t = \mathsf{w}) \Pr(\bar{\omega}_1^t = \mathsf{w}) \\
&\quad\leq \sum_{\mathsf{w} \geq (k+1+p)d\log n} \Pr(A_k|\bar{\omega}_1^t = \mathsf{w}) \Pr(\bar{\omega}_1^t = \mathsf{w}) \\
&\quad\leq \sum_{\mathsf{w} \geq (k+1+p)d\log n} \Big(\Pr(\omega_1^t < (k+1)d\log n|\bar{\omega}_1^t = \mathsf{w}) \\
&\qquad\qquad\qquad\qquad \cdot \Pr(\bar{\omega}_1^t = \mathsf{w}) \Big) \\
&\quad\leq \sum_{\mathsf{w} \geq (k+1+p)d\log n} \Big(\Pr(\omega_1^t < \bar{\omega}_1^t - (\bar{\omega}_1^t - (k+1)d\log n)
\end{aligned}$$

$$
\begin{aligned}
&\qquad\qquad\qquad\qquad\qquad\qquad \mid \bar{\omega}_1^t = \mathsf{w}) \Pr(\bar{\omega}_1^t = \mathsf{w})\Big) \\
&\leq \sum_{\mathsf{w} \geq (k+1+p) d \log n} 2\mathsf{e}^{-2(\mathsf{w}-(k+1)d\log n)^2(1-\delta)/\mathsf{w}} \Pr(\bar{\omega}_1^t = \mathsf{w}) \\
&\leq \sum_{\mathsf{w} \geq (k+1+p) d \log n} 2\mathsf{e}^{-2(pd\log n)^2(1-\delta)/((k+1+p)d\log n)} \Pr(\bar{\omega}_1^t = \mathsf{w}) \\
&\leq 2\mathsf{e}^{-2p^2 d\log n(1-\delta)/(k+1+p)} \left(\sum_{\mathsf{w} \geq (k+1+p) d \log n} \Pr(\bar{\omega}_1^t = \mathsf{w}) \right) \\
&= \Theta\Big(\frac{\log n}{n^2} \Big). \qquad \text{(C.8)}
\end{aligned}
$$

Using (C.7) and (C.8), we see that

$$
\begin{aligned}
&\Pr(A_k, \{D_j : j \geq 3\}, \overline{C_k}) \\
&= \Pr(A_k, \{D_j : j \geq 3\}, \{\bar{\omega}_1^t < (k-p) d \log n\}) \\
&\quad + \Pr(A_k, \{D_j : j \geq 3\}, \{\bar{\omega}_1^t \geq (k+1+p) d \log n\}) \\
&= \mathsf{O}\Big(\frac{\log n}{n^2} \Big). \qquad \text{(C.9)}
\end{aligned}
$$

Now, starting with the first term in the R.H.S. of (C.5), we have

$$
\begin{aligned}
&\Pr(A_k, \{D_j : j \geq 3\}, C_k) \\
&\leq \Pr(\{D_j : j \geq 3\}, \bar{\omega}_1^t \in I_k) \qquad \text{(C.10)} \\
&= \sum_{\mathsf{w} \in I_k} \Pr(\{D_j : j \geq 3\} | \bar{\omega}_1^t = \mathsf{w}) \Pr(\bar{\omega}_1^t = \mathsf{w}) \\
&= \frac{1}{n} \cdot \sum_{\mathsf{w} \in I_k} \Pr(\{D_j : j \geq 3\} | \bar{\omega}_1^t = \mathsf{w}) \\
&= \frac{1}{n} \cdot \sum_{\mathsf{w} \in I_k} \Pr(\{D_j : j \geq 3\} | \mathsf{S}(\underline{y}_2) = L - \mathsf{w}/(1-\delta) + 1) \\
&= \frac{1}{n} \cdot \sum_{\mathsf{w} \in I_k} \left(1 - \frac{L - \mathsf{w}/(1-\delta)}{n} \right)^{K-2}. \qquad \text{(C.11)}
\end{aligned}
$$

Using (C.2), (C.3), (C.5), (C.9) and (C.11), we get

$$
\begin{aligned}
&\frac{\log n}{n} \sum_{\tau=0}^{1} (1-\tau) \bar{G}(\tau) \\
&\qquad \leq \frac{\log n}{n} \cdot (K(K-1)) \Big(\sum_{k=0}^{1/d-1} (1-kd) \cdot
\end{aligned}
$$

$$
\begin{aligned}
&\frac{1}{n}\cdot\sum_{\mathsf{w}\in I_k}\left(1-\frac{L-\mathsf{w}/(1-\delta)}{n}\right)^{K-2}\Big)\\
&\quad+\frac{\log n\cdot K(K-1)}{n}\Pr(A_k,\{D_j:j\geq 3\},\overline{C_k}))\\
&\leq\frac{\log n}{n^2}\cdot(K(K-1))\cdot\\
&\left(\sum_{k=0}^{1/d-1}(1-kd)\cdot\left(\sum_{\mathsf{w}\in I_k}\left(1-\frac{L-\mathsf{w}/(1-\delta)}{n}\right)^{K-2}\right)\right)\\
&\quad+\mathsf{O}\left(\frac{1}{n}\right)\\
&\leq\frac{\log n}{n^2}\cdot(K(K-1))\cdot\\
&\left[\sum_{k=0}^{1/d-1}(1-kd)\cdot\right.\\
&\left.\left(\sum_{\mathsf{w}\in I_k}\left(1-\frac{(\bar{L}-(k+1+p)d/(1-\delta))\log n}{n}\right)^{K-2}\right)\right]\\
&\quad+\mathsf{O}\left(\frac{1}{n}\right)\\
&\leq\frac{\log n}{n^2}\cdot\left(\frac{cn}{\bar{L}\log n}\right)^2\left[\sum_{k=0}^{1/d-1}(1-kd)\cdot\right.\\
&\left.\cdot\left(\sum_{\mathsf{w}\in I_k}\left(1-\frac{(\bar{L}-(k+1+p)d/(1-\delta))\log n}{n}\right)^{K-2}\right)\right]\\
&\quad+\mathsf{O}\left(\frac{1}{n}\right)\\
&\overset{(a)}{\leq}\frac{(1+2p)d\log n}{(1-\delta)}\cdot\frac{\log n}{n^2}\cdot\left(\frac{cn}{\bar{L}\log n}\right)^2\\
&\left[\sum_{k=0}^{1/d-1}(1-kd)\cdot\right.\\
&\left.\left(1-\frac{(\bar{L}-(k+1+p)d/(1-\delta))\log n}{n}\right)^{K-2}\right]\\
&\quad+\mathsf{O}\left(\frac{1}{n}\right)\\
&\leq\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot\\
&\left[\sum_{k=0}^{1/d-1}(1-kd)\cdot\right.
\end{aligned}
$$

$$\left(1-\frac{(\bar{L}-(k+1+p)d/(1-\delta))\log n}{n}\right)^{K-2}\Bigg]$$
$$+\mathsf{O}\left(\frac{1}{n}\right)$$
$$\leq \frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot$$
$$\Bigg[\sum_{k=0}^{1/d-1}(1-kd)\cdot$$
$$\left(1-\frac{(\bar{L}-(k+1+p)d/(1-\delta))\log n}{n}\right)^{\frac{cn}{\bar{\bar{L}}\log n}-2}\Bigg]$$
$$+\mathsf{O}\left(\frac{1}{n}\right)$$

where (a) due to the size of interval I_k.

Now as $n\to\infty$, the above value goes to

$$\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot\sum_{k=0}^{1/d-1}(1-kd)e^{\frac{-c(\bar{L}(1-\delta)-(k+1+p)d)}{\bar{L}(1-\delta)}}$$
$$=\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot e^{-c}\cdot\sum_{k=0}^{1/d-1}(1-kd)e^{\frac{c((k+1+p)d)}{\bar{L}(1-\delta)}}$$

Taking $\alpha=\frac{c}{\bar{L}(1-\delta)}$, we get

$$\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot e^{\alpha pd-c}\cdot\sum_{k=0}^{1/d-1}(1-kd)e^{\alpha(k+1)d}$$
$$=\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot e^{\alpha pd-c}\cdot$$
$$\left(\sum_{k=0}^{1/d-1}e^{\alpha(k+1)d}-\sum_{k=0}^{1/d-1}kde^{\alpha(k+1)d}\right)$$
$$=\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot e^{\alpha pd-c}\cdot$$
$$\cdot\left(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)}-de^{\alpha d}\sum_{k=0}^{1/d-1}ke^{\alpha kd}\right)$$
$$\overset{(a)}{=}\frac{(1+2p)d}{(1-\delta)}\cdot\left(\frac{c}{\bar{\bar{L}}}\right)^2\cdot e^{\alpha pd-c}\cdot$$
$$\Bigg(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)}$$
$$-\frac{e^{2\alpha d}\big(\big(e^{\alpha(1+d)}-e^{\alpha}\big)-d\big(e^{\alpha(1+d)}-1\big)\big)}{(e^{\alpha d}-1)^2}\Bigg),$$

where (a) holds because $\sum_{i=0}^{N} iq^i = \sum_{i=1}^{N}\sum_{j=i}^{N} q^j = N\frac{q^{N+1}}{q-1} - \frac{q^{n+1}-1}{(q-1)^2}$, for any $q \neq 0$. $\square$

Appendix D

Proof of (3.19)(expression for $\lim_{d\to 0}\beta(d)$

Recall that

$$\begin{aligned}\beta(d) &= \frac{d}{(1-\delta)}\cdot\Big(\frac{c}{\bar{\bar{L}}}\Big)^2\cdot e^{-c}\cdot\left(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)}-\frac{e^{2\alpha d}\big(\big(e^{\alpha(1+d)}-e^{\alpha}\big)-d\big(e^{\alpha(1+d)}-1\big)\big)}{(e^{\alpha d}-1)^2}\right)\\ &= \left(\frac{1}{(1-\delta)}\cdot\Big(\frac{c}{\bar{\bar{L}}}\Big)^2\cdot e^{-c}\right)\cdot d\left(\frac{e^{\alpha d}(e^{\alpha}-1)}{(e^{\alpha d}-1)}-\frac{e^{2\alpha d}\big(\big(e^{\alpha(1+d)}-e^{\alpha}\big)-d\big(e^{\alpha(1+d)}-1\big)\big)}{(e^{\alpha d}-1)^2}\right)\\ &= \big((1-\delta)\alpha^2e^{-c}\big)\cdot\left(\frac{d.e^{\alpha d}(e^{\alpha}-1)(e^{\alpha d}-1)+d^2e^{2\alpha d}(e^{\alpha(1+d)}-1)-de^{2\alpha d}(e^{\alpha(1+d)}-e^{\alpha})}{(e^{\alpha d}-1)^2}\right),\end{aligned}$$

where $\alpha=\frac{c}{\bar{L}(1-\delta)}$.

We start by observing that $\beta(d)$ at $d=0$ gives an indeterminate $\big(\frac{0}{0}\big)$ form, and thus we can apply L'Hôpital's rule twice to obtain the limit [1].

Now, we can re-express $\beta(d)$ as,

$$\beta(d)=\gamma\cdot\frac{\beta_1+\beta_2+\beta_3}{\beta_4}, \tag{D.1}$$

where,

$$\begin{aligned}\gamma &= (1-\delta)\alpha^2e^{-c}\\ \beta_1 &= d.e^{\alpha d}(e^{\alpha}-1)(e^{\alpha d}-1)\\ \beta_2 &= d^2(e^{\alpha(1+d)}-1)e^{2\alpha d}\\ \beta_3 &= -de^{2\alpha d}(e^{\alpha(1+d)}-e^{\alpha})\\ \beta_4 &= (e^{\alpha d}-1)^2.\end{aligned}$$

Now,

$$\begin{aligned}\frac{\partial\beta_1}{\partial d} &= (e^{\alpha}-1)(e^{\alpha d}(e^{\alpha d}-1)+\alpha de^{2\alpha d}+\alpha de^{\alpha d}(e^{\alpha d}-1))\\ &= (e^{\alpha}-1)(e^{2\alpha d}(1+2\alpha d)-e^{\alpha d}(1+\alpha d)),\end{aligned}$$

[1] By observation, it was found that the expression remains in the indeterminate $\big(\frac{0}{0}\big)$ form after applying L'Hôpital's rule once, i.e., after differentiating both numerator and denominator once. Only on applying L'Hôpital's rule twice, i.e., after double-differentiating both numerator and denominator, a determinate expression is obtained.

and,

$$\frac{\partial^2 \beta_1}{\partial d^2} = (e^\alpha - 1)(2\alpha e^{2\alpha d}(1 + 2\alpha d) + 2\alpha e^{2\alpha d} - \alpha e^{\alpha d} - \alpha e^{\alpha d}(1 + \alpha d)).$$

Hence,

$$\lim_{d \to 0} \frac{\partial^2 \beta_1}{\partial d^2} = (e^\alpha - 1)(2\alpha + 2\alpha - \alpha - \alpha) = 2\alpha(e^\alpha - 1). \tag{D.2}$$

Further,

$$\begin{aligned}\frac{\partial \beta_2}{\partial d} &= 2d(e^{\alpha(1+d)} - 1)e^{2\alpha d} + \alpha d^2 e^{\alpha(1+d)} e^{2\alpha d} + 2\alpha d^2 (e^{\alpha(1+d)} - 1)e^{2\alpha d} \\ &= de^{2\alpha d}(e^{\alpha(1+d)}(2 + 3\alpha d) - 2(1 + \alpha d)),\end{aligned}$$

and,

$$\frac{\partial^2 \beta_2}{\partial d^2} = e^{2\alpha d}(e^{\alpha(1+d)}(2 + 3\alpha d) - 2(1 + \alpha d)) + d \cdot \frac{\partial}{\partial d}(e^{2\alpha d}(e^{\alpha(1+d)}(2 + 3\alpha d) - 2(1 + \alpha d))).$$

Hence,

$$\lim_{d \to 0} \frac{\partial^2 \beta_2}{\partial d^2} = 2(e^\alpha - 1). \tag{D.3}$$

Furthermore,

$$\begin{aligned}\frac{\partial \beta_3}{\partial d} &= -(e^{2\alpha d}(e^{\alpha(1+d)} - e^\alpha) + 2\alpha d e^{2\alpha d}(e^{\alpha(1+d)} - e^\alpha) + \alpha d e^{\alpha(1+3d)}) \\ &= -(e^{2\alpha d}(e^{\alpha(1+d)} - e^\alpha)(1 + 2\alpha d) + \alpha d e^{\alpha(1+3d)}),\end{aligned}$$

and,

$$\begin{aligned}\frac{\partial^2 \beta_3}{\partial d^2} = -(&2\alpha e^{2\alpha d}(e^{\alpha(1+d)} - e^\alpha)(1 + 2\alpha d) + \alpha e^{\alpha(1+3d)}(1 + 2d) \\ &+ 2\alpha e^{2\alpha d}(e^{\alpha(1+d)} - e^\alpha) + \alpha e^{\alpha(1+3d)} + 3\alpha^2 d e^{\alpha(1+3d)}).\end{aligned}$$

Hence,

$$\lim_{d \to 0} \frac{\partial^2 \beta_3}{\partial d^2} = -(\alpha e^\alpha + \alpha e^\alpha) = -2\alpha e^\alpha. \tag{D.4}$$

Also,

$$\frac{\partial \beta_4}{\partial d} = 2\alpha(e^{\alpha d} - 1)e^{\alpha d},$$

and,

$$\frac{\partial^2 \beta_4}{\partial d^2} = 2\alpha^2 e^{2\alpha d} + 2\alpha^2 (e^{\alpha d} - 1)e^{\alpha d}.$$

Hence,

$$\lim_{d\to 0} \frac{\partial^2 \beta_4}{\partial d^2} = 2\alpha^2. \tag{D.5}$$

Thus, using (D.2), (D.3), (D.4), and (D.5) in (D.1),

$$\begin{aligned}
\lim_{d\to 0} \beta(d) &= (1-\delta)\alpha^2 e^{-c} \cdot \left(\frac{2\alpha(e^\alpha - 1) + 2(e^\alpha - 1) - 2\alpha e^\alpha}{2\alpha^2} \right) \\
&= (1-\delta)e^{-c}(e^\alpha - (1+\alpha)) \\
&= (1-\delta)e^{-c}\left(e^{\frac{c}{\bar{L}(1-\delta)}} - \left(1 + \frac{c}{\bar{L}(1-\delta)}\right)\right) \\
&= (1-\delta)(e^{-c\left(1-\frac{1}{\bar{L}(1-\delta)}\right)} - e^{-c}) + \frac{c}{\bar{L}} e^{-c}.
\end{aligned}$$

Related Publications

- H. Narayanan, P. Krishnan, and N. Parekh, "On achievable rates for the shotgun sequencing channel with erasures", Accepted for presentation at the 2024 IEEE International Symposium on Information Theory (ISIT), to be held at Athens, Greece during July 7-12, 2024. (Full version available on ArXiv at `https://arxiv.org/abs/2401.16342`).

Bibliography

[1] A. N. Ravi, A. Vahid, and I. Shomorony, "Coded shotgun sequencing," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 1, pp. 147–159, 2022.

[2] K. Weide-Zaage, "DNA digital-storage: Advantages, approach and technical implementation," in *2024 Pan Pacific Strategic Electronics Symposium (Pan Pacific)*, 2024, pp. 1–6.

[3] R. Kim, L. Pschetz, C. Linehan, C. H. Lee, and S. Poslad, "Archives in DNA: Workshop exploring implications of an emerging bio-digital technology through design fiction," in *Proceedings of the 24th International Academic Mindtrek Conference*, ser. Academic Mindtrek '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 102–105. [Online]. Available: https://doi.org/10.1145/3464327.3464966

[4] C. Ezekannagha, A. Becker, D. Heider, and G. Hattab, "Design considerations for advancing data storage with synthetic DNA for long-term archiving," *Materials Today Bio*, vol. 15, p. 100306, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590006422001041

[5] E. Green, "Shotgun sequencing," 2024. [Online]. Available: https://www.genome.gov/genetics-glossary/Shotgun-Sequencing

[6] A. M. Nafea, Y. Wang, D. Wang, A. M. Salama, M. A. Aziz, S. Xu, and Y. Tong, "Application of next-generation sequencing to identify different pathogens," *Frontiers in Microbiology*, vol. 14, 2024. [Online]. Available: https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2023.1329330

[7] C. Baum, "New approaches and concepts to study complex microbial communities," Theses, Université Paris-Saclay ; New England Biolabs France, Oct. 2021. [Online]. Available: https://theses.hal.science/tel-03531325

[8] A. Beckett, K. Cook, and S. Robson, "A pandemic in the age of next-generation sequencing," *The Biochemist*, vol. 43, 12 2021.

[9] K. Matange, J. M. Tuck, and A. J. Keung, "DNA stability: a central design consideration for DNA data storage systems," *Nature Communications*, vol. 12, no. 1, p. 1358, Mar 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-21587-5

[10] A. S. Motahari, G. Bresler, and D. N. C. Tse, "Information theory of DNA shotgun sequencing," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6273–6289, 2013.

[11] R. P. Feynman, "There's plenty of room at the bottom," *Engineering and Sciences.*, vol. 23, no. 5, pp. 22–36, Feb. 1960.

[12] T. Hey, "Quantum computing: an introduction," *Computing & Control Engineering Journal*, vol. 10, no. 3, pp. 105–112, 1999.

[13] A. Gibbons, M. Amos, and D. Hodgson, "DNA computing," *Current Opinion in Biotechnology*, vol. 8, no. 1, pp. 103–106, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0958166997801644

[14] V. Balzani, A. Credi, F. Raymo, and J. Stoddart, "Artificial molecular machines," *Angewandte Chemie International Edition*, vol. 39, no. 19, pp. 3348–3391, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1521-3773%2820001002%2939%3A19%3C3348%3A%3AAID-ANIE3348%3E3.0.CO%3B2-X

[15] R. Keyes, "Physical limits in digital electronics," *Proceedings of the IEEE*, vol. 63, no. 5, pp. 740–767, 1975.

[16] Github, "Arctic world program: Our approach," 2024. [Online]. Available: https://archiveprogram.github.com/approach/#:~:text=Arctic%20world%20archive,will%20last%20twice%20as%20long

[17] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1226355

[18] D. Carmean, L. Ceze, G. Seelig, K. Stewart, K. Strauss, and M. Willsey, "DNA data storage and hybrid molecular–electronic computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 63–72, 2019.

[19] C. N. Takahashi, B. H. Nguyen, K. Strauss, and L. Ceze, "Demonstration of end-to-end automation of DNA data storage," *Scientific Reports*, vol. 9, no. 1, p. 4998, Mar 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-41228-8

[20] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angewandte*

Chemie International Edition, vol. 54, no. 8, pp. 2552–2555, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201411378

[21] G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinformatics*, vol. 14, no. 5, p. S18, Jul 2013. [Online]. Available: https://doi.org/10.1186/1471-2105-14-S5-S18

[22] I. Shomorony, S. H. Kim, T. A. Courtade, and D. N. C. Tse, "Information-optimal genome assembly via sparse read-overlap graphs," *Bioinformatics*, vol. 32, no. 17, pp. i494–i502, 08 2016. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw450

[23] S. Nassirpour, I. Shomorony, and A. Vahid, "Reassembly codes for the chop-and-shuffle channel," *arXiv preprint arXiv:2201.03590*, 2022.

[24] D. Bar-Lev, S. Marcovich, E. Yaakobi, and Y. Yehezkeally, "Adversarial torn-paper codes," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 2934–2939.

[25] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Coding over sets for DNA storage," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2331–2351, 2020.

[26] I. Shomorony, T. A. Courtade, and D. Tse, "Fundamental limits of genome assembly under an adversarial erasure model," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 2, pp. 199–208, 2016.

[27] K. Levick, R. Heckel, and I. Shomorony, "Achieving the capacity of a DNA storage channel with linear coding schemes," in *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, 2022, pp. 218–223.

[28] R. Heckel, I. Shomorony, K. Ramchandran, and D. N. C. Tse, "Fundamental limits of DNA storage systems," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 3130–3134.

[29] I. Shomorony and R. Heckel, "Capacity results for the noisy shuffling channel," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 762–766.

[30] I. Shomorony and A. Vahid, "Torn-paper coding," *IEEE Transactions on Information Theory*, vol. 67, no. 12, pp. 7904–7913, 2021.

[31] A. N. Ravi, A. Vahid, and I. Shomorony, "Capacity of the torn paper channel with lost pieces," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1937–1942.

[32] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "An upper bound on the capacity of the DNA storage channel," in *2019 IEEE Information Theory Workshop (ITW)*, 2019, pp. 1–5.

[33] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakohi, "Achieving the capacity of the DNA storage channel," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8846–8850.

[34] I. Shomorony and R. Heckel, "DNA-based storage: Models and fundamental limits," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3675–3689, 2021.

[35] Eurofins, "How to do NGS 50% faster – the NGS platforms," 2021. [Online]. Available: https://the-dna-universe.com/2021/11/25/how-to-do-ngs-50-faster-with-our-ngs-platforms

[36] O. Sabary, A. Yucovich, G. Shapira, and E. Yaakobi, "Reconstruction algorithms for DNA-storage systems," *Scientific Reports*, vol. 14, no. 1, p. 1951, Jan 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-51730-3

[37] M. Li, M. Nordborg, and L. M. Li, "Adjust quality scores from alignment and improve sequencing accuracy," *Nucleic Acids Res.*, vol. 32, no. 17, pp. 5183–5191, Sep. 2004.

[38] I. Shomorony and R. Heckel, "Information-theoretic foundations of DNA data storage," *Foundations and Trends® in Communications and Information Theory*, vol. 19, no. 1, pp. 1–106, 2022. [Online]. Available: http://dx.doi.org/10.1561/0100000117

[39] S. Shin, R. Heckel, and I. Shomorony, "Capacity of the erasure shuffling channel," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8841–8845.

[40] S. R. Srinivasavaradhan, S. Gopi, H. D. Pfister, and S. Yekhanin, "Trellis BMA: Coded trace reconstruction on IDS channels for DNA storage," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2453–2458.

[41] W. H. Press, J. A. Hawkins, S. K. Jones, J. M. Schaub, and I. J. Finkelstein, "Hedges error-correcting code for DNA storage corrects indels and allows sequence constraints," *Proceedings of the National Academy of Sciences*, vol. 117, no. 31, pp. 18 489–18 496, 2020. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2004821117

[42] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[43] E. S. Lander and M. S. Waterman, "Genomic mapping by fingerprinting random clones: A mathematical analysis," *Genomics*, vol. 2, no. 3, pp. 231–239, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0888754388900079

[44] N. Alon and J. H. Spencer, *The Probabilistic Method*, 4th ed., ser. Wiley Series in Discrete Mathematics and Optimization. Nashville, TN: John Wiley & Sons, Jan. 2016.

[45] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. [Online]. Available: https://doi.org/10.1093/acprof:oso/9780199535255.001.0001

[46] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830