Probabilistic Modeling and Analysis Methods for Large-scale Data Compression

Hao Lou

Department of Electrical and Computer Engineering
University of Virginia, Charlottesville, USA
haolou@virginia.edu

A dissertation presented to
the Faculty of the School of Engineering and Applied Sciences
at the
University of Virginia

In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
in
Electrical Engineering

Contents

Abstract

As a result of the rapid increase in the size of digital data, addressing current data storage needs and provisioning for future growth have become challenging tasks. Data stored in computing systems however is redundant, with blocks that are repeated one or more times, such as an image embedded in multiple documents. Data deduplication, which eliminates redundant data at the file or subfile level, has gained increasing attention and popularity in large-scale storage systems.

However, there are significant shortcomings in the current approaches to deduplication, which are mainly due to the lack of a mathematical framework and analytical methods. In particular, our knowledge of the fundamental limits of data deduplication is severely limited. Information-theoretic analysis of data deduplication has only been considered by a limited number of works. Moreover, in these works, deduplication algorithms are evaluated over information sources with known statistical properties. While in reality, given a particular data sequence, we rarely know the specific distribution of the source producing it. In this dissertation, we study data deduplication theoretically by developing and analyzing different models for information sources and various deduplication algorithms as well as their performance evaluated over source models. Moreover, we extend the study of data deduplication to the context of universal compression, where we assume only a family of distributions to which the source distribution belongs is known, by drawing equivalence between encoding chunks and the problem of encoding pattern sequences.

One of the significant contributors to the increase in the size of digital data is genomic sequencing data, which is growing at a rate much faster than the decrease in the cost of storage media due to development of high-throughput sequencing methods. Hence, developing compression methods tailored to genomic data can be effective in addressing data storage challenges. Developing and leveraging models for biological processes that generate this type of data are also crucial steps in this direction. Biological sequences are created over billions of years by mutation processes that shape their statistical properties. In this dissertation, we present a framework of modeling biological sequences as outcomes of such evolutionary processes driven by random mutations. The statistics k-mer frequencies are studied both asymptotically and in finite-time. Based on these findings, the entropy of evolutionary processes is bounded, thus providing an lower bound for compression tasks. Additionally, estimates for waiting times problems, with potential applications to study of diseases such as cancer, are derived.

I. Introduction

A. Motivation and Overview

The ubiquity of mobile and embedded devices with a multitude of sensors and the universal integration of computing devices in personal life and corporate workflows have substantially increased our ability to produce and collect data. The amount of data created or copied globally in 2020 is estimated to be around 60 Zettabytes (1 ZB = 10^{12} GB), and predicted to double every three years [88]. Storage of such vasts amount of data is costly and resource-intensive. For example, data centers alone account for about 1% of the world's electricity usage [68], which demonstrates the large scale of the problem. As a result of this 'data deluge', addressing current data storage needs and provisioning for future growth have become challenging tasks.

It is however estimated that only a tenth of digital data is unique and the rest are all duplicates. On a single desktop computer, the amount of unique data is found to be between 1/10 and 1/3 of the total amount of data [70]. Data deduplication, which is part of the focus in this dissertation, is a particular form of data compression used in large-scale storage systems to reduce redundancy [75], [85]. Data deduplication divides the input data stream into chunks and compresses the data by replacing repeated chunks with pointers to their earlier occurrences. Compared to conventional data compression, which usually applies to a single file, deduplication is especially effective for handling large volumes of data. Deduplication gain, i.e., the factor by which deduplication reduces the size of the data, is found to be between 3 and 10 for data stored on desktop computers [70].

While extensively studied in practice, data deduplication is considered from an information-theoretic point of view by only a limited number of works [62], [64], [77], where [62] and [64] are our prior works. Without mathematical frameworks and analytical methods, our knowledge of the fundamental limits of data deduplication is severely limited. Given a probabilistic representation of the data, we do not know what the best possible deduplication gain is. We thus cannot evaluate deduplication algorithms against what may be achieved and lack the road map for designing optimal algorithms with performance guarantees. Therefore, in this dissertation, we aim to develop a flexible mathematical modeling framework for data deduplication. In Section II, a detailed discussion about the existing works [62], [64], [77] will be presented, where varieties of source models are proposed and deduplication algorithms are evaluated over the source models with the source entropy being the baseline.

In addition to developing general models for repeat-rich data, this dissertation focuses on tailor-made modeling of a specific form of data: genomic sequence data. In the past two decades, genomic sequence data is also growing at a fast rate with the development and the increasing availability of high-throughput sequencing tools. The cost of sequencing a human-sized genome has fallen from $100M in 2001 to just over a $1000 in 2015 [76]. At the same time, the utility of DNA sequence data has become more evident in various fields, from phylogenetics to immunology to precision medicine. A single genetics study can produce TBs of genome sequencing data [39]. Compression and storage of biological data also benefit from developing and leveraging models of biological processes that generate this type of data. Biological sequences are created over billions of years by mutation processes, including substitution, insertion, deletion, and duplication, which fuel evolution. These processes shape the statistical properties of sequence data and play a critical role in determining the efficacy of compression algorithms. In Section III, we present results from prior works [61], [63] where genomic sequences are modeled as outcomes of random evolutionary processes driven by mutations. In particular, among various types of mutations, duplication is particularly interested as it is hypothesized to be the primary mechanism that initiates the creation of new genetic material [79]. Through the study of the statistics k-mer frequencies, results about the evolutionary process entropy and string waiting times are derived in [61] and [63], respectively.

B. Background

Reducing the size of data is an old problem that arises both in communications and data storage. One of the most common approaches to this problem is lossless data compression, which we will discuss first in order to introduce fundamental concepts from information theory. Data compression decreases the size of the data by representing common elements with short bit strings and uncommon elements with longer ones. As an example, consider English text, which for simplicity we assume to consist only of capital letters and space. Since there are 27 possible symbols, we can represent each with a unique binary sequence of length 5 (there are 32 binary sequences of length 5). However, this is not efficient. If we assign shorter representations to more common letters, e.g., represent 'e' by '10' and 'q' by '000101', on average, we can use a smaller number of bits per alphabet symbol. The central question of data compression is that, given a source of information such as text, speech, or data collected by a sensor, what is the shortest possible representation?

A framework for finding the answer to this question is provided by information theory, founded by Claude Shannon [97]. Mathematically, the information source is viewed as a random process. For example, a simple model of English text can be obtained by repeatedly choosing letters of the alphabet at random, with probabilities determined by their frequencies [97]. In other words, text is modeled as a sequence of independent and identically distributed random variables. For a random variable X with distribution p, the entropy $H(X)$ is defined as

$$H(X) = \sum_{x} p(x) \log \frac{1}{p(x)}.$$

One of the major results in information theory, called the source coding theorem, states that the average number of bits required to represent each of the outcomes in a sequence of trials cannot be less than the entropy. Furthermore, there exists an encoding scheme, i.e., a mapping between outcomes and binary sequences, that achieves the entropy lower bound. Based on this simple model, the entropy of English is approximately 4 bits/letter and no compression method can achieve a better average rate. (This model is of course too simple to provide accurate results. More accurate models based on Markov processes provide estimates of about 2 bits/letter.)

In practice, it is often the case that we do not know the distribution of the information source that generates data. What we are given is just an individual outcome, e.g., text or music. If we do not put any restrictions on the class of algorithms, then we can always compress a particular sequence to just one bit and this is clearly ‘overfitting’. A common assumption in such situations is to assume a family $\mathcal{P}$ of distributions to which the information source belongs. For instance, $\mathcal{P}$ can be the collection of iid [21], [82], [95], [96] or Markov [20], [45], [46] sources. The question ‘how well can we compress the sequence’ thus becomes how many bits will we waste if our guess about the true distribution is not correct. This is the universal compression framework. A good compressor needs to have ‘universality’ over every possible source in the family, unlike the extreme case where only a particular sequence is compressed to 1 bit. Another challenge that we face in practice is the unknown/large alphabet. Given information sources like texts, the underlying alphabet can be of large size. In applications like data deduplication, the number of possible units is even infinity. Computational issues occur in such large alphabet scenario and conventional results assuming fixed alphabet size often do not apply. A solution is to consider compressing the pattern sequences under the universal compression framework. Formal notions and definitions about universal compression and patterns will be introduced in Section II-F.

II. An Information-theoretic Analysis and Methods Development for Data Deduplication

Data deduplication is an efficient data reduction approach that reduces storage space by eliminating duplicate data [70], [99], [111]. It is usually used in large-scale storage systems, e.g., LBFS (low-bandwidth network file system) [75] and Venti [85]. A typical *chunk-level* data deduplication system uses a chunking scheme to parse the data stream (e.g., backup files, database snapshots, virtual machine images, etc) into multiple data ‘chunks’ that are each associated with a hash signature, also called a *fingerprint*. These chunks can be fixed in size [85] or of variable sizes determined by the content itself [75]. Deduplication only stores the unique chunks on disk. It also records the list of constituent chunks in *metadata* that will be used to reconstruct the original file. A typical mathematical abstraction of data deduplication algorithms can be described as follows [77]. After parsing the data stream into chunks, a sequential processing scheme is performed over them. If a chunk is new, then it is encoded by a bit 1 followed by the chunk itself. If a chunk has appeared before, it will be replaced by a bit 0 followed by a pointer to its first appearance. For example, if the data stream 0100001011 is parsed as $01, 000, 01, 011$, then it is encoded by $101|1000|00|1011$.

Conventional data compression approaches, such as LZ77/LZ78 [117], [118], LZW [113], LZO [78] and DEFLATE [22], only identify redundancy for short strings (e.g., 16B) and compress data in a much smaller region, e.g., a single file or multiple small files, due to their time and space complexity. For large-scale storage systems, data deduplication is more scalable and efficient than traditional compression algorithms. Deduplication eliminates redundancy at the chunk- (e.g., 8KB) or file-level.

Deduplication systems identifies duplicate content (files or chunks) by its hash-based fingerprints which is of size order of magnitude smaller than that of the original data and thus saves storage space.

As mentioned, files or data streams are divided into small data chunks so that each can be fingerprinted. The simplest chunking approach is to cut the file/data stream into equal, fixed-sized chunks, referred to as **fixed-size chunking** (**FSC**). In FSC, if a part of a file or data stream, no matter how small, is modified by the operation of insertion or deletion, not only is the data chunk containing the modified part changed but also all subsequent data chunks will change. This is because the boundaries of all these chunks are shifted. This can cause identical chunks (before modification) to be completely different, resulting in a significantly reduced duplicate identification ratio of FSC-based data deduplication. To address this boundary-shift problem [52], [85], the **content-defined chunking** (**CDC**) is more widely adopted. CDC uses a sliding-window technique on the content of files and computes a hash value. A chunk breakpoint is determined if the hash value of this sliding window satisfies some predefined condition and hence modification does not affect subsequent chunks. An an example, the Rabin algorithm [86] is currently widely used in computing the hash value of the sliding window for CDC.

Data chunks are fingerprinted and this technique largely simplifies the process of duplicate identification. In conventional data reduction approaches (such as LZ compression [118] and Xdelta [65]), the duplicates are first matched by their calculated weak hash digest and then further confirmed by a byte-by-byte comparison. In data deduplication systems, the duplicates are completely represented by their cryptographic hash-based fingerprints (e.g., SHA1, SHA256) and the matched fingerprints mean that their represented contents are, with high probability, identical to each other. The probability of a hash collision when data deduplication is carried out in an EB-scale storage system, based on the average chunk size of 8 KB and finger prints of SHA1, is smaller than 10^{-20} [114]. In contrast, in computer systems, the probability of a hard disk drive error is about $10^{-12} \sim 10^{-15}$ [93], which is much higher than the aforementioned probability of SHA1-fingerprint collisions in data deduplication. Consequently, SHA1 has become the most wildly used finger- printing algorithm for data deduplication because most existing approaches, such as LBFS [75], Venti [85], and DDFS [115]. More recently, stronger hash algorithms, such as SHA256, have been considered for fingerprinting in some data deduplication systems to further reduce the risk of hash collision.

Data deduplication has been studied extensively practically. Microsoft [70], [99] and EMC [98], [111] analyzed workloads of primary and secondary storage systems and showed that deduplication schemes effectively reduced redundancy. Varieties of chunking methods are proposed and studied [32], [52], [75]. Methods for efficient indexing of fingerprints are studied in [7], [58], [115]. Applications of data deduplication in cloud storage to save network bandwidth and accelerate synchronization are discussed in [24], [25], [110].

However, due to the lack of a mathematical framework and analytical methods, our knowledge of the fundamental limits of data deduplication is severely limited. Given a probabilistic representation of the data, we do not know what the best possible deduplication gain is. We thus cannot evaluate deduplication algorithms against what may be achieved and lack the road map for designing optimal algorithms with performance guarantees. Therefore, an information-theoretic analysis of deduplication algorithms is important.

The first information-theoretic analysis of data deduplication was conducted by Niesen [77]. The performances of three deduplication algorithms FLD, VLD and MCD, with formal definitions given in Section II-B3, were studied over a source model which produces data streams that are composed of blocks with each block being an exact copy of one of the source symbols, where the source symbols are pre-selected strings. We state Niesen's results as a preliminary in Section II-C.

It is often the case, however, that the copies of a block of data that is repeated many times are approximate, rather than exact. This may occur, for example, due to edits to the data, or in the case of genomic data[1], due to mutations. Therefore, in [64] and [62], the problem of deduplication when the repeats are approximate is was considered. In particular, in [62], data deduplication algorithms are analyzed over source models with probabilistic edits. For the fixed-length chunking scheme, three algorithms: a generalization of FLD [77] named **modified fixed-length deduplication** (**mFLD**), a variant of mFLD named **adaptive fixed-length deduplication** (**AFLD**), and the **edit-distance deduplication** (**EDD**), are presented and analyzed. Due to the boundary-shift problem, algorithms in the fixed-length scheme are studied over the source model where all source symbols have the same length. It was shown that for mFLD, if the chunk length is not properly chosen, the average length of the compressed strings is greater than source entropy by an arbitrarily large multiplicative factor for small enough edit

[1] Repeats are common in genomic data. For example, a majority of the human genome consists of interspersed and tandem repeated sequences [53].

probability. Meanwhile, AFLD and EDD take source model parameters into account and are shown to have performances within a constant factor of optimal. For the variable-length scheme, a general scenario where source symbols are of random lengths are considered. It is shown that VLD can achieve large compression ratios relative to the length of the uncompressed strings. Results derived over this probabilistic edit model are presented in Section II-D

While [62] extended the information-theoretic analysis of data deduplication to approximate repeats, the studied model has entropy linear in the length of the uncompressed string and the gain in compression is at best a constant factor. This makes compression less challenging and the distinction between the performance of compression methods less clear. In [64], each source block is assumed to contain a constant number of substitutions (randomly distributed) instead of iid bit flips, leading to the entropy being of smaller order than the length of the uncompressed string and thus high compression ratio can be achieved. Moreover, [64] also studied the MCD algorithm proposed by [6], which has not studied before in models with edits. Results derived over the source model with a constant number of substitutions are presented in Section II-E.

In Section II-F, we present a study of data deduplication under the framework on universal compression. Classic universal compression [81], [90], [95], [101], [103] considers encoding sequences generated by sources with a known alphabet. In particular, $\mathcal{I}_k^n$, the class of iid distributions of length-n sequences over an alphabet of size k is considered. The worst case and average case redundancy were determined for different values of k, e.g., when $k = o(n)$, it was shown that $\hat{R}(\mathcal{I}_k^n)$ and $\bar{R}(\mathcal{I}_k^n)$ are both dominated by $\frac{k-1}{2}\log\left(\frac{n}{k}\right)$. However, in applications like language modeling, alphabets can be very large or even unknown. To address this problem, [82] took an approach of describing the sequences in two separate parts: the symbols appeared and the pattern they form. The pattern of a sequence is the sequence of integers representing orders in which the symbols appear. For example, the pattern of the sequence “lossless” is “12331433”. The encoding of sequence “lossless” thus consists of the encoding of appeared symbols ‘l’, ‘o’, ‘s’, ‘e’ (ordered), and the pattern “12331433”.

Data deduplication is an application of this encoding scheme [58], [115]. Recall that if a chunk is new, then it is encoded by a bit 1 followed by the chunk itself; if a chunk has appeared before, it will be replaced by a bit 0 followed by a pointer to its first appearance. The above algorithm adopts the separate encoding scheme with chunks viewed as new unit ‘symbols’. Unique chunks are stored in full, corresponding to storage of symbols. The one-bit indicators and pointers are independent of the actual content of chunks, corresponding to encoding of the pattern. Due to the scale of data, encoding schemes in deduplication algorithms are of low complexity. As described, every chunk is encoded by an indicator followed by a pointer if it has appeared before. To keep the whole process simple, the pointer is encoded by number of bits equal to log of the number of distinct chunks seen before. Therefore, the only information needs to be kept in memory is the collection of chunks that have previously appeared and the computation is simple as all chunks appeared before are viewed as equally likely. However, pattern compressors that were shown in [82], [96] to have performance close to optimal need to store how many times symbols appear so that frequent integers get short representation. So relatively higher computation complexity is required.

A. Preliminaries and Notation

We consider the binary alphabet $\{0, 1\}$, denoted Σ. The set of all finite strings over Σ (including the unique empty string) is denoted Σ^*. A j-(sub)string is a (sub)string of length j. For a non-negative integer m, let Σ^m be the set of all strings of length m over Σ. For strings $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$, the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted $\boldsymbol{uv}$, and the concatenation of i copies of $\boldsymbol{u}$ is denoted $\boldsymbol{u}^i$. We denote the substring of length ℓ starting from the j-th symbol of $\boldsymbol{u}$ by $\boldsymbol{u}_{j,\ell}$, which is also referred to as the j-th ℓ-substring of $\boldsymbol{u}$. The length of $\boldsymbol{u}$ is denoted $|\boldsymbol{u}|$. The cardinality of a set S is also denoted $|S|$. For a set T of strings, $\boldsymbol{u}$ is said to be a substring of T if $\boldsymbol{u}$ is a substring of one or more strings in T.

All logarithms are to the base 2. For $0 \le p \le 1$, $H(p)$ denotes the binary entropy function: $p\log\left(\frac{1}{p}\right) + (1-p)\log\left(\frac{1}{1-p}\right)$. For $0 \le p, q \le 1$, $H(p, q)$ denotes the cross entropy function: $p\log\left(\frac{1}{q}\right) + (1-p)\log\left(\frac{1}{1-q}\right)$. For an event $\mathcal{E}$, we use $\bar{\mathcal{E}}$ to denote its complement and use $I_{\mathcal{E}}$ to denote the indicator variable for $\mathcal{E}$, which takes value 1 when $\mathcal{E}$ is true, and 0 otherwise.

The following inequalities are used frequently: for $x \in (0, 1)$ and a positive integer n,

$$\frac{1}{2}\min(1, nx) \le 1 - (1-x)^n \le \min(1, nx). \tag{1}$$

A binary string is k-runlength-limited (k-RLL) [67] if it does not contain k consecutive zeros, i.e., all runs of zeros in the string are of lengths less than k. We denote the set of binary k-RLL strings by R_k and denote the set of binary k-RLL strings of length n by R_k^n. The following lemma provides bounds on the size of R_k^n.

Lemma 1. *Let k be a positive integer. The number of binary k-RLL strings of length n, $|R_k^n|$, satisfies*

$$(2-\frac{1}{2^{k-2}})^n \leq |R_k^n| \leq 2(2-\frac{1}{2^k})^n. \tag{2}$$

Proof: Clearly, if $0 \leq n \leq k-1$, then any string of length n is a k-RLL string (we consider the empty string as the only string of length 0). Therefore, for all $0 \leq n \leq k-1$,

$$|R_k^n| = 2^n \geq (2-\frac{1}{2^{k-2}})^n, \tag{3}$$

and

$$|R_k^n| = 2^n = 2^{n+1}2^{-1} \leq 2^{n+1}(1-\frac{1}{2^{k+1}})^{k-1} \leq 2^{n+1}(1-\frac{1}{2^{k+1}})^n = 2(2-\frac{1}{2^k})^n. \tag{4}$$

For $n \geq k$, we prove the lemma by induction on n. Suppose the desired results hold for all $n' < n$. It is shown in [94, Chapter 8] that $\left|R_k^N\right| = \sum_{i=1}^k \left|R_k^{N-i}\right|$ for all $N \geq k$. Therefore,

$$\begin{aligned} |R_k^n| = \sum_{i=1}^k \left|R_k^{n-i}\right| \geq \sum_{i=1}^k (2-\frac{1}{2^{k-2}})^{n-i} &= \frac{(2-\frac{1}{2^{k-2}})^n - (2-\frac{1}{2^{k-2}})^{n-k}}{1-\frac{1}{2^{k-2}}} & (5)\\ &= \left(2-\frac{1}{2^{k-2}}\right)^n + \frac{(2-\frac{1}{2^{k-2}})^{n-k}2^k}{2^{k-2}-1}\left((1-\frac{1}{2^{k-1}})^k - \frac{1}{4}\right) \geq (2-\frac{1}{2^{k-2}})^n, & (6)\end{aligned}$$

and

$$\begin{aligned} |R_k^n| = \sum_{i=1}^k \left|R_k^{n-i}\right| \leq \sum_{i=1}^k 2(2-\frac{1}{2^k})^n &= 2\frac{(2-\frac{1}{2^k})^n - (2-\frac{1}{2^k})^{n-k}}{1-\frac{1}{2^k}} & (7)\\ &= 2(2-\frac{1}{2^k})^n + \frac{2(2-\frac{1}{2^k})^{n-k}}{1-\frac{1}{2^k}}\left(\left(\frac{2-\frac{1}{2^k}}{2}\right)^k - 1\right) \leq 2(2-\frac{1}{2^k})^n. & (8)\end{aligned}$$

■

By Lemma 1, we bound the number of binary k-RLL strings of lengths at most 2^k in the following corollary.

Corollary 2. *The number of binary k-RLL strings of lengths at most 2^k satisfies*

$$\sum_{n=0}^{2^k} |R_k^n| \geq \sum_{n=0}^{2^k} \left(2-\frac{1}{2^{k-2}}\right)^n \geq 2^{2^k-2}. \tag{9}$$

Sequences and patterns

Let $x^n = x_1x_2\cdots x_n$ be a sequence of n symbols. We use $|x^n|$ to denote the length and use $N(x^n)$ to denote the number of distinct symbols in x^n. We define the index $\iota(x)$ of x to be one more than the number of distinct symbols preceding x's first appearance in x^n. The *pattern* of x^n is defined as the sequence of indexes, i.e.,

$$\Psi(x^n) = \iota(x_1)\iota(x_2)\cdots\iota(x_n). \tag{10}$$

As an example, in the sequence "$abacbbc$", $\iota(a) = 1, \iota(b) = 2, \iota(c) = 3$, and hence,

$$\Psi(abacbbc) = 1213223. \tag{11}$$

In the following, we use ψ to denote a generic pattern. Elements in patterns are referred to as index integers.

We consider a discrete alphabet $\mathcal{A}$ of size k. Let $\mathcal{A}^n$ denote the set of all sequences of length n over $\mathcal{A}$ and let $\Psi(\mathcal{A}^n)$ denote the set of patterns of all sequences in $\mathcal{A}^n$, i.e,

$$\Psi(\mathcal{A}^n) = \{\Psi(x^n) : x^n \in \mathcal{A}^n\}. \tag{12}$$

It is clear that $\Psi(\mathcal{A}^n)$ is the same for any alphabet $\mathcal{A}$ of size k. It contains all patterns of length n and at most k distinct index integers. So we will write $\Psi^n_{\leq k}$ instead. For example, if $k = 2$ and $n = 3$, then

$$\Psi^3_{\leq 2} = \{111, 112, 121, 122\}. \tag{13}$$

Let Ψ^n_k denote the set of patterns of length n and with exactly k distinct index integers. It follows that

$$\Psi^n_{\leq k} = \cup^k_{m=1} \Psi^n_m. \tag{14}$$

For a pattern sequence $\boldsymbol{\psi}$, the profile of $\boldsymbol{\psi}$ is a vector of length $|\boldsymbol{\psi}|$, defined as

$$\Phi(\boldsymbol{\psi}) = \big(\varphi_1, \varphi_2, \ldots, \varphi_{|\boldsymbol{\psi}|}\big), \tag{15}$$

where φ_j is the number of index integers that appear j times in $\boldsymbol{\psi}$. For example, in pattern 12131, one integer (which is 1) appears 3 times and two integers (which are 2 and 3) appear once, so $\Phi(12131) = (2, 0, 1, 0, 0)$.

Moreover, we define the innovation vector $\Lambda(\boldsymbol{\psi})$ of $\boldsymbol{\psi}$ to be the vector containing indexes of new symbols. Formally,

$$\Lambda(\boldsymbol{\psi}) = (\lambda_1, \lambda_2, \ldots, \lambda_{N(\boldsymbol{\psi})}), \tag{16}$$

where λ_j is the index of the first occurrence of integer j. For example, in pattern 12131, integers 2 and 3 first appear at positions 2 and 4, respectively. So $\Lambda(12131) = (1, 2, 4)$. Note that we always have $\lambda_1 = 1$. We use Λ^n_k to denote the set of innovation vectors of all patterns in Ψ^n_k, i.e.,

$$\Lambda^n_k = \{\Lambda(\boldsymbol{\psi}) : \boldsymbol{\psi} \in \Psi^n_k\}, \tag{17}$$

and write $\Lambda^n_{\leq k} = \cup^k_{m=1} \Lambda^n_m$.

B. Models & Algorithms

In this section, we introduce source models and algorithms for data deduplication.

1) Source model $\mathcal{I}$

In [77], data in storage systems is modeled as a concatenation of blocks with each block being an exact copy of one of the source symbols, where the source symbols are pre-selected strings. Specifically, the information source $\mathcal{I}$ is modeled as follows. From a fixed distribution $\mathbb{P}_l$ over positive integers, A iid numbers are first drawn, denoted $L_1, \ldots, L_A$. A source alphabet $\mathcal{X} = \{\mathsf{X}_1, \ldots, \mathsf{X}_A\}$ which contains A binary strings is then generated: starting from $a = 1$, X_a is uniformly randomly chosen from the set $\{0, 1\}^{L_a} \setminus \{\mathsf{X}_1, \ldots, \mathsf{X}_{a-1}\}$. The output data string S is then a concatenation of B iid samples from $\mathcal{X}$, i.e., $\mathsf{S} = Y_1 \cdots Y_B$, where each Y_b is selected uniformly from $\mathcal{X}$ and independent from the others. Strings $Y_1, \ldots, Y_B$ are referred to as source blocks. The entropy of source $\mathcal{I}$ is denoted $H(\mathcal{I})$. For simplicity, the length distribution $\mathbb{P}_l$ is assumed to be concentrated around the mean, specifically, $\mathbb{P}_L(L/2 \leq l \leq 2L) = 1$ where L is the mean.

Example 1. Assume the source alphabet is randomly generated as $\mathcal{X} = (1, 00, 01100, 001100)$. From this, $B = 2$ elements are drawn iid uniformly at random, say $Y_1 = 10$ and $Y_2 = 01100$. The resulting source sequence is then $\boldsymbol{s} = Y_1Y_2 = 1001100$. Note that given $\boldsymbol{s}$ alone, the boundary between 10 and 01100 can not be observed.

2) Source models with random substitution

The source model $\mathcal{I}_b(\delta)$ proposed in [62] extends $\mathcal{I}$ by allowing probabilistic substitutions. The output data stream $\boldsymbol{s}$ is a concatenation of approximate copies of source symbols. The A source symbols, denoted $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_A$, are iid binary strings generated in the following way. Again, fix a length distribution $\mathbb{P}_l$ over positive integers with mean L. For each $1 \leq a \leq A$, we draw L_a from $\mathbb{P}_l$ and draw X_a uniformly from Σ^{L_a}. It is important to note that, as a result of sampling with replacement, the source symbols are distributed uniformly and independently. The probability that $(\mathsf{X}_1, \ldots, \mathsf{X}_A) = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_A)$ given the lengths L_a is $\prod^A_{a=1} \frac{1}{2^{L_a}}$, for any set of strings $(\boldsymbol{x}_a)$ where $\boldsymbol{x}_a$ has length L_a. So the same sequence can be drawn multiple times as source symbols. The draws are treated as separate symbols, but with the same content. We use $\mathcal{X}$ to denote the source symbol alphabet, i.e., $\mathcal{X} = \{\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_A\}$. The alphabet is thus a multiset. To simplify some of the derivations, we adopt the same assumption that $\mathbb{P}_l$ is concentrated around its mean, specifically, $\mathbb{P}_l(\frac{L}{2} \leq l \leq 2L) = 1$.

After generating the source symbols $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_A$, we generate an iid sequence of length B, denoted $Y_1, \ldots, Y_B$, where each Y_b is an approximate copy of a randomly chosen source symbol. Specifically, for each $1 \leq b \leq B$, we first pick J_b

uniformly at random from $\{1, 2, \ldots, A\}$. Next, we generate Y_b by flipping each bit of X_{J_b} independently with probability δ, as a way of simulating edits and other changes to the data in a simple manner. The bit flipping process is referred to as a δ-edit. The data stream $\boldsymbol{s}$ will be a concatenation of $Y_1, Y_2, \ldots, Y_B$, i.e., $\boldsymbol{s} = Y_1 Y_2 \cdots Y_B$. The approximate copies $Y_1, Y_2, \ldots, Y_B$ are referred to as source blocks. The real number δ is referred to as the *edit probability*. The entropy of this source is denoted $H(\mathcal{I}_b(\delta))$.

Example 2. Following Example 1, $\mathcal{X} = (1, 00, 01100, 001100)$ and $B = 2$. Source strings 10 and 01100 are first chosen and δ-edits are applied. Possible outcomes could be 11 and 11000, with probabilities $\delta(1-\delta)$ and $\delta^2(1-\delta)^3$, respectively. The output data string $\boldsymbol{s}$ is therefore 1111000.

While $\mathcal{I}_b(\delta)$ studied probabilistic substitution edits where the total number of edits is linear in the sequence length, [64] considered the source model $\mathcal{I}_f(t)$ where a fixed number of substitutions occur in each block Y_b. After generating the source alphabet $\mathcal{X}$, we sample B times from $\mathcal{X}$ uniformly at random with replacement and get $\mathsf{X}_{J_1}, \mathsf{X}_{J_2}, \ldots, \mathsf{X}_{J_B}$ in order. For every X_{J_b}, we then flip t ($t \leq L/2$) symbols uniformly at random. The number of flipped symbols t will be referred to as the *substitution number*. The flipped version of X_{J_b} is denoted Y_b. The source string $\boldsymbol{s}$ is again constructed to be the concatenation of source blocks, i.e., $\boldsymbol{s} = Y_1 Y_2 \ldots Y_B$. The entropy of this source is denoted $H(\mathcal{I}_f(t))$.

Example 3. Following Example 1, $\mathcal{X} = (1, 00, 01100, 001100)$ and $B = 2$. Source strings 001100 and 01100 are chosen and $t = 2$ edits are applied. Possible outcomes could be 101101 and 00000, with probabilities $1/\binom{6}{2}$ and $1/\binom{5}{2}$, respectively. The output data string $\boldsymbol{s}$ is therefore 00110000000.

Note that in the three source models, the boundaries between source blocks are not known given $\boldsymbol{s}$. Moreover, the source models are studied over the asymptotic regime in which $B, A, L \to \infty$ while the edit probability δ and substitution number t remain constants. The following Lemmas bounds the source entropy of $\mathcal{I}_b(\delta)$ and $\mathcal{I}_f(t)$, which will serve as fundamental lower bounds on the performance of deduplication algorithms.

Lemma 3 ([62]). *As $B \to \infty$, the entropy of the source model $\mathcal{I}_b(\delta)$ satisfies*

$$H(\delta)BL \leq H(\mathcal{I}_b(\delta)) \leq H(\delta)BL + B\log A + A(2L+1). \tag{18}$$

Proof: For the lower bound,

$$H(\mathcal{I}_b(\delta)) \geq H(\mathcal{I}_b(\delta)|\mathsf{X}_{J_1}, \ldots, \mathsf{X}_{J_B}) = \sum_{b=1}^{B} H(Y_b|\mathsf{X}_{J_b}) = H(\delta)BL. \tag{19}$$

For the upper bound,

$$\begin{aligned} H(\mathcal{I}_b(\delta)) \leq & H(\mathcal{I}_b(\delta)|\mathsf{X}_{J_1}, \ldots, \mathsf{X}_{J_B}) + H(\mathsf{X}_{J_1}, \ldots, \mathsf{X}_{J_B}|\mathcal{X}) + H(\mathcal{X}) & (20)\\ \leq & H(\delta)BL + B\log A + A(2L+1), & (21)\end{aligned}$$

where $H(\mathcal{X}) \leq A(2L+1)$ follows from the fact that for each X_a, there are at most 2^{2L+1} different possibilities since we assume $L_a \leq 2L$. ■

Lemma 4 ([64]). *The entropy of the source model $\mathcal{I}_f(t)$ satisfies*

$$B\log\binom{L/2}{t} \leq H(\mathcal{I}_f(t)) \leq B\log\left(A\binom{2L}{t}\right) + (2L+1)A.$$

Proof: For the lower bound,

$$\begin{aligned} H(\mathcal{I}_f(t)) &\geq H(\mathcal{I}_f(t)|\mathsf{X}_{J_1}, \ldots, \mathsf{X}_{J_B}) = H(Y_1^B|\mathsf{X}_{J_1}, \ldots, \mathsf{X}_{J_B}) \\ &= \sum_{b=1}^{B} H(Y_b|\mathsf{X}_{J_b}) \geq B\log\binom{L/2}{t}. \end{aligned}$$

For the upper bound,

$$\begin{aligned} H(\mathcal{I}_f(t)) &\le H(Y_1,\dots,Y_B) \le H(Y_1,\dots,Y_B|\mathcal{X}) + H(\mathcal{X}) \\ &\le B\log\left(A\binom{2L}{t}\right) + (2L+1)A, \end{aligned}$$

where $H(\mathcal{X}) \le (2L+1)A$ follows from the assumption that $\mathbb{P}_l(L_a \le 2L) = 1$. ■

3) Deduplication Algorithms

In this section, we formally state the deduplication algorithms which are studied as mathematical abstractions of real-world deduplication systems. All algorithms are dictionary-based and composed of two stages: chunking and encoding. In particular, the conventional **fixed-length deduplication** (**FLD**), **variable-length deduplication** (**VLD**) and **multi-chunk deduplication** (**MCD**) were formalized in [77]. The *modified fixed-length* deduplication (mFLD) and *edit-distance* deduplication (EDD) were proposed and discussed in [62].

In FLD, a chunk length ℓ is fixed. Source string $\boldsymbol{s}$ is parsed into segments of length ℓ, i.e., $\boldsymbol{s} = \boldsymbol{z}_1\boldsymbol{z}_2\cdots\boldsymbol{z}_{C+1}$, where $|\boldsymbol{z}_1| = |\boldsymbol{z}_2| = \cdots = |\boldsymbol{z}_C| = \ell$, $C = \lfloor |\boldsymbol{s}|/\ell \rfloor$. The substrings $\{\boldsymbol{z}_c\}_{c=1}^{C+1}$ are collected as deduplication chunks. The encoding process starts with encoding the length of $\boldsymbol{s}$ by a prefix-free code for the positive integers (such as an Elias gamma code [27]), to ensure that the whole scheme is prefix-free. The chunks are then encoded sequentially. Starting with $c = 1$, if chunk $\boldsymbol{z}_c$ appears for the first time, i.e., $\boldsymbol{z}_c \ne \boldsymbol{z}_i$ for all $i < c$, then it is encoded as the bit 1 followed by $\boldsymbol{z}_c$ itself and is entered into the dictionary. Otherwise, when there already exists an entry in the dictionary storing the same string as $\boldsymbol{z}_c$, it will be encoded as the bit 0 followed by a pointer to that entry. The pointer is an index of the dictionary entries and thus can be encoded by at most $\log\left|T^{c-1}\right| + 1$ bits, where T^{c-1} denotes the dictionary just after $\boldsymbol{z}_{c-1}$ is processed. The number of bits FLD takes to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_F(\boldsymbol{s})$. This encoding process is referred to as the dictionary encoding.

Example 4. For $\boldsymbol{s} = 0111010$ and $\ell = 2$, the chunks generated by fixed-length chunking are $\boldsymbol{z}_1 = 01, \boldsymbol{z}_2 = 11, \boldsymbol{z}_3 = 01, \boldsymbol{z}_4 = 0$. The encoding of length $|\boldsymbol{s}| = 7$ by Elias gamma coding is 00111. Chunks $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ and $\boldsymbol{z}_4$ are new chunks and thus are encoded as $101, 111, 10$, respectively. Chunk $\boldsymbol{z}_3$ is a duplicate of $\boldsymbol{z}_1$. When $\boldsymbol{z}_3$ is processed, the dictionary contains two strings 01 and 11. So $\boldsymbol{z}_3$ is encoded as 00, where the second 0 represents the first entry of the dictionary. Concatenating all components, the final encoding of $\boldsymbol{s}$ is 001111011110010. Note that after encoding terminates, the dictionary is the ordered set $\{01, 11, 0\}$.

The *modified fixed-length* deduplication (mFLD) has the same encoding process as FLD but with a two-stage chunking process. In mFLD, first, the source string $\boldsymbol{s}$ is parsed into segments of length D, and then, each segment is parsed into chunks of length ℓ, where $\ell \le D$. Specifically, the source string $\boldsymbol{s}$ is parsed as

$$\boldsymbol{s} = \boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_{K+1}, \quad |\boldsymbol{x}_1| = |\boldsymbol{x}_2| = \cdots = |\boldsymbol{x}_K| = D, \tag{22}$$

where $K = \lfloor |\boldsymbol{s}|/D \rfloor$ and

$$\boldsymbol{x}_k = \boldsymbol{z}_k^1\boldsymbol{z}_k^2\cdots\boldsymbol{z}_k^{N+1}, \quad \left|\boldsymbol{z}_k^1\right| = \left|\boldsymbol{z}_k^2\right| = \cdots = \left|\boldsymbol{z}_k^N\right| = \ell, \tag{23}$$

with $1 \le k \le K$, $N = \lfloor D/\ell \rfloor$ ($\boldsymbol{x}_{K+1}$ is also parsed in the same way). The number of bits mFLD takes to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_{mF}(\boldsymbol{s})$.

Note that mFLD is a generalization of FLD since with $D = \ell$, mFLD is equivalent to FLD with the same chunk length ℓ. For FLD to perform well, the source symbols must all have the same length L and the chunk length ℓ must also be chosen equal to L to maintain synchronization between the chunks and symbols. The generalization to mFLD allows us to maintain synchronization by setting $D = L$ and frees us to choose other values for ℓ. This flexibility enables us to study the effect of chunk length, which as we will see, will provide important intuitions for more practical algorithms such as VLD. We will focus on analyzing the performance of mFLD and report that of FLD as a corollary.

The *adaptive fixed-length* deduplication (AFLD) is a specialization of mFLD for the source model $\mathcal{I}_b(\delta)$. Source model parameters are taken into account by AFLD. Given A, B, L, δ, AFLD is the version of mFLD with chunk length specified as $\ell = \left\lceil \frac{\log(B/A)}{H(\gamma,\delta)} \right\rceil$ ($\ell = D$ if $D < \left\lceil \frac{\log(B/A)}{H(\gamma,\delta)} \right\rceil$) for some $\gamma \in (\delta, 1/2)$. AFLD thus contains two parameters D and γ. Note that in practice, source model parameters can be estimated from data. We will show later that AFLD is an optimized version of mFLD. The distinction in names is made to emphasize the optimality and also for the convenience of referring to this version of the algorithm. The number of bits AFLD takes to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_{AF}(\boldsymbol{s})$.

The *edit-distance* deduplication (EDD) extends FLD by encoding chunks by representing differences compared to chunks that have appeared before. EDD also takes the source model parameters into account and is only defined for source models with edit probability $\delta < 1/4$. EDD contains two parameters, chunk length ℓ and mismatch ratio β, where $\delta < \beta \le 1/4$. The chunking scheme is the same as in FLD, i.e., parsing source string $\boldsymbol{s}$ into chunks of length ℓ, denoted $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{C+1}$. The encoding starts with a prefix-free code representing the length of the source string. Next, for each chunk $\boldsymbol{z}_c$, it is encoded as the bit 1 followed by itself if no chunk has appeared before whose Hamming distance from $\boldsymbol{z}_c$ is at most $2\beta\ell$. Otherwise, let c' be the smallest index such that the Hamming distance between $\boldsymbol{z}_{c'}$ and $\boldsymbol{z}_c$ is less than or equal to $2\beta\ell$. Chunk $\boldsymbol{z}_c$ will be encoded as the bit 0 followed by a pointer to the dictionary entry where $\boldsymbol{z}_{c'}$ is stored in addition with the bits describing the mismatches between $\boldsymbol{z}_c$ and $\boldsymbol{z}_{c'}$. The mismatches are the indexes of positions in which $\boldsymbol{z}_{c'}$ and $\boldsymbol{z}_c$ differ. Since we restrict the number of mismatches to be no larger than $2\beta\ell$, the mismatches can be encoded in at most $\log\Big(\sum_{i=0}^{\lfloor 2\beta\ell \rfloor} \binom{\ell}{i}\Big) + 1 \le H(2\beta)\ell + 1$ bits. The number of bits EDD takes to store $\boldsymbol{s}$ is denoted by $\mathcal{L}_{ED}(\boldsymbol{s})$.

In *variable-length* deduplication (VLD), a string of length M (we assume 0^M) is fixed to be the marker string. Source string $\boldsymbol{s}$ is parsed into chunks that end with the marker string. Specifically, the source string $\boldsymbol{s}$ is parsed as $\boldsymbol{s} = \boldsymbol{z}_1 \cdots \boldsymbol{z}_C$, where each $\boldsymbol{z}_c$ (except for perhaps the last one) contains a single appearance of 0^M at the end. We again use $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_C$ to represent the chunks. After splitting $\boldsymbol{s}$ into the chunks $\{\boldsymbol{z}_c\}_{c=1}^{C}$, the same dictionary encoding process as in FLD is conducted. The number of bits variable-length deduplication takes to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_{VL}(\boldsymbol{s})$.

Example 5. Consider $\boldsymbol{s} = 01101101$. VLD, with marker length $M = 1$, parses $\boldsymbol{s}$ as chunks $0, 110, 110, 1$. The length of $\boldsymbol{s}$ is still encoded by 0001000. Chunks $0, 110, 1$ are new and are encoded with $10, 1110, 11$, respectively. The second occurrence of 110 is encoded by a 0 followed by the pointer 1. The final encoding of $\boldsymbol{s}$ is thus 00010001011100111.

In *multi-chunk* deduplication (MCD), the source string $\boldsymbol{s}$ is again split into chunks by the marker 0^M, but with an additional requirement that chunk lengths are at least 2^{M-1}. We call the chunking process *multi-chunking*. With an abuse of notation, we still denote the chunks by $Z_1, \ldots, Z_C$. The encoding starts with a prefix-free code representing the length of $\boldsymbol{s}$. Chunks are encoded sequentially with a growing dictionary. Consider the chunk Z_c. We assume first that Z_c is new, i.e., it is different from any previously appeared chunk. Let V_c be the largest integer such that chunks $Z_c, Z_{c+1}, \ldots, Z_{c+V_c-1}$ are also new. These new chunks are bundled up and encoded as the bit 1, followed by an encoding of V_c using a prefix-free code for the positive integers, followed by the binary string $Z_c Z_{c+1} \cdots Z_{c+V_c-1}$. Moreover, $Z_c, \ldots, Z_{c+V_c-1}$ are entered into the dictionary in order. Note that each of them is identifiable because they end with the marker 0^M. On the other hand, assume Z_c is not new. Let $\tilde{c} < c$ be the smallest integer satisfying $Z_{\tilde{c}} = Z_c$. Consider the dictionary entry containing $Z_{\tilde{c}}$ and the list of subsequent entries. Let W_c be the largest integer such that the first W_c entries in this list are equal to $Z_c, Z_{c+1}, \ldots, Z_{c+W_c-1}$. Then the chunks Z_c through Z_{c+W_c-1} are bundled up and encoded together as the bit 0, followed by an encoding of W_c using a prefix-free code for the positive integers, followed by a pointer into the dictionary entry containing chunk $Z_{\tilde{c}}$. The expected number of bits for multi-chunk deduplication to encode $\boldsymbol{s}$ is denoted $\mathcal{L}_{MC}(\boldsymbol{s})$.

C. Performance of Algorithms over Source Model $\mathcal{I}$

In [77], algorithms FLD, VLD and MCD were studied over the source model $\mathcal{I}$. The fixed-length deduplication was analyzed over a constant source-symbol length , i.e., $\mathbb{P}_l(l = L) = 1$.

Theorem 5 ([77])**.** *Consider the source model $\mathcal{I}$ with B source blocks drawn with replacement from the A source symbols of constant length L. The performance of VLD with chunk length $\ell = L$ satisfies*

$$1 \le \frac{\mathbb{E}[\mathcal{L}_F(\boldsymbol{s})]}{H(\mathcal{I})} \le 1 + 7\frac{B + \log L}{\min\{A, B\}(L-1) + \max(B - A, 0)\log(A/2)} \tag{24}$$

for B large enough.

In particular, as long as $\omega(1) \le A \le (1-\epsilon)B$ for some $\epsilon > 0$, the upper bound in Theorem 5 becomes $\mathbb{E}[\mathcal{L}_F(\boldsymbol{s})] \le (1 + o(1))H(\mathcal{I})$ as B grows, showing the asymptotic optimality of FLD with known and constant source-symbol lengths.

However, the assumption that source symbols are of a known fixed length is unrealistic. As soon as this assumption is relaxed, FLD can be substantially suboptimal, due to the loss of synchronization between source blocks and deduplication chunks.

Example 6 ([77])**.** Consider the scenario with $A = 2, B = 3$, and let the source symbol length distribution $\mathbb{P}_l$ assigns equal mass to the values L and $L + 1$. The fixed-length deduplication with chunk length $\ell = L$ has rate

$$\frac{\mathbb{E}[\mathcal{L}_F(\boldsymbol{s})]}{H(\mathcal{I})} \geq \Omega(B) \tag{25}$$

as $B \to \infty$.

Unlike fixed-length chunking, variable-length chunking utilizes marker sequences to achieve the synchronization of deduplication chunks and source blocks and performs better.

Theorem 6 ([77])**.** *Consider the source model with B source blocks drawn with replacement from the A source symbols of expected length L. The performance of the variable-length deduplication scheme with optimized marker length M satisfies*

$$1 \leq \frac{\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]}{H(\mathcal{I})} \leq 1 + \frac{5B(1+L)\log(BL)}{\Big(0, \min(A,B)(L-1) + (B-A)^+ \log(A/2) - 2B\log(2L)\Big)^+}. \tag{26}$$

The preceding result is illustrated in the following two examples.

Example 7 ([77])**.** Following Example 6, consider again the scenario with $A = 2$ source symbols with symbol-length distribution $\mathbb{P}_l$ assigning equal mass to the values L and $L+1$, and with $B = 3L$ source blocks. The rate of the variable-length deduplication satisfies

$$\frac{\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]}{H(\mathcal{I})} \leq O\big(\log^3 B\big). \tag{27}$$

Thus, VLD is only suboptimal by at most a polylogarithmic as opposed to a linear factor for FLD.

Example 8 ([77])**.** Consider the scenario with $A = 10^5$ source symbols and with $B = 10^6$ source blocks. Let the expected value of source symbol length be $L = 10^6$ bits as well. Theorem 6 shows that VLD has performance

$$\frac{\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]}{H(\mathcal{I})} \leq 3. \tag{28}$$

i.e., is within a factor of 3 of optimal.

Note that performance bounds on VLD in Theorem 6, Examples 7 and 8 are obtained for the optimal choice of the marker length M. Value M governs the expected chunk length of VLD and balances two competing requirements: On the one hand, for each already encountered chunk, we need to encode a pointer into the dictionary. A smaller chunk length increases both the number of chunks that need to be encoded and the size of the pointers. On the other hand, a larger chunk length increases the lengths of unique chunks which are inefficiently encoded.

Unfortunately, even with the optimal choice of marker length, VLD can be asymptotically significantly suboptimal performance, as shown in the following example.

Example 9 ([77])**.** Consider the scenario with $A = \sqrt{B}$ source symbols of constant length $L = \sqrt{B}$. It can be shown that

$$\Omega\Big(B^{1/4}\log^{-2}(B)\Big) \leq \frac{\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]}{H(\mathcal{I})} \leq O\Big(B^{1/4}\Big). \tag{29}$$

The multi-chunk deduplication (MCD) circumvents this trade-off by encoding multiple chunks jointly. This allows to choose the expected chunk length to be quite small, thereby limiting the effect of the boundary chunks, without the penalty of increased number of dictionary points. The next theorem bounds the performance of MCD.

Theorem 7 ([77])**.** *Consider the source model with B source blocks drawn with replacement from the A source symbols of* *expected length L. The performance of MCD with optimized marker length M satisfies*

$$1 \leq \frac{\mathbb{E}[\mathcal{L}_{MC}(\boldsymbol{s})]}{H(\mathcal{I})} \leq 1 + O\left(\frac{B \log(ABL)}{\Big(\min\{A,B\}(L-1) + (B-A)^{+}\log(A/2) - 2B\log(2L)\Big)^{+}}\right) \tag{30}$$

as $B \to \infty$.

By the preceding theorem, if $B^{\Omega(1)} \leq A \leq (1-\epsilon)B$ and $L \leq A^{\epsilon/3}$ for some constant $\epsilon > 0$, MCD is asymptotically within a constant factor of optimal. Further, if $A \leq B \leq o(AL/\log(AL))$, then multi-chunk deduplication is asymptotically optimal as $B \to \infty$.

Example 10 ([77])**.** Following Example 9, consider again the scenario with $A = \sqrt{B}$ source symbols of constant length $L = \sqrt{B}$. Theorem 7 shows that the rate of MCD satisfies

$$\frac{\mathbb{E}[\mathcal{L}_{MC}(\boldsymbol{s})]}{H(\mathcal{I})} \leq O(1) \tag{31}$$

as $B \to \infty$. Thus MCD is order optimal in this case, as opposed to the polynomial loss factor of VLD.

D. Performance of algorithms over source $\mathcal{I}_b(\delta)$

In this section, we study performances of various algorithms over the source model $\mathcal{I}_b(\delta)$. We consider the asymptotic scenario where A, L are functions of B with $A \leq B^{1-k_2}$ for some $0 < k_2 < 1$ and $L = \Theta\big(B^{k_1}\big)$ for some $k_1 > 0$. We allow A to grow large because it is reasonable to assume that as the dataset gets larger, the number of unique blocks is also higher. This necessitates L to also grow large. The assumption $A \leq B^{1-k_2}$ ensures that, on average, every source symbol has repeats. The polynomial relationship between L and B ensures that B is much smaller than $2^{\Theta(L)}$. So only a small fraction of all possible strings of length $\Theta(L)$ can appear as source symbols, or edited versions of the source symbols, in the datastream. This is compatible with our intuition that only a small number of all possible strings are valid data, e.g., an image, or a piece of text or code. Furthermore, the polynomial relationship between B and L appears to agree with results from experiments in [99] (also referred to in [77]) suggesting that the reasonable range for L is from a few KB to a few MB ($\approx 10^4$ to 10^7 bits) and for B is on the order of 10^5 to 10^9. Nevertheless, other asymptotic regimes may also be appropriate but are left to future work for simplicity.

A deduplication algorithm is said to *(asymptotically) achieve a constant factor of optimal* if there exists a constant c (independent of δ) such that $\mathbb{E}[\mathcal{L}(\boldsymbol{s})] \leq cH(\mathcal{I}_b(\delta))$, for all $0 < \delta < \frac{1}{2}$ and all sufficiently large B, where $\mathcal{L}(\boldsymbol{s})$ is the length of the encoding produced by the algorithm. Given our assumptions on A, B, L, and the result from Lemma 3, the entropy $H(\mathcal{I}_b(\delta))$ is dominated by the term $H(\delta)\mathbb{E}[|\boldsymbol{s}|]$. If δ is close to $\frac{1}{2}$, $H(\mathcal{I}_b(\delta))$ is close to the length of the uncompressed sequence ($\boldsymbol{s}$ is close to an iid Bernoulli(1/2) process), while if δ is close to 0, there is large gap between the two. Hence, to determine whether an algorithm achieves a constant factor of optimal, the case of small δ is especially important, which is also the case where compression is more beneficial.

We also define the compression ratio $R = \frac{\mathbb{E}[|\boldsymbol{s}|]}{\mathbb{E}[\mathcal{L}(\boldsymbol{s})]}$. Note that if there exists a constant c_1 independent of δ such that $R \leq c_1$, then the algorithm uses more bits than the entropy by an arbitrarily large multiplicative factor as δ goes to 0. While if $R \to \infty$ as $\delta \to 0$, then the algorithm can achieve arbitrarily large compression ratios as entropy decreases. Finally, if there exists a constant c_2 such that $R \geq \frac{c_2}{H(\delta)}$ for all valid δ, then the algorithm achieves a constant factor of optimal.

Before presenting results on performances of various deduplication algorithms, we discuss some strategies for computing $\mathbb{E}[\mathcal{L}(\boldsymbol{s})]$. We say X_{J_b} is the ancestor of Y_b and Y_b is a descendant of X_{J_b}. For each a, we use $Y(a)$ to denote the set $\{1 \leq b \leq B : J_b = a\}$ and use $Y_{1/2}(a)$ to denote the set $\{1 \leq b \leq \lceil B/2 \rceil : J_b = a\}$. In other words, $Y(a)$ is the set of source block indexes of the descendants of X_a and $Y_{1/2}(a)$ is the set of source block indexes of the descendants of X_a among the first half of source blocks.

Note that $\mathbb{E}[|Y(a)|] = B/A$ and $\mathbb{E}[|Y_{1/2}(a)|] = B/(2A)$. We use $\mathcal{E}_u$ to denote the event that $|Y(a)| \leq \frac{3B}{2A}$ for all $1 \leq a \leq A$, and use $\mathcal{E}_l$ to denote the event that $\left|Y_{1/2}(a)\right| \geq \frac{B}{4A}$ for all $1 \leq a \leq A$. Since $|Y(a)| = \sum_{b=1}^{B} I_{J_b = a}$, where all summands are iid with expected value $\frac{1}{A}$, by the Chernoff bound [72] and the union bound,

$$\Pr(\mathcal{E}_u) \geq 1 - Ae^{-\frac{B}{10A}}, \quad \Pr(\mathcal{E}_l) \geq 1 - Ae^{-\frac{B}{16A}}. \tag{32}$$

Given our assumption that $A \leq B^{1-k_2}$, asymptotically $\frac{B}{16A} - \log A$ goes to infinity. So the probability of $\mathcal{E}_u$ goes to 1 (also true for $\mathcal{E}_l$). In the performance analysis of deduplication algorithms, we generally only need to consider the case in which $\mathcal{E}_l$ or $\mathcal{E}_u$ holds. Specifically, we use the following inequalities as bounds on $\mathbb{E}[\mathcal{L}(\boldsymbol{s})]$:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{s})] \leq \mathbb{E}[\mathcal{L}(\boldsymbol{s})|\mathcal{E}_u] + \mathbb{E}[\mathcal{L}(\boldsymbol{s})|\bar{\mathcal{E}_u}] \cdot \Pr(\bar{\mathcal{E}_u}), \tag{33}$$

$$\mathbb{E}[\mathcal{L}(\boldsymbol{s})] \geq \mathbb{E}[\mathcal{L}(\boldsymbol{s})|\mathcal{E}_l] \cdot \Pr(\mathcal{E}_l) = \mathbb{E}[\mathcal{L}(\boldsymbol{s})|\mathcal{E}_l] \cdot \big(1 - \Pr(\bar{\mathcal{E}_l})\big). \tag{34}$$

To find $\mathbb{E}[\mathcal{L}(\boldsymbol{s})]$, we compute the terms $\mathbb{E}[\mathcal{L}(\boldsymbol{s})|\mathcal{E}_u]$, $\mathbb{E}[\mathcal{L}(\boldsymbol{s})|\mathcal{E}_l]$ and show that the terms $\mathbb{E}[\mathcal{L}(\boldsymbol{s})|\bar{\mathcal{E}_u}] \cdot \Pr(\bar{\mathcal{E}_u})$ and $\mathbb{E}[\mathcal{L}(\boldsymbol{s})|\mathcal{E}_l] \cdot \Pr(\bar{\mathcal{E}_l})$ are asymptotically negligible, using trivial bounds on $\mathcal{L}(\boldsymbol{s})$.

1) Deduplication in the fixed-length scheme

It is pointed out by [77] that when all source symbols have the same length and there are no edits, FLD with knowledge of the symbol length can parse data strings in a way that chunk boundaries align with source block boundaries (by setting the chunk length equal to source block length) and achieve asymptotically optimal performance under mild conditions. However, when symbols have different lengths, the loss of synchronization leads to poor performance. For instance, [77] considered the scenario in which there are $A = 2$ source symbols, with the source symbol length distribution $\mathbb{P}_l$ assigning equal probability to L and $L+1$ (here L is an independent parameter rather than the expected value of $\mathbb{P}_l$) and with $B = 3L$ source blocks. FLD with chunk length $\ell = L$ was shown to satisfy $\frac{\mathbb{E}[\mathcal{L}_F(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))} \geq \Omega(B)$. In the case where copies are not exact, the question of interest is then whether fixed-length deduplication can still perform well when chunk boundaries align with repeat boundaries. To answer this question, we need to ensure that the two groups of boundaries are aligned. So we consider only source models where source symbols all have the same length L ($\mathbb{P}_l$ is degenerate). The first-stage parsing length of mFLD (including AFLD) and the chunk length of EDD are both assumed to be equal to L.

We present a lemma that will be used frequently. For positive integers m, ℓ and $\delta \in \big(0, \frac{1}{2}\big)$, define

$$\mathcal{S}_\delta(\ell, m) = \sum_{t=0}^{\ell} \binom{\ell}{t} \min\big(1, m\delta^t (1-\delta)^{\ell - t}\big). \tag{35}$$

Lemma 8. *Let $\boldsymbol{r}$ be a string drawn uniformly at random from Σ^ℓ. Let $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m$ be m iid descendants of $\boldsymbol{r}$ by δ-edit and let $\boldsymbol{r}_{[m]} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m\}$. For any $\boldsymbol{w} \in \Sigma^\ell$, let $\boldsymbol{w} \in \boldsymbol{r}_{[m]}$ denote the event that $\boldsymbol{w} = \boldsymbol{r}_i$ for some i. Then*

$$\frac{1}{2} \frac{\mathcal{S}_\delta(\ell, m)}{2^\ell} \leq \Pr\big(\boldsymbol{w} \in \boldsymbol{r}_{[m]}\big) \leq \frac{\mathcal{S}_\delta(\ell, m)}{2^\ell}, \tag{36}$$

and thus the expected number of unique strings in $\boldsymbol{r}_{[m]}$ is bounded between $\frac{1}{2}\mathcal{S}_\delta(\ell, m)$ and $\mathcal{S}_\delta(\ell, m)$.

Furthermore, $\mathcal{S}_\delta(\ell, m)$ takes the following values for different values of ℓ and m:

- *If $\ell \geq \frac{\log m}{H(\delta)}$, then*

$$\mathcal{S}_\delta(\ell, m) \geq \frac{1}{4} m. \tag{37}$$

In particular if $\ell \geq \frac{\log m}{\log(\frac{1}{1-\delta})}$, then

$$\mathcal{S}_\delta(\ell, m) = m. \tag{38}$$

- *If $\ell \leq \frac{\log m}{H(\frac{1}{2}, \delta)}$, then*

$$\mathcal{S}_\delta(\ell, m) \geq 2^{\ell - 1}. \tag{39}$$

In particular if $\ell \leq \frac{\log m}{\log(\frac{1}{\delta})}$, then

$$\mathcal{S}_\delta(\ell, m) = 2^\ell. \tag{40}$$

- *For any $\delta < \delta' < \frac{1}{2}$,*

$$\mathcal{S}_\delta(\ell, m) \leq 2^{\ell H(\delta')} + m 2^{-\ell D(\delta' || \delta)}. \tag{41}$$

In particular if $\ell = \frac{\log m}{H(\delta',\delta)}$, *then*

$$\mathcal{S}_\delta(\ell,m) \le 2^{\ell H(\delta')} + m2^{-\ell D(\delta'||\delta)} = 2^{\ell H(\delta')+1}. \tag{42}$$

- *For any values of* ℓ *and* m*,*

$$\mathcal{S}_\delta(\ell,m) \le \min\bigl(2^{\ell}, m\bigr). \tag{43}$$

The proof of Lemma 8 is presented in Appendix A1.

Modified and adaptive fixed-length deduplication

We show that, even with knowledge of the source symbol length, if the chunk length is not properly chosen, mFLD encodes $\boldsymbol{s}$ with a constant number of bits per symbol regardless of δ. Therefore, the ratio $\frac{\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))}$ can be arbitrarily large for small δ. Meanwhile for AFLD, with the adaptive chunk length $\ell = \left\lceil \frac{\log(B/A)}{H(\gamma,\delta)} \right\rceil$, the ratio $\frac{\mathbb{E}[\mathcal{L}_{AF}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))}$ is shown to be upper bounded by a constant for all δ and for γ properly chosen.

Consider the two-stage parsing of $\boldsymbol{s}$ with $D = L$. The length-D segments after the first-stage parsing are exactly the source blocks $Y_1, Y_2, \ldots, Y_B$. Let $C = \lfloor L/\ell \rfloor$ and $r = L - C\ell$. Each Y_b, $1 \le b \le B$, is then parsed into chunks $Z_1^b, Z_2^b, \ldots, Z_{C+1}^b$ with $\left|Z_c^b\right| = \ell$ for all $c \le C$ and $\left|Z_{C+1}^b\right| = r$ (see Figure 1). If we also divide each source symbol X_a into substrings of length ℓ as $\mathsf{X}_a = U_1^a U_2^a \cdots U_{C+1}^a$, then for all $1 \le c \le C+1$, $\{Z_c^b\}_{b \in Y(a)}$ are iid δ-edit descendants of U_c^a.

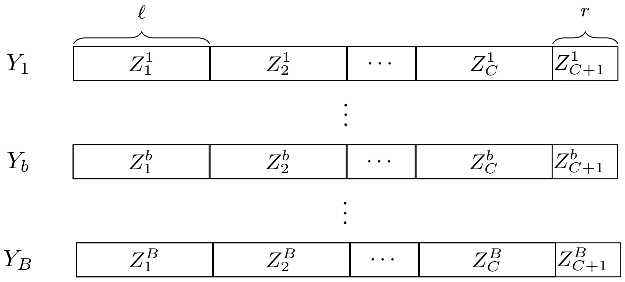


Figure 1: Modified fixed-length chunking with segment length $D = L$ and chunk length ℓ.

Before performing a detailed evaluation of the algorithm, let us first provide a rough analysis for a special case, which will provide some insights into the general problem. Suppose the alphabet $\mathcal{X}$ only has a single symbol X of length L, whose ℓ-prefix is denoted by U_1. We consider encoding *only* the set $Z_1^1, Z_1^2, \ldots, Z_1^B$, where each Z_1^b is a descendant of U_1 by δ-edit. The expected size of the dictionary, i.e., the number of distinct ℓ-strings in $\{Z_1^1, Z_1^2, \ldots, Z_1^B\}$, by Lemma 8 is approximately

$$S := \mathcal{S}_\delta(\ell, B) = \sum_{t=0}^{\ell} \min\left(\binom{\ell}{t}, \binom{\ell}{t} B\delta^t(1-\delta)^{\ell-t}\right). \tag{44}$$

We can interpret (44) as follows. At a given distance t from U_1, there are $\binom{\ell}{t}$ sequences of length ℓ. Further, if we generate B sequences, the expected number of sequences at distance t is $\binom{\ell}{t}B\delta^t(1-\delta)^{\ell-t}$. The number of sequences in the dictionary at distance t is then approximated by the minimum of the two terms. (This analysis of S is helpful whenever $\mathcal{S}_\delta(\cdot,\cdot)$ appears in the sequel as well.)

We would like S to be small enough that $\log S \ll \ell$ (so that pointers to the dictionary have much smaller lengths than the sequences being encoded) and $S \ll B$ (so that each sequence in the dictionary is repeated many times).[2] As t ranges from 0 to ℓ in the sum in (44), the term $\binom{\ell}{t}$ attain its maximum at $t \simeq \ell/2$ while the second term inside the min attains its maximum at $t \simeq \ell\delta$. We investigate which term determines the behavior of the sum. Let $\ell = \frac{\log B}{H(\gamma,\delta)}$ for a constant $0 \le \gamma \le 1$. Note that since $\delta < \frac{1}{2}$, $H(\gamma,\delta)$ and ℓ are increasing and decreasing functions of γ, respectively. With this choice, $B\delta^t(1-\delta)^{\ell-t} \ge 1$ for $t \le \ell\gamma$ and $B\delta^t(1-\delta)^{\ell-t} \le 1$ for $t \ge \ell\gamma$.

- If $\gamma < \delta$, then $B\delta^{\delta\ell}(1-\delta)^{(1-\delta)\ell} < 1$, and $S \ge \sum_{t=\lceil\gamma\ell\rceil}^{\ell} \binom{\ell}{t} B\delta^t(1-\delta)^{\ell-t} \ge B(1-2^{-\ell D(\gamma||\delta)})$. In this case, almost all Z_1^b are distinct and thus not compressible.

[2] Note that the size of the dictionary, and hence the length of the pointers, vary as the encoding progresses; we ignore this fact for now and approximate pointer lengths based on the final size of the dictionary.

- If $\gamma = \delta$, then $\ell = \frac{\log B}{H(\delta)}$, and $S \geq \frac{B}{4}$ by (37). In this case, a constant fraction of Z_1^b are distinct and thus not compressible.
- If $\gamma \geq 1/2$, then $\ell \leq \frac{\log B}{H(\frac{1}{2},\delta)}$, and $S \geq 2^{\ell-1}$ by (39). In this case, due to the fact that ℓ is chosen too small, the dictionary is so large that pointers to the dictionary are as long as the chunks and there is no compression gain.
- If $\delta < \gamma < 1/2$, then by (42),

$$S \leq 2^{\ell H(\gamma)+1}. \tag{45}$$

  Hence, pointers have an approximate length of $\ell H(\gamma)$ and are smaller than ℓ by a factor of $\frac{1}{H(\gamma)}$. Furthermore, each sequence is repeated approximately $2^{\ell D(\gamma||\delta)}$ times since $B = 2^{\ell H(\gamma,\delta)}$. The number of bits required to encode the dictionary is $2\ell 2^{\ell H(\gamma)}$, which is negligible compared to $B\ell$, the length of the uncoded sequences since $\gamma \neq \delta$. Hence, we can encode $\{Z_1^1, \ldots, Z_1^B\}$ using essentially $B\ell H(\gamma)$ bits, achieving a compression ratio of $\frac{1}{H(\gamma)}$.

This analysis highlights that ℓ should be chosen appropriately to avoid a large dictionary or a situation in which there are no repetitions in the sequence. If these conditions are satisfied, then we can successfully deduplicate the data, as shown rigorously in Theorem 15 for AFLD.

Now we return to the general setting. It can be seen from the description of mFLD that the compressed string is composed of two parts: the bits used to encode the chunks at their first occurrences and the bits used to encode repeated chunks by pointers to the dictionary. For both parts, our first step is to compute the expected size of the dictionary, i.e., the number of distinct chunks, for which we present Lemma 9 and Lemma 10.

Lemma 9. *Suppose K strings of length n are chosen independently and uniformly from Σ^n. Assume each string produces at least m_1 and at most m_2 descendants by δ-edits. For any string $\boldsymbol{w}$ with $|\boldsymbol{w}| = n$, let $G_{\boldsymbol{w}}$ denote the event that $\boldsymbol{w}$ equals one or more descendants. Then*

$$\frac{1}{2}\min\biggl(1, \frac{1}{2}K\frac{\mathcal{S}_\delta(n, m_1)}{2^n}\biggr) \leq \Pr(G_{\boldsymbol{w}}) \leq \min\biggl(1, K\frac{\mathcal{S}_\delta(n, m_2)}{2^n}\biggr). \tag{46}$$

The proof of Lemma 9 is presented in Appendix A2. This lemma considers the probability of observing a string $\boldsymbol{w}$ when multiple random strings produce δ-edit descendants simultaneously. This setting models exactly our source string generation process where the A source symbols correspond to K random strings, and the source blocks correspond to the δ-edit descendants. In particular, $\mathcal{E}_u$ being true corresponds to $m_2 = \frac{3B}{2A}$ and $\mathcal{E}_l$ being true corresponds to $m_1 = \frac{B}{4A}$.

Let $T_F^1(\boldsymbol{s})$ denote the dictionary after all chunks of $\boldsymbol{s}$ are processed, i.e., $T_F^1(\boldsymbol{s})$ contains all distinct strings in $\{Z_c^b\}_{b,c}$. Let $T_F^{1/2}(\boldsymbol{s})$ denote the dictionary immediately after all chunks in the first half of $\boldsymbol{s}$, i.e., $Y_1 Y_2 \cdots Y_{\lceil B/2 \rceil}$, are processed. We apply Lemma 9 to find bounds on the sizes of $T_F^1(\boldsymbol{s})$ and $T_F^{1/2}(\boldsymbol{s})$ in the following lemma.

Lemma 10. *Consider the two-stage fixed-length chunking process with first-stage parsing length $D = L$ and chunk length ℓ. The dictionary sizes $T_F^1(\boldsymbol{s})$ and $T_F^{1/2}(\boldsymbol{s})$ satisfy*

$$\mathbb{E}\bigl[\bigl|T_F^1(\boldsymbol{s})\bigr||\mathcal{E}_u\bigr] \leq \min\biggl(2^\ell, AC\mathcal{S}_\delta\Bigl(\ell, \frac{3B}{2A}\Bigr)\biggr) + B, \tag{47}$$

$$\mathbb{E}\Bigl[\Bigl|T_F^{1/2}(\boldsymbol{s})\Bigr||\mathcal{E}_l\Bigr] \geq \frac{1}{2}\min\biggl(2^\ell, \frac{1}{2}AC\mathcal{S}_\delta\Bigl(\ell, \frac{B}{4A}\Bigr)\biggr). \tag{48}$$

The proof of Lemma 10 is presented in Appendix A2.

Next, we show using Lemma 10 that if ℓ is chosen too small relative to the scale of the system, then mFLD spends a constant number of bits per symbol. The proof strategy is as follows: with ℓ small enough, the term $\min\bigl(2^\ell, \frac{1}{2}AC\mathcal{S}_\delta\bigl(\ell, \frac{B}{4A}\bigr)\bigr)$ in (48) equals 1, which makes $\mathbb{E}\Bigl[\Bigl|T_F^{1/2}(\boldsymbol{s})\Bigr||\mathcal{E}_l\Bigr]$ greater than $2^{\ell-1}$. Therefore, when encoding duplicated chunks using pointers, each pointer takes approximately ℓ bits and there is no compression gain.

Theorem 11. *Consider the source model in which source symbols have the same length L. For mFLD with first-stage parsing length $D = L$ and chunk length ℓ, if $\ell 2^\ell = O(AL)$ or $\ell \leq \frac{\log(B/A)-2}{H(\frac{1}{2},\delta)}$, then*

$$\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})] \geq \frac{1}{12}BL(1+o(1)), \quad \textit{as } B \to \infty, \tag{49}$$

where the $o(1)$ term is independent of δ.

Proof: We first claim, to be proved later, that if $\ell 2^\ell = O(AL)$ or $\ell \le \frac{\log(B/A)-2}{H(\frac{1}{2},\delta)}$, then

$$\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \ge 2^{\ell-1}. \tag{50}$$

It follows from Markov's inequality that

$$\Pr\bigg(2^\ell - \Big|T_F^{1/2}(\boldsymbol{s})\Big| \ge \frac{3}{4}\cdot 2^\ell |\mathcal{E}_l\bigg) \le \frac{\frac{1}{2}\cdot 2^\ell}{\frac{3}{4}\cdot 2^\ell} = \frac{2}{3}, \tag{51}$$

which is equivalent to

$$\Pr\bigg(\Big|T_F^{1/2}(\boldsymbol{s})\Big| \ge \frac{2^\ell}{4} |\mathcal{E}_l\bigg) \ge \frac{1}{3}. \tag{52}$$

Next, we consider the second half of $\boldsymbol{s}$, $Y_{\lceil B/2\rceil+1}\cdots Y_B$. There are $\lfloor B/2 \rfloor C$ chunks of length ℓ, and encoding each of them takes at least either ℓ or $\log\Big|T_F^{1/2}(\boldsymbol{s})\Big|$ bits plus an additional bit indicating whether the chunk is stored in full or represented by a pointer. So in total, we need at least

$$\Big(\min\Big(\ell, \log\Big|T_F^{1/2}(\boldsymbol{s})\Big|\Big) + 1\Big)\cdot \left\lfloor\frac{B}{2}\right\rfloor C \tag{53}$$

bits. It follows that for B sufficiently large,

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})|\mathcal{E}_l] &\ge \mathbb{E}\bigg[\Big(\min\Big(\ell, \log\Big|T_F^{1/2}(\boldsymbol{s})\Big|\Big) + 1\Big)\cdot \left\lfloor\frac{B}{2}\right\rfloor C|\mathcal{E}_l\bigg] &(54)\\ &\ge \frac{1}{3}\bigg(\min\bigg(\ell, \log\frac{2^\ell}{4}\bigg) + 1\bigg)\cdot \left\lfloor\frac{B}{2}\right\rfloor C &(55)\\ &\ge \frac{BL}{12}(1+o(1)), &(56)\end{aligned}$$

where the second inequality follows from (52).

Finally, since (32) gives that $\Pr(\mathcal{E}_l) = 1 + o(1)$, we get

$$\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})] \ge \mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})|\mathcal{E}_l]\Pr(\mathcal{E}_l) \ge \frac{BL}{12}(1+o(1)). \tag{57}$$

It remains to prove the claim: $\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \ge 2^{\ell-1}$ when $\ell 2^\ell = O(AL)$ or $\ell \le \frac{\log(B/A)-2}{H(\frac{1}{2},\delta)}$. Consider the case when $\ell 2^\ell = O(AL)$. For sufficiently large B (and thus A and L), $\frac{B}{4A} \ge \frac{4\ell 2^\ell}{AL}$. Therefore, by Lemma 10,

$$\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \ge \frac{1}{2}\min\bigg(2^\ell, \frac{1}{2}AC\mathcal{S}_\delta\bigg(\ell, \frac{B}{4A}\bigg)\bigg) \ge \frac{1}{2}\min\bigg(2^\ell, \frac{1}{2}AC\mathcal{S}_\delta\bigg(\ell, \frac{4\ell 2^\ell}{AL}\bigg)\bigg), \tag{58}$$

where the last inequality follows from the fact that $\mathcal{S}_\delta(\ell, m)$ is non-decreasing in m. By (38), if $m(1-\delta)^\ell \le 1$, then $\mathcal{S}_\delta(\ell, m) = m$. Since asymptotically $\frac{4\ell 2^\ell}{AL}(1-\delta)^\ell \le 1$,

$$\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \ge 2^{\ell-1}\min\bigg(1, \frac{4\ell C}{2L}\bigg) \ge 2^{\ell-1}, \tag{59}$$

where the last step follows from the fact that $C \ge \frac{L}{2\ell}$.

When $\ell \le \frac{\log(B/A)-2}{H(\frac{1}{2},\delta)}$, again by Lemma 10,

$$\begin{aligned}\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] &\ge \frac{1}{2}\min\bigg(2^\ell, \frac{1}{2}AC\mathcal{S}_\delta\bigg(\ell, \frac{B}{4A}\bigg)\bigg) &(60)\\ &\ge 2^{\ell-1}\min\bigg(1, \frac{AC}{4}\bigg) \ge 2^{\ell-1}, &(61)\end{aligned}$$

where the second inequality follows from (39) that when $\ell \le \frac{\log m}{H(\frac{1}{2},\delta)}$, $\mathcal{S}_\delta(\ell, m) \ge 2^{\ell-1}$. ■

The preceding theorem shows that when ℓ is chosen too small, the size of the dictionary will be of order 2^ℓ. Specifically, if $\ell 2^\ell = O(AL)$, the number of distinct ℓ-substrings in the source alphabet is already of order 2^ℓ. If $\ell \le \frac{\log(B/A)-2}{H(\frac{1}{2},\delta)}$, then the δ-edits are able to produce almost all ℓ-strings instead of only producing strings that are on the $\delta\ell$ Hamming sphere.

In the next theorem, we show that if ℓ is chosen too large, then mFLD again spends a constant number of bits per symbol.

The proof strategy is to show that if ℓ is chosen too large, then almost every chunk is distinct, thus making the source string incompressible.

Theorem 12. *Consider the source model in which source symbols have the same length L. For mFLD with first-stage parsing length $D = L$ and chunk ℓ, if $\ell \geq \frac{\log(B/A)-2}{H(\delta)}$, then*

$$\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})] \geq \frac{1}{128} BL(1+o(1)), \quad \text{as } B \to \infty, \tag{62}$$

where the $o(1)$ term is independent of δ.

Proof: When $\ell \leq \frac{\log(B/A)-2}{H(\frac{1}{2},\delta)}$,

$$\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \geq \frac{1}{2}\min\Big(2^\ell, \frac{1}{2}AC\mathcal{S}_\delta\Big(\ell, \frac{B}{4A}\Big)\Big) \geq 2^{\ell-1}\min\Big(1, \frac{1}{2}AC\cdot\frac{1}{4}\cdot\frac{B}{4A2^\ell}\Big) \tag{63}$$

$$= 2^{\ell-1}\min\Big(1, \frac{BC}{32\cdot 2^\ell}\Big), \tag{64}$$

where the first inequality follows from Lemma 10 and the second from (37).

In the case where $1 \leq \frac{BC}{32\cdot 2^\ell}$ and hence $\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \geq 2^{\ell-1}$, the proof follows from the discussion that follows (50). So it remains to consider the case when $\frac{BC}{32\cdot 2^\ell} \leq 1$, i.e.,

$$\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] \geq 2^{\ell-1}\cdot\frac{BC}{32\cdot 2^\ell} = \frac{BC}{64}. \tag{65}$$

Since it takes $\ell+1$ bits to store distinct chunks in the dictionary,

$$\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})|\mathcal{E}_l] \geq (\ell+1)\mathbb{E}\Big[\Big|T_F^{1/2}(\boldsymbol{s})\Big| |\mathcal{E}_l\Big] = \ell\frac{B\lfloor L/\ell\rfloor}{64} \tag{66}$$

$$\geq \frac{1}{64}B\max(\ell, L-\ell) \geq \frac{1}{128}BL. \tag{67}$$

The desired result thus follows again from $\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})] \geq \mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})|\mathcal{E}_l]\Pr(\mathcal{E}_l)$ and the fact that $\Pr(\mathcal{E}_l) = 1+o(1)$. ■

Theorems 11 and 12 imply Corollaries 13 and 14, shown as follows.

Corollary 13. *Consider the source model in which source symbols all have length L. For mFLD with $D = L$, if the chunk length $\ell = o(\log B) \cup \omega(\log B)$, the compression ratio $\frac{\mathbb{E}[|\boldsymbol{s}|]}{\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})]}$ is upper bounded by a universal constant for any edit probability $\delta > 0$.*

Corollary 13 characterizes the performance of mFLD when the chunk length ℓ is chosen too small or too large. With the chunk length improperly chosen, the average length of the compressed strings is always at least a constant factor of the original length, regardless of the edit probability δ. This is not desirable for small δ since, as δ goes to 0, the entropy gets smaller and the ratio $\frac{\mathbb{E}[\mathcal{L}_{mF}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))}$ grows unboundedly. It can be seen from the proofs of Theorems 11 and 12 that when the chunk length is chosen too small, the dictionary becomes so large that the pointers become of similar lengths to the chunks. On the other hand, when the chunk length is chosen too large, repeats can not be identified and deduplication thus fails. It is therefore important to pick a suitable chunk length when implementing deduplication algorithms in practice.

If we pick $\ell = L$, mFLD becomes FLD with chunk length equal to source symbol length, which was shown in [77] to be asymptotically optimal on sources with fixed symbol length and no edits. However, in the case when edit probability δ is nonzero, since we assume $L = \Theta\big(B^{k_1}\big)$, Corollary 13 implies that the compression ratio of FLD is bounded and the gap between FLD and entropy can be arbitrarily large.

Corollary 14. *Consider the source model in which source symbols all have length L. For FLD, with chunk length L, the compression ratio $\frac{\mathbb{E}[|\boldsymbol{s}|]}{\mathbb{E}[\mathcal{L}_F(\boldsymbol{s})]}$ is upper bounded by a universal constant for any edit probability $\delta > 0$.*

Next, we show that with the adapted chunk length, AFLD can achieve performance within a constant factor of optimal.

Theorem 15. *Consider the source model in which source symbols have the same length L. The performance of AFLD with*

$D = L$ and $\ell = \left\lceil \frac{\log(B/A)}{H(\gamma,\delta)} \right\rceil$ satisfies

$$1 \le \frac{\mathbb{E}[\mathcal{L}_{AF}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))} \le \frac{1+k_1}{k_2} \cdot \frac{H(\gamma,\delta)}{H(\delta)} \cdot (1+o(1)), \tag{68}$$

as $B \to \infty$, for any $\gamma \in (\delta, \frac{1}{2})$.

Proof: We first note that the length of $\boldsymbol{s}$ can be encoded in at most $2\log(|\boldsymbol{s}|)+3$ bits with Elias gamma coding.

The number of bits used to encode chunks at their first occurrences is upper bounded by $\left|T_F^1(\boldsymbol{s})\right|(\ell+1)$ since chunks are all of lengths less than or equal to ℓ. Consider the upper bound on $\mathbb{E}\big[\left|T_F^1(\boldsymbol{s})\right| |\mathcal{E}_u\big]$ in Lemma 10. Note that by (41) and $\frac{B}{A} \le 2^{\ell H(\gamma,\delta)}$ with our choice of ℓ,

$$\mathcal{S}_\delta\left(\ell, \frac{3B}{2A}\right) \le 2^{\ell H(\gamma)} + \frac{3B}{2A} 2^{-\ell D(\gamma||\delta)} \le \frac{5}{2} \cdot 2^{\ell H(\gamma)}. \tag{69}$$

It follows that

$$\begin{aligned}
\mathbb{E}\big[\left|T_F^1(\boldsymbol{s})\right| |\mathcal{E}_u\big](\ell+1) &\le \left(\min\left(2^\ell, AC\mathcal{S}_\delta\left(\ell, \frac{3B}{2A}\right)\right) + B\right)(\ell+1) && (70)\\
&\le \min\left(2^\ell, \frac{5AC}{2} \cdot 2^{\ell H(\gamma)}\right)(\ell+1) + B(\ell+1) && (71)\\
&\le \frac{5AL}{2\ell} \cdot 2^{\ell H(\gamma)} \cdot (\ell+1) + B(\ell+1) && (72)\\
&= \frac{5}{2} AL \left(\frac{B}{A}\right)^{H(\gamma)/H(\gamma,\delta)} \left(1 + \Theta\left(\frac{1}{\log(B/A)}\right)\right) + \Theta(B\log B) && (73)\\
&= o(BL), && (74)
\end{aligned}$$

where the last equality follows from $\frac{H(\gamma)}{H(\gamma,\delta)} < 1$ and thus $B^{\frac{H(\gamma)}{H(\gamma,\delta)}} A^{1-\frac{H(\gamma)}{H(\gamma,\delta)}} = o(B)$.

Next, we derive an upper bound on the number of bits used by pointers for encoding repeated chunks. There are $(C+1)B$ chunks and the number of bits needed for encoding one pointer is at most $\log(BL)+1$. So in total, the number of bits we need is at most

$$\begin{aligned}
(C+1)B(\log(BL)+1) \le (L+\ell)B\frac{\log(BL)+1}{\ell} &\le \frac{BL}{\ell}\log(BL)\left(1 + O\left(\frac{1}{\log B}\right)\right) && (75)\\
&\le H(\gamma,\delta)BL \cdot \frac{\log(BL)}{\log(B/A)}. && (76)
\end{aligned}$$

Combining (74), (76), and including the number of bits used for encoding the length of $\boldsymbol{s}$ by Elias coding, we get

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_{AF}(\boldsymbol{s})|\mathcal{E}_u] &\le H(\gamma,\delta)BL \cdot \frac{\log(BL)}{\log(B/A)} + o(BL) && (77)\\
&\le H(\gamma,\delta)BL\frac{1+k_1}{k_2}(1+o(1)), && (78)
\end{aligned}$$

by noting that $\frac{\log(BL)}{\log(B/A)} \le \frac{1+k_1}{k_2}(1+o(1))$.

On the complement of $\mathcal{E}_u$, the number of bits needed for storing the dictionary is at most $2BL$ since the lengths of chunks in total is at most BL and there are at most BL chunks. The number of bits for encoding repeated chunks by pointers is at most $BL(\log(BL)+1)$. It follows that

$$\mathbb{E}\big[\mathcal{L}_{AF}(\boldsymbol{s})|\bar{\mathcal{E}}_u\big]\Pr\big(\bar{\mathcal{E}}_u\big) \le (2ABL + 2\log(BL) + 3)\log(BL)e^{-\frac{B}{10A}} = o(1). \tag{79}$$

The desired result thus follows from

$$\mathbb{E}[\mathcal{L}_{AF}(\boldsymbol{s})] = \mathbb{E}[\mathcal{L}_{AF}(\boldsymbol{s})|\mathcal{E}_u]\Pr(\mathcal{E}_u) + \mathbb{E}\big[\mathcal{L}_{AF}(\boldsymbol{s})|\bar{\mathcal{E}}_u\big]\Pr\big(\bar{\mathcal{E}}_u\big), \tag{80}$$

and the fact that $\Pr(\mathcal{E}_u) = 1 + o(1)$. ■

For any $\delta < \frac{1}{2}$ and $a > 1$, we can find γ in the range $(\delta, \frac{1}{2})$ such that $H(\gamma,\delta)/H(\delta) \le a$. It thus follows from Theorem 15 that adaptive fixed-length deduplication can compress the sequence within a constant factor of the entropy, as stated in Corollary 16.

Corollary 16. *For any edit probability $\delta \in (0, \frac{1}{2})$ and any $a > 1$, there exists $\delta < \gamma < \frac{1}{2}$ such that*

$$\frac{\mathbb{E}[\mathcal{L}_{AF}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))} \leq \frac{a(1+k_1)}{k_2}(1+o(1)). \tag{81}$$

With k_1, k_2 being fixed constants, the preceding corollary states that AFLD achieves a constant factor of optimal for any edit probability δ. Thus, to achieve high compression ratio, deduplication algorithm parameters, especially the chunk length, should be chosen based on the data. In practice, it can thus be beneficial to first obtain an estimate of the parameters of the data and then apply deduplication with algorithm parameters properly chosen. A fixed chunk length is unlikely to be universally effective for all datasets.

Edit-distance deduplication

Next, we study the edit-distance deduplication algorithm. EDD identifies positions in which the current chunk and previously observed similar chunks differ. We show that with chunk length being equal to source symbol length, EDD can achieve a constant factor of optimal.

Theorem 17. *Consider the source model in which source symbols have the same length L and the edit probability is $\delta < \frac{1}{4}$. The performance of edit-distance deduplication with chunk length $\ell = L$ and mismatch ratio β satisfies*

$$1 \leq \frac{\mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))} \leq \frac{H(2\beta)}{H(\delta)}(1+o(1)), \quad \textit{as } B \to \infty, \tag{82}$$

for any $\delta < \beta \leq \frac{1}{4}$.

Proof: With $\ell = L$, the B source blocks, $Y_1, \ldots, Y_B$, are parsed as chunks. We know that each Y_b is a descendant of one of the source symbols. Let $\mathcal{E}_d$ denote the event that every source block Y_b is within Hamming distance βL from its ancestor. By the Chernoff bound, the probability that more than βL symbols of a source symbol are flipped in a δ-edit is at most $2^{-D(\beta||\delta)L}$. We then apply the union bound and get $\Pr(\mathcal{E}_d) \geq 1 - B2^{-D(\beta||\delta)L}$.

When $\mathcal{E}_d$ holds, the source blocks are covered by A Hamming balls of radius βL. Therefore, with mismatch ratio β, the dictionary is of size at most A, and takes $A(L+1)$ bits to store. The pointer length is thus upper bounded by $\log A + 1$. The difference with the referenced chunk can be encoded in at most $H(2\beta)L + 1$ bits. Including the $2\log(BL) + 3$ bits for encoding $|\boldsymbol{s}|$ at the beginning, we get

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})|\mathcal{E}_d] &\leq 2\log(BL) + 3 + A(L+1) + (1 + \log A + 1 + H(2\beta)L + 1)B \quad (83)\\ &= H(2\beta)BL + o(BL). \quad (84)\end{aligned}$$

When the complement of $\mathcal{E}_d$ holds, we trivially upper bound dictionary size by B. It follows that

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})|\bar{\mathcal{E}_d}] &\leq 2\log(BL) + 3 + B(L+1) + (1 + \log B + 1 + H(2\beta)L + 1)B \quad (85)\\ &\leq 2BL. \quad (86)\end{aligned}$$

Thus,

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})] &= \Pr(\mathcal{E}_d)\mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})|\mathcal{E}_d] + \Pr(\bar{\mathcal{E}_d})\mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})|\bar{\mathcal{E}_d}] \quad (87)\\ &\leq H(2\beta)BL(1+o(1)) + 2B^2L2^{-D(\beta||\delta)L} \quad (88)\\ &= H(2\beta)BL(1+o(1)), \quad (89)\end{aligned}$$

where the term $2B^2L2^{-D(\beta||\delta)L}$ is absorbed into the $o(1)$ term since $D(\beta||\delta) > 0$. ■

Note that for any $\delta < \frac{1}{4}$, we can always find β larger than but close enough to δ such that $\frac{H(2\beta)}{H(\delta)}$ is upper bounded by a constant value. With such choices of β, the preceding theorem states that $\mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})]$ is at most a constant factor of $H(\mathcal{I}_b(\delta))$. As an example, let $\beta = \min\left(\frac{3\delta}{2}, \frac{1}{4}\right)$. The ratio $\frac{H(2\beta)}{H(\delta)}$ is upper bounded by

$$\frac{H(2\beta)}{H(\delta)} \leq \frac{H(\min(3\delta, 1/2))}{H(\delta)} \leq 3, \tag{90}$$

where the last inequality follows from the fact that $\frac{H(3p)}{H(p)} \leq 3$ for all $p \leq \frac{1}{3}$ and $H(\frac{1}{6}) \leq \frac{1}{2}$. Hence, EDD also achieves a constant factor of optimal, as formalized in the following corollary.

Corollary 18. *Consider the source model in which source symbols have the same length L and edit probability $\delta < \frac{1}{4}$. There exists a mismatch ratio β such that the performance of EDD with chunk length $\ell = L$ satisfies*

$$\frac{\mathbb{E}[\mathcal{L}_{ED}(\boldsymbol{s})]}{H(\mathcal{I}_b(\delta))} \leq \frac{H(3\delta)}{H(\delta)}(1+o(1)) \leq 3(1+o(1)). \tag{91}$$

We note however that EDD is more complex than AFLD as it identifies chunks that are within a certain Hamming distance.

2) Deduplication in the variable-length scheme

In this section, we study the variable-length deduplication algorithm, which is more widely applicable than the algorithms in the fixed-length scheme and does not require the source symbol lengths to be the same or known. In the previous section, we saw that for AFLD to achieve optimality, the chunk length should be adapted to the source. Similarly for VLD, the performance depends on chunk lengths which in turn depend on the length of the marker M.

Before presenting the detailed analysis, we provide some insights on how the marker length M affects the distribution of chunk contents. In variable-length chunking, the chunks (except perhaps the last one) end with the marker string 0^M. We write $\boldsymbol{s} = U_1 0^M U_2 0^M \cdots 0^M U_N$, where each $U_n, n < N$, is either empty or of the form $\boldsymbol{u}1$ for some M-RLL string $\boldsymbol{u}$. We can approximately treat $\boldsymbol{s}$ as a Bernoulli(1/2) process for now. The lengths of strings U_n are thus equivalent to the stopping time in an infinite-length Bernoulli(1/2) process untill the beginning of the first occurrence of 0^M, which is of expected length approximately 2^M. The behavior of VLD with marker length M is thus similar to that of mFLD with chunk length 2^M. When M is chosen so small that the number N of chunks becomes much larger than the total number of strings of lengths around 2^M, the dictionary becomes exhaustive and pointers have similar lengths to chunks. When M is chosen too large, most of $U_1, \ldots, U_N$ are distinct and thus not compressible. In the following, we study in detail how $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ varies for different values of M.

Similar to the fixed-length schemes, the dictionary size is an essential first-step in computing $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$. To determine the expected dictionary size, we again start with the probability of occurrences of chunks. However, now the chunks are of different lengths and the occurrences are not restricted to a fixed set of positions. So we bound the probability of occurrences of a chunk by the probability of occurrences of certain substrings. Specifically, we consider strings of the forms $10^M\boldsymbol{u}10^M$ or $0^M\boldsymbol{u}10^M$ ($\boldsymbol{u} \in R_M$): Except the first and the last chunks, the probability of occurrence of chunk $\boldsymbol{u}10^M$ is greater than the probability of occurrence of a substring $10^M\boldsymbol{u}10^M$ since the prefix 10^M always marks an ending of the previous chunk; similarly, the probability of occurrence of chunk $\boldsymbol{u}10^M$ is less than or equal to the probability of occurrence of a substring $0^M\boldsymbol{u}10^M$ since any occurrences of chunk $\boldsymbol{u}10^M$ must follow a 0^M which is the ending marker of the previous chunk.

Let $\boldsymbol{w} \in Y_1^B$ denote the event that $\boldsymbol{w}$ appears as a substring of Y_b for some $1 \leq b \leq B$ and let $\boldsymbol{w} \in Y_1^{B/2}$ denote the event that $\boldsymbol{w}$ appears as a substring of Y_b for some $1 \leq b \leq \lceil B/2 \rceil$.[3] We first present in Lemmas 19, 20 and 21 two lower bounds on $\boldsymbol{w} \in Y_1^{B/2}$ and an upper bound on $\boldsymbol{w} \in Y_1^B$.

Lemma 19. *Suppose K strings of length n are chosen independently and uniformly from Σ^n. Assume each string produces at least m_1 and at most m_2 descendants by δ-edits. For any string $\boldsymbol{w}$ with $|\boldsymbol{w}| \leq n$, let $H_{\boldsymbol{w}}$ denote the event that $\boldsymbol{w}$ appears as a substring of one or more descendants. Then,*

$$\frac{1}{2}\min\left(1, \frac{1}{2}\left\lfloor\frac{n}{|\boldsymbol{w}|}\right\rfloor K\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_1)}{2^{|\boldsymbol{w}|}}\right) \leq \Pr(H_{\boldsymbol{w}}) \leq \min\left(1, (n-|\boldsymbol{w}|+1)K\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_2)}{2^{|\boldsymbol{w}|}}\right). \tag{92}$$

The proof of Lemma 19 is presented in Appendix A3. Similar to Lemma 9, the setting described in Lemma 19 matches the model for the generation of source strings. This time, we allow string $\boldsymbol{w}$ to be any substring of the descendants because chunks can now be in any position of the source string. Note that Lemma 19 is also a generalization of Lemma 9.

Next, we use Lemma 19 to bound the probability of $\boldsymbol{w} \in Y_1^{B/2}$ and $\boldsymbol{w} \in Y_1^B$.

[3] Here we only consider string/chunk occurrences inside source blocks and leave the study of strings/chunks that occur across the boundaries of source blocks for later.

Lemma 20. *Consider the source model with edit probability δ. For any string $\boldsymbol{w} \in \Sigma^*$ with $|\boldsymbol{w}| \leq 2L$,*

$$\Pr(\boldsymbol{w} \in Y_1^B | \mathcal{E}_u) \leq \min\left(1, 2AL \frac{\mathcal{S}_\delta\left(|\boldsymbol{w}|, \frac{3B}{2A}\right)}{2^{|\boldsymbol{w}|}}\right). \tag{93}$$

For any string $\boldsymbol{w} \in \Sigma^*$ with $|\boldsymbol{w}| \leq \lceil \frac{1}{2} L \rceil$,

$$\Pr\Big(\boldsymbol{w} \in Y_1^{B/2} | \mathcal{E}_l\Big) \geq \frac{1}{2} \min\left(1, \frac{AL}{8|\boldsymbol{w}|} \frac{\mathcal{S}_\delta\left(|\boldsymbol{w}|, \frac{B}{4A}\right)}{2^{|\boldsymbol{w}|}}\right). \tag{94}$$

The proof of Lemma 20 is presented in Appendix A3. Although Lemma 20 holds for any string $\boldsymbol{w}$, we will later restrict $\boldsymbol{w}$ to be of the forms $10^M \boldsymbol{u} 10^M$ or $0^M \boldsymbol{u} 10^M$.

Next, we consider another lower bound as an alternative to (94) for the cases when $\boldsymbol{w}$ is of larger lengths. From the proofs of Lemmas 19 and 20, the lower bound (94) is obtained by only taking into account the possibilities of $\boldsymbol{w}$ appearing in non-overlapping positions of each Y_b. Lemma 21 considers every possible substring of Y_b to be equal to $\boldsymbol{w}$ and gets the lower bound by the inclusion-exclusion principle and turns out to be more accurate for $\boldsymbol{w}$ with large lengths. Note that Lemma 21 directly considers $\boldsymbol{w}$ to be of the form $10^M \boldsymbol{u} 10^M$ and the bound is given in the form of a summation.

Lemma 21. *Consider the source model with edit probability $\delta < \frac{1}{2}$. For any n such that $\frac{\log(B/A) - 2}{H(\delta)} \leq n + 2M + 2 \leq \frac{L}{4}$,*

$$\sum_{\boldsymbol{u} \in R_M^n} \Pr\Big(10^M \boldsymbol{u} 10^M \in Y_1^{B/2} | \mathcal{E}_l\Big) \geq \frac{BL}{2^7 \cdot 2^{2M+2}} \cdot \left(1 - \frac{1}{2^{M-1}}\right)^n - \frac{3B^2 L^2}{2^{n+2M+2}}. \tag{95}$$

The proof of Lemma 21 is presented in Appendix A4.

After characterizing the probabilities of strings (and thus chunks) occurring, we consider in Lemma 22 the number of chunks. Let $C_{VL}^M(\boldsymbol{s})$ denote the number of chunks of length over 2^{M-4} in $Y_{\lceil B/2 \rceil + 1} \cdots Y_B$ for variable-length chunking with marker length M. We show that when $2^M = o(L)$, with high probability, $C_{VL}^M(\boldsymbol{s})$ is of order $|\boldsymbol{s}|/2^M$.

Lemma 22. *Consider the source string $\boldsymbol{s} = Y_1 Y_2 \ldots Y_B$. When $2^M = o(L)$, for B, L sufficiently large,*

$$\Pr\left(C_{VL}^M(\boldsymbol{s}) \geq \frac{1}{4} \cdot \left\lfloor \frac{B}{2} \right\rfloor \left(\frac{L}{2^{M+8}} - 1\right)\right) \geq \frac{5}{6}. \tag{96}$$

The proof of Lemma 22 is presented in Appendix A5. It can be seen from the proof that Lemma 22 can be extended to the case when $\mathcal{E}_l$ holds since each source block Y_b by itself is still a Bernoulli(1/2) process. Therefore, the following corollary holds.

Corollary 23. *When $2^M = o(L)$, for B, L sufficiently large,*

$$\Pr\left(C_{VL}^M(\boldsymbol{s}) \geq \frac{1}{4} \cdot \left\lfloor \frac{B}{2} \right\rfloor \left(\frac{L}{2^{M+8}} - 1\right) | \mathcal{E}_l\right) \geq \frac{5}{6}. \tag{97}$$

Next, we use Lemmas 20, 21 and Corollary 23 to bound $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ from below. As marker length M takes different values, different lower bounds of $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ are presented in Theorems 24, 25 and 27. Let $T_{VL}^1(\boldsymbol{s})$ denote the dictionary when all chunks in $\boldsymbol{s}$ are processed and let $T_{VL}^{1/2}(\boldsymbol{s})$ denote the dictionary immediately after chunks in $Y_1 \cdots Y_{\lceil B/2 \rceil}$ are processed.

We first show in Theorem 24 that similar to the fixed-length schemes, small values for M lead to an oversized dictionary.

Theorem 24. *Consider the source model with edit probability δ and the variable-length deduplication algorithm with marker length M. If $2^M = o(\log B)$, then*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \frac{1}{3 \cdot 2^{16}} BL(1 + o(1)), \quad \text{as } B \to \infty, \tag{98}$$

where the $o(1)$ term is independent of δ.

Proof: We show that with high probability, $\left|T_{VL}^{1/2}(\boldsymbol{s})\right|$ is of the order 2^{2^M}. So encoding each chunk in $Y_{\lceil B/2 \rceil + 1} \cdots Y_B$ takes number of bits either equal to the chunk length or pointer length 2^M. We then show using Lemma 22 that the length of the compressed string is a constant fraction of BL.

If a string $\boldsymbol{w}$ of the form $\boldsymbol{w} = 10^M \boldsymbol{u} 10^M$, $\boldsymbol{u} \in R_M$, occurs as a substring of some data block $Y_b, b \leq \lceil \frac{B}{2} \rceil$, then $\boldsymbol{u}10^M$ must be contained in $T_{VL}^{1/2}(\boldsymbol{s})$. For any $\boldsymbol{w} = 10^M \boldsymbol{u} 10^M$ with $|\boldsymbol{u}| \leq 2^M$, by Lemma 20,

$$\Pr\Big(\boldsymbol{w} \in Y_1^{B/2} | \mathcal{E}_l\Big) \geq \frac{1}{2} \min\left(1, \frac{AL}{8|\boldsymbol{w}|} \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}}\right) \tag{99}$$

$$\geq \frac{1}{2} \min\left(1, \frac{AL}{8|\boldsymbol{w}|}\right) \geq \frac{1}{2}, \tag{100}$$

where the second inequality follows from $|\boldsymbol{w}| \leq 2^M + 2M + 2 = o(\log B)$ and the property that $\mathcal{S}_\delta(\ell, m) = 2^\ell$ if $m\delta^\ell \geq 1$.

Denote the set of all M-RLL strings of lengths less than 2^M by $R_M^{\leq 2^M}$. Let $\zeta = \left|\left\{\boldsymbol{u} \in R_M^{\leq 2^M} : 10^M \boldsymbol{u} 10^M \in Y_1^{B/2}\right\}\right|$. Then (100) gives $\mathbb{E}[\zeta | \mathcal{E}_l] \geq |R_M^{\leq 2^M}|/2$ and thus $\mathbb{E}[|R_M^{\leq 2^M}| - \zeta | \mathcal{E}_l] \leq \frac{|R_M^{\leq 2^M}|}{2}$. By Markov inequality, $\Pr(|R_M^{\leq 2^M}| - \zeta \geq 3|R_M^{\leq 2^M}|/4) \leq \frac{2}{3}$ and thus $\Pr(\zeta > |R_M^{\leq 2^M}|/4) \geq \frac{1}{3}$. Noting that $|T_{VL}^{1/2}| \geq \zeta$ and $|R_M^{\leq 2^M}| \geq 2^{2^M - 2}$ by Corollary 2, we get

$$\Pr\Big(\Big|T_{VL}^{1/2}(\boldsymbol{s})\Big| \geq 2^{2^M - 4} | \mathcal{E}_l\Big) \geq \frac{1}{3}. \tag{101}$$

For each chunk in $Y_{\lceil B/2 \rceil + 1} \cdots Y_B$ of length at least 2^{M-4}, we need at least either 2^{M-4} or $\log\Big|T_{VL}^{1/2}(\boldsymbol{s})\Big|$ bits. So by Corollary 23 and inequality (101),

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s}) | \mathcal{E}_l] \geq \mathbb{E}\Big[\min\Big(2^{M-4}, \log\Big|T_{VL}^{1/2}(\boldsymbol{s})\Big|\Big) \cdot C_{VL}^M(\boldsymbol{s}) | \mathcal{E}_l\Big] \tag{102}$$

$$\geq \left(1 - \frac{2}{3} - \frac{1}{6}\right) \min\big(2^{M-4}, 2^M - 4\big) \cdot \frac{1}{4} \left\lfloor \frac{B}{2} \right\rfloor \left(\frac{L}{2^{M+8}} - 1\right) \tag{103}$$

$$\geq \frac{BL}{3 \cdot 2^{16}} (1 + o(1)). \tag{104}$$

The desired result follows from

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s}) | \mathcal{E}_l] \Pr(\mathcal{E}_l)$$

and $\Pr(\mathcal{E}_l) = 1 + o(1)$. ■

We then show in Theorems 25 and 27 that an oversized M leads to a large number of distinct chunks, each of which needs to be encoded in full and thus compression becomes ineffective. In particular, Theorem 25 covers the case when 2^M is of larger order than $\log B$ but still much smaller than the expected source symbol length L. Theorem 27 considers the case when $2^M = \Omega(L)$, and therefore a large number of chunks can be of lengths close to or even larger than the expected source symbol length.

Theorem 25. *Consider the source model with edit probability δ and the variable-length deduplication algorithm with marker length M. If $2^M = \omega(\log B) \cap o(L)$, then*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \frac{1}{2^{10} e^2} BL(1 + o(1)), \quad \textit{as } B \to \infty, \tag{105}$$

where the $o(1)$ term is independent of δ.

Proof: We show that if 2^M is in $\omega(\log B)$ and $o(L)$, the sum of the lengths of distinct chunks is a constant fraction of $|\boldsymbol{s}|$.

Each new chunk is encoded as a bit 1 followed by itself. Given $\mathcal{E}_l$, the expected number of bits needed for encoding distinct chunks is greater than or equal to

$$\mathbb{E}\left[\sum_{\boldsymbol{v} \in T_{VL}^1(\boldsymbol{s})} (|\boldsymbol{v}| + 1) | \mathcal{E}_l\right] = \sum_{\boldsymbol{v} \in \Sigma^*} \Pr\big(\boldsymbol{v} \in T_{VL}^1(\boldsymbol{s}) | \mathcal{E}_l\big)(|\boldsymbol{v}| + 1) \tag{106}$$

$$\geq \sum_{\boldsymbol{u} \in R_M} \Pr\Big(10^M \boldsymbol{u} 10^M \in Y_1^{B/2} | \mathcal{E}_l\Big)(|\boldsymbol{u}| + M + 2). \tag{107}$$

As a lower bound, we consider M-RLL strings with lengths in the range $\Big[2^M, \Big\lceil \big(2^M L\big)^{1/2} \Big\rceil\Big]$. Since asymptotically we have

$2^M \geq \frac{\log(B/A)-2}{H(\delta)}$, we apply Lemma 21 on (107) and get

$$\begin{aligned}
&\sum_{\ell=2^M}^{\left\lceil (2^M L)^{1/2} \right\rceil} \sum_{\boldsymbol{u}\in R_M^\ell} \Pr\Big(10^M \boldsymbol{u} 10^M \in Y_1^{B/2} | \mathcal{E}_l\Big)(\ell+M+1) && (108)\\
\geq &\sum_{\ell=2^M}^{\left\lceil (2^M L)^{1/2} \right\rceil} \left(\frac{BL}{2^7\cdot 2^{2M+2}}\left(1-\frac{1}{2^{M-1}}\right)^\ell - \frac{3B^2L^2}{2^{\ell+2M+2}}\right)\cdot(\ell+M+1) && (109)\\
\geq &\sum_{\ell=2^M}^{\left\lceil (2^M L)^{1/2} \right\rceil} \left(\frac{BL}{2^7\cdot 2^{2M+2}}\left(1-\frac{1}{2^{M-1}}\right)^\ell \ell\right) - \frac{3B^2L^4}{2^{2^M}} && (110)\\
\geq &\frac{BL}{2^7\cdot 2^{2M+2}} 2^{2(M-1)}\left(\frac{2^M-1}{2^{M-1}}+1\right)e^{-2}(1+o(1)) && (111)\\
&-\frac{3B^2L^4}{2^{2^M}} && (112)\\
\geq &\frac{BL}{2^{10}e^2}(1+o(1)) - \frac{3B^2L^4}{2^{2^M}} && (113)\\
= &\frac{BL}{2^{10}e^2}(1+o(1)), \quad \text{as } B\to\infty, && (114)
\end{aligned}$$

where the second inequality follows from $\left(2^M L\right)^{1/2} + M + 1 \leq L$ and the equality follows from $2^M = \omega(\log B)$. The second to last inequality follows from applying summation (513) in Appendix A6 with $a = 2^M, b = \left\lceil \left(2^M L\right)^{1/2} \right\rceil, \beta = 2^{M-1}$ and noting that $\frac{1}{2^{M-1}} \left\lceil \left(2^M L\right)^{1/2} \right\rceil = \omega(1)$.

Thus,

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})|\mathcal{E}_l] \geq \mathbb{E}\left[\sum_{\boldsymbol{v}\in T_{VL}^1(\boldsymbol{s})} (|\boldsymbol{v}|+1)|\mathcal{E}_l\right] \geq \frac{BL}{2^{10}e^2}(1+o(1)), \tag{115}$$

and the desired result follows from

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})|\mathcal{E}_l]\Pr(\mathcal{E}_l)$$

and $\Pr(\mathcal{E}_l) = 1 + o(1)$. ■

Next, we present a lemma that will be used in the proof of Theorem 27.

Lemma 26. *Consider the source string $\boldsymbol{s} = Y_1Y_2\cdots Y_B$, with each Y_b being a descendant of source symbol X_{J_b}. For any integer h and any pairs of integers $(b_1, b_2), (i_1, i_2)$, the probability of Y_{b_1} and Y_{b_2} having identical substrings of length h starting at positions i_1 and i_2, respectively, is*

$$\Pr\Big((Y_{b_1})_{i_1,h} = (Y_{b_2})_{i_2,h}\Big) = \frac{1}{2^h}, \tag{116}$$

if $J_{b_1} \neq J_{b_2}$ or $i_1 \neq i_2$.

The proof of Lemma 26 is presented in Appendix A5.

Theorem 27. *Consider the source model with edit probability δ and the variable-length deduplication algorithm with marker length M. If $2^M = \Omega(L)$, then*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \frac{1}{360}BL(1+o(1)), \quad \textit{as } B\to\infty, \tag{117}$$

where the $o(1)$ term is independent of δ.

Proof: Let $q = \min\big(2^{M-5}, L/2\big)$. We find a set of distinct M-RLL q-substrings of $\boldsymbol{s}$ that are encoded in full. In other words, any two such q-substrings are contained in two distinct chunks, or in two chunks that are duplicates, or in a single chunk without overlapping with each other. The total length of these q-substrings thus provides a lower bound on $\mathcal{L}_{VL}(\boldsymbol{s})$.

Let $L_1, \ldots, L_A$ be given and assume $\mathcal{E}_l$ holds. We consider the first $\lceil B/(4A) \rceil$ descendants of each source symbol. Let G_a denote the set of the first $\lceil B/(4A) \rceil$ descendants of X_a. Let Q_a be the set containing all non-overlapping q-substrings of G_a, i.e., $Q_a = \{\boldsymbol{x}_{1+(c-1)q,q} : \boldsymbol{x} \in G_a, 1 \le c \le c_a\}$, where $c_a = \lfloor L_a/q \rfloor$ and let $Q = \cup_{a=1}^{A} Q_a$. For $\boldsymbol{w} \in \Sigma^q$, let $\boldsymbol{w} \in Q$ denote the event that one of the substrings in Q equals $\boldsymbol{w}$. Applying Lemma 9 on Q (with substring length equal to descendant length) yields

$$\begin{aligned}\Pr(\boldsymbol{w} \in Q) &\ge \frac{1}{2}\min\left(1, \frac{1}{2}\left(\sum_{a=1}^{A} c_a\right)\frac{\mathcal{S}_\delta\left(q, \lceil \frac{B}{4A} \rceil\right)}{2^q}\right) &(118)\\ &= \frac{1}{4}\left\lceil \frac{B}{4A} \right\rceil \frac{\sum_{a=1}^{A} c_a}{2^q}, &(119)\end{aligned}$$

where the equality follows from $q = \Omega(L)$ and the property that $\mathcal{S}_\delta(\ell, m) = m$ if $m(1-\delta)^\ell \le 1$. So the expected number of distinct M-RLL strings in Q is at least

$$\begin{aligned}\sum_{\boldsymbol{w} \in R_M^q} \frac{1}{4}\left\lceil \frac{B}{4A} \right\rceil \frac{\sum_{a=1}^{A} c_a}{2^q} &\ge \frac{1}{4}\left(2 - \frac{1}{2^{M-2}}\right)^q \left\lceil \frac{B}{4A} \right\rceil \frac{\sum_{a=1}^{A} c_a}{2^q} &(120)\\ &\ge \frac{1}{5} \cdot \left\lceil \frac{B}{4A} \right\rceil \sum_{a=1}^{A} c_a, &(121)\end{aligned}$$

for all $M > 5$. Since the size of Q is $\lceil \frac{B}{4A} \rceil \sum_{a=1}^{A} c_a$, by the Markov bound, with probability at least $\frac{1}{9}$, the number of distinct M-RLL q-strings in Q is at least $\frac{1}{10}\lceil \frac{B}{4A} \rceil \sum_{a=1}^{A} c_a$.

Let $q' = \lceil q/2 \rceil$. Consider the q'-substrings of source blocks $Y_1, \ldots, Y_B$, i.e., $(Y_b)_{i,q'}$ for all $b \in [B], i \in [|Y_b|]$. Define $\mathcal{E}_d$ to be the following event: for every two source blocks Y_{b_1} and Y_{b_2}, the substring of Y_{b_1} starting at position i_1 is different from the substring of Y_{b_2} starting at position i_2, i.e., $(Y_{b_1})_{i_1,q'} \ne (Y_{b_2})_{i_2,q'}$, as long as $J_{b_1} \ne J_{b_2}$ or $i_1 \ne i_2$. Since there are at most $(2BL)^2$ pairs of such substrings, by the union bound and Lemma 26, $\mathcal{E}_d$ holds with probability at least

$$1 - (2BL)^2/2^{q'}. \tag{122}$$

When $\mathcal{E}_d$ holds, the distinct M-RLL q-substrings in Q are then non-overlapping substrings of the dictionary and it takes q-bits to encode each of them. To see this, we consider the first time such q-strings appear in the source string. Let $(Y_b)_{j,q}$ be one of the M-RLL strings in Q. Given $\mathcal{E}_d$, the only possible substrings of $\boldsymbol{s}$ that equal $(Y_b)_{k,q}$ are $(Y_1)_{k,q}, \ldots, (Y_B)_{k,q}$. Let b' be the smallest integer such that $(Y_{b'})_{j,q} = (Y_b)_{j,q}$. By the M-RLL property, $(Y_{b'})_{j,q}$ must be fully contained in a chunk. Moreover, this chunk must be a new chunk by the minimality of b' and is entered into the dictionary. Similarly, every distinct M-RLL q-substring corresponds to a q-substring in the dictionary. Since strings in Q do not overlap with each other, the corresponding q-substrings in the dictionary also do not overlap, and each takes q bits to store.

Combining the two arguments, with probability at least $\frac{1}{9} - \frac{(2BL)^2}{2^{q'}}$, there are $\frac{1}{10}\lceil \frac{B}{4A} \rceil \sum_{a=1}^{A} c_a$ distinct non-overlapping RLL substrings of length q, and each needs q bits to be encoded. It sums up to

$$q \cdot \frac{1}{10}\left\lceil \frac{B}{4A} \right\rceil \sum_{a=1}^{A} c_a \ge \frac{B}{40A}\sum_{a=1}^{A}(L_a - q) \tag{123}$$

bits. Therefore,

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})|\mathcal{E}_l] &\ge \left(\frac{1}{9} - \frac{(2BL)^2}{2^{q'}}\right)\frac{B}{40A}\sum_{a=1}^{A}(L - q) &(124)\\ &\ge \frac{BL}{360}(1 + o(1)). &(125)\end{aligned}$$

The desired result thus follows from (32). ■

As a summary of Theorems 24, 25, and 27, the following corollary shows that an inappropriate choice of M leads to poor performance.

Corollary 28. *Consider the source model with edit probability δ and variable-length deduplication with marker length M. If $2^M = o(\log B) \cup \omega(\log B)$, the compression ratio $\frac{\mathbb{E}[|\boldsymbol{s}|]}{\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]}$ is upper bounded by a universal constant for any edit probability*

$\delta > 0$.

In the next theorem, we give our upper bound on $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$. We consider the case when 2^M is of order $\Theta(\log B)$ and show that variable-length deduplication achieves high compression ratios.

Theorem 29. *Consider the source model with edit probability $\delta < \frac{1}{2}$. For any $\gamma \in (\delta, 1/2)$, the performance of variable-length deduplication with marker length M such that $2^M = \Theta(\log(B/A))$ satisfies*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \leq \left(12e^{-c_M}(c_M+1) + 4H(\gamma,\delta)\frac{(1+k_1)}{k_2}c_M\right)BL(1+o(1)), \tag{126}$$

as $B \to \infty$, where $c_M = \frac{\log(B/A)}{H(\gamma,\delta)2^{M+1}}$.

Proof: First, encoding the length $|\boldsymbol{s}|$ takes $2\log|\boldsymbol{s}| + 3 \leq 2\log(BL) + 5$ bits. We study next the encoding of chunks. We adopt the same strategy as [77]: dividing chunks into two categories, interior chunks and boundary chunks. Consider all chunks whose first symbols are in Y_b (see Figure 2). Some chunks depend on the values of the neighboring source blocks Y_{b-1} and Y_{b+1}, i.e., it is possible to alter the chunk by replacing Y_{b-1} or Y_{b+1} with other strings. We call these the 'boundary' chunks of Y_b. Other chunks are independent of the values of the neighboring source blocks. We call these the 'interior' chunks of Y_b. Denote the set of interior chunks in $\boldsymbol{s}$ by $\mathcal{C}^\circ(\boldsymbol{s})$. Note that we consider the first chunk and the last chunk of the whole data stream as boundary chunks. It is pointed out in [77] that the number of boundary chunks is upper bounded by $3(B+1)$ and the expected total length of boundary chunks is upper bounded by $B2^{M+2}$.[4] Therefore, encoding unique boundary chunks takes at most $3(B+1) + B2^{M+2}$ bits.

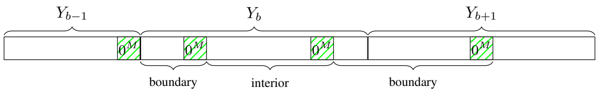


Figure 2: Occurrences of boundary chunks and interior chunks of Y_b in variable-length chunking.

We consider next encoding unique interior chunks. Clearly, every interior chunk follows a 0^M, i.e., the ending marker of the previous chunk. Moreover, this 0^M must also fully lie in the same source block as the chunk since otherwise this chunk is not an interior chunk. Therefore, the probability of occurrence of an interior chunk $\boldsymbol{u}10^M$ is at most the probability of the occurrence of $0^M\boldsymbol{u}10^M$ as a source block substring. It follows that

$$\begin{aligned}\mathbb{E}\left[\sum_{\boldsymbol{w}\in\mathcal{C}^\circ(\boldsymbol{s})}(|\boldsymbol{w}|+1)|\mathcal{E}_u\right] &\leq (M+1) + \sum_{\boldsymbol{u}\in R_M}\Pr\big(\boldsymbol{u}10^M \in \mathcal{C}^\circ(\boldsymbol{s})|\mathcal{E}_u\big)(|\boldsymbol{u}|+M+2) &(127)\\ &\leq (M+1) + \sum_{\boldsymbol{u}\in R_M}\Pr\big(0^M\boldsymbol{u}10^M \in Y_1^B|\mathcal{E}_u\big)(|\boldsymbol{u}|+M+2), &(128)\end{aligned}$$

where the term $M+1$ accounts for the chunk 0^M. We compute the summation in (128). Fix $\gamma \in (\delta, 1/2)$ and let $\ell_\gamma = \frac{\log(B/A)}{H(\gamma,\delta)}$.

- For all $\boldsymbol{u}$ such that $\left|0^M\boldsymbol{u}10^M\right| \leq \log B$, we trivially bound $\Pr\big(0^M\boldsymbol{u}10^M \in Y_1^B|\mathcal{E}_u\big)$ from above by 1. It follows that

$$\begin{aligned}\sum_{\ell=0}^{\lfloor\log B\rfloor-2M-1}\sum_{\boldsymbol{u}\in R_M^\ell}\Pr\big(0^M\boldsymbol{u}10^M \in Y_1^B|\mathcal{E}_u\big)(\ell+M+2) &\leq \sum_{\ell=0}^{\lfloor\log B\rfloor-2M-1}\sum_{\boldsymbol{u}\in R_M^\ell}(\ell+M+2) &(129)\\ &\leq \sum_{\ell=0}^{\lfloor\log B\rfloor-2M-1}2^\ell(\ell+M+2) &(130)\\ &\leq (\lfloor\log B\rfloor - M+1)2^{\log B-2M} &(131)\\ &\leq \frac{B\log B}{2^{2M}}. &(132)\end{aligned}$$

[4] Although in [77], the upper bounds are derived for source strings produced by an edit-free source, the same upper bounds hold when edits exist since every source block is still a Bernoulli(1/2) process by itself.

- For $\boldsymbol{u}$ such that $\left|0^M\boldsymbol{u}10^M\right| \geq \ell_\gamma$, we apply Lemma 20 and find
$$\Pr\big(0^M\boldsymbol{u}10^M \in Y_1^B|\mathcal{E}_u\big) \leq 2AL\frac{\mathcal{S}_\delta\big(\left|0^M\boldsymbol{u}10^M\right|, \frac{3B}{2A}\big)}{2^{\left|0^M\boldsymbol{u}10^M\right|}} \leq \frac{3BL}{2^{\left|0^M\boldsymbol{u}10^M\right|}}. \tag{133}$$
It follows that
$$\sum_{\ell=\lceil\ell_\gamma\rceil-2M-1}^{2L}(\ell+M+2)\sum_{\boldsymbol{u}\in R_M^\ell}\Pr\big(0^M\boldsymbol{u}10^M \in Y_1^B|\mathcal{E}_u\big) \tag{134}$$
$$\leq \sum_{\ell=\lceil\ell_\gamma\rceil-2M-1}^{2L}\sum_{\boldsymbol{u}\in R_M^\ell}\frac{3BL}{2^{\ell+2M+1}}(\ell+M+2) \tag{135}$$
$$\leq \sum_{\ell=\lceil\ell_\gamma\rceil-2M-1}^{2L} 2\left(2-\frac{1}{2^M}\right)^\ell\frac{3BL}{2^{\ell+2M+1}}(\ell+M+2) \tag{136}$$
$$= \frac{3BL}{2^{2M}}\left(\sum_{\ell=\lceil\ell_\gamma\rceil-2M-1}^{2L}\left(1-\frac{1}{2^{M+1}}\right)^\ell(M+2)\right. \tag{137}$$
$$\left.+\sum_{\ell=\lceil\ell_\gamma\rceil-2M-1}^{2L}\left(1-\frac{1}{2^{M+1}}\right)^\ell\ell\right) \tag{138}$$
$$= (1+o(1))\frac{3BL}{2^{2M}}\left(2^{M+1}\cdot e^{-\frac{\lceil\ell_\gamma\rceil-2M-1}{2^{M+1}}}\right. \tag{139}$$
$$\left.+\,2^{2(M+1)}\cdot e^{-\frac{\lceil\ell_\gamma\rceil-2M-1}{2^{M+1}}}\left(\frac{\lceil\ell_\gamma\rceil-2M-1}{2^{M+1}}+1\right)\right) \tag{140}$$
$$= 12BL\cdot e^{-\frac{\ell_\gamma}{2^{M+1}}}\left(\frac{\ell_\gamma}{2^{M+1}}+1\right)(1+o(1)), \tag{141}$$
where the second equality follows by applying summations (505) and (513) in Appendix A6 with $a = \lceil\ell_\gamma\rceil - 2M - 1$, $b = 2L$, $\beta = 2^{M+1}$ and noting that $\frac{2L}{2^{M+1}} = \omega(1)$.
- If $\log B \leq \ell_\gamma$, then there are additional terms corresponding to string $\boldsymbol{u}$ such that $\log B \leq \left|0^M\boldsymbol{u}10^M\right| \leq \ell_\gamma$. Again by Lemma 20,
$$\Pr\big(0^M\boldsymbol{u}10^M \in Y_1^B|\mathcal{E}_u\big) \leq 2AL\frac{\mathcal{S}_\delta\big(\left|0^M\boldsymbol{u}10^M\right|, \frac{3B}{2A}\big)}{2^{\left|0^M\boldsymbol{u}10^M\right|}} \tag{142}$$
$$\leq 5BL2^{-\left|0^M\boldsymbol{u}10^M\right|(1+D(\gamma||\delta))}, \tag{143}$$
where the second inequality follows from (41) and the fact that $2^{nH(\gamma)} \leq \frac{B}{A}2^{-nD(\gamma||\delta)}$ if $n \leq \frac{\log(B/A)}{H(\gamma,\delta)}$.

Thus,

$$\sum_{\ell=\lceil\log B\rceil-2M-1}^{\lfloor\ell_\gamma\rfloor-2M-1}\sum_{\boldsymbol{u}\in R_M^\ell}\Pr\left(0^M\boldsymbol{u}10^M\in Y_1^B|\mathcal{E}_u\right) \tag{144}$$

$$\cdot(\ell+M+2) \tag{145}$$

$$\leq\sum_{\ell=\lceil\log B\rceil-2M-1}^{\lfloor\ell_\gamma\rfloor-2M-1}\sum_{\boldsymbol{u}\in R_M^\ell}5BL2^{-(\ell+2M+1)(1+D(\gamma||\delta))} \tag{146}$$

$$\cdot(\ell+M+2) \tag{147}$$

$$\leq\frac{5BL}{2^{2M}}\sum_{\ell=\lceil\log B\rceil-2M-1}^{\lfloor\ell_\gamma\rfloor-2M-1}\left(1-\frac{1}{2^{M+1}}\right)^{\ell}2^{-(\ell+2M+1)D(\gamma||\delta)} \tag{148}$$

$$\cdot(\ell+M+2) \tag{149}$$

$$\leq\frac{5BL\ell_\gamma^2}{2^{2M}}\left(1-\frac{1}{2^{M+1}}\right)^{\log B-2M-1}2^{-D(\gamma||\delta)\log B} \tag{150}$$

$$=\Theta\left(B^{1-D(\gamma||\delta)}L\right) \tag{151}$$

$$=o(BL), \tag{152}$$

where the first equality follows from the fact that $\frac{\ell_\gamma^2}{2^{2M}}$ and $\left(1-\frac{1}{2^{M+1}}\right)^{\log B-2M-1}$ are both $\Theta(1)$ since 2^M and ℓ_γ are $\Theta(\log(B/A))$.

Plugging (132), (141) and (152) in (128), we find that as $B\to\infty$ (also $A,L\to\infty$),

$$\mathbb{E}\left[\sum_{\boldsymbol{w}\in\mathcal{C}^\circ(\boldsymbol{s})}(|\boldsymbol{w}|+1)|\mathcal{E}_u\right]\leq 12e^{-c_M}(c_M+1)BL+o(BL), \tag{153}$$

where $c_M=\frac{\ell_\gamma}{2^{M+1}}$.

If the complement of $\mathcal{E}_u$ holds, then the number of bits needed for encoding interior chunks at their first occurrences is at most $4BL$, since the total length of interior chunks is at most $2BL$ and the total number of chunks is at most $2BL$. By noting that $\Pr\left(\bar{\mathcal{E}}_u\right)\leq Ae^{-\frac{B}{10A}}$,

$$\mathbb{E}\left[\sum_{\boldsymbol{w}\in\mathcal{C}^\circ(\boldsymbol{s})}(|\boldsymbol{w}|+1)\right]\leq 12e^{-c_M}(c_M+1)BL+o(BL)+4BLAe^{-\frac{B}{10A}} \tag{154}$$

$$=12e^{-c_M}(c_M+1)BL(1+o(1)). \tag{155}$$

The number of bits needed for encoding pointers of repeated chunks can be bounded from above in a trivial way. Note that there are at most $\frac{|\boldsymbol{s}|}{M}+1$ strings in the dictionary T. So a pointer takes at most $\log\left(\frac{|\boldsymbol{s}|}{M}+1\right)+1\leq\log|\boldsymbol{s}|$ bits. Moreover, the total number of chunks is less than the number of occurrences of 0^M plus 1 since every chunk except possibly the last one ends with 0^M. On average, the number of occurrences of 0^M in Y_b is at most $\frac{|Y_b|}{2^M}$. So given $|\boldsymbol{s}|$, the expected number of chunks in $\boldsymbol{s}$ is at most $\frac{|\boldsymbol{s}|}{2^M}+B+1$. Therefore the expected number of bits used by pointers is at most

$$\mathbb{E}\left[(\log|\boldsymbol{s}|+1)\cdot\left(\frac{|\boldsymbol{s}|}{2^M}+B+1\right)\right]\leq\log(2BL+1)\left(\frac{2BL}{2^M}+B+1\right) \tag{156}$$

$$\leq 2BL\frac{\log(BL)}{2^M}(1+o(1)) \tag{157}$$

$$\leq 4H(\gamma,\delta)\frac{(1+k_1)}{k_2}c_M\cdot BL(1+o(1)), \tag{158}$$

where the last inequality follows from $\frac{\log(BL)}{\log(B/A)}\leq\frac{1+k_1}{k_2}(1+o(1))$.

The desired result follows from summing (155) and (158) and noting that the number of bits used for encoding the length of $\boldsymbol{s}$ and the unique boundary chunks are $o(BL)$. ■

We perform the following analysis for minimizing the upper bound given by Theorem 29. For any given $c > 0$, there exists an integer value for M such that $c \leq c_M \leq 2c$. For this M, (126) is upper bounded by

$$\left(12e^{-c}(c+1) + 8H(\gamma,\delta)\frac{(1+k_1)}{k_2}c\right)BL(1+o(1)), \tag{159}$$

since $e^{-c}(c+1)$ is decreasing in c when $c > 0$. We can always find γ such that $H(\gamma,\delta) \leq 2H(\delta)$. Such γ gives

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \leq \left(12e^{-c}(c+1) + 16H(\delta)\frac{(1+k_1)}{k_2}c\right)BL(1+o(1)). \tag{160}$$

Let $h = 4H(\delta)\frac{(1+k_1)}{3k_2}$. Upper bounding the above expression is equivalent to upper bounding the function $f(c) = e^{-c}(c+1)+hc$, $c \in (0,+\infty)$. If $h < e^{-1}$, then $f(c)$ has a local minimum at $c = -W_{-1}(-h)$, where W_{-1} is the lower branch of the Lambert W function. If $h \geq e^{-1}$, then $f(c)$ is monotonically increasing in $(0,+\infty)$. Therefore, $c = -W_{-1}(-\min(e^{-1},h))$ provides an upper bound on $f(c)$. As an example, for $A = L = B^{1/2}$ (i.e., $k_1 = k_2 = \frac{1}{2}$), Figure 3 shows the upper bound given by (160) with $c = -W_{-1}(-\min(e^{-1},h))$, as well as $H(\delta)$, as δ ranges from 10^{-5} to 10^{-1}.

Note that $h \leq e^{-1}$ holds for small enough δ. When this holds, the upper bound (160) can be rewritten as

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] &\leq \left(12e^{-c}(c+1) + 16H(\delta)\frac{(1+k_1)}{k_2}c\right)BL(1+o(1)) && (161)\\ &\leq 12e^{-c}(c^2+c+1)BL(1+o(1)), && (162)\end{aligned}$$

where $c = -W_{-1}(-4H(\delta)(1+k_1)/(3k_2))$. Hence the upper bound on the normalized expected compressed length approaches 0 as δ approaches 0. This means that as the entropy becomes smaller, the compression ratio grows if the length of the marker is chosen appropriately. In particular, it can be seen that the proper length of the marker depends on δ, which represents the degree of variability between the copies.

Large compression ratios when entropy is small is desirable and variable-length deduplication achieves this. However, it can be shown and also observed in Figure 3 that the upper bound of the ratio $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]/H(\mathcal{I}_b(\delta))$ given by Theorem 29 increases as δ decreases. Therefore, despite the large compression ratios, the gap to entropy may become large for small δ. Determining whether this is indeed the case or the bound provided here is loose is left to future work.

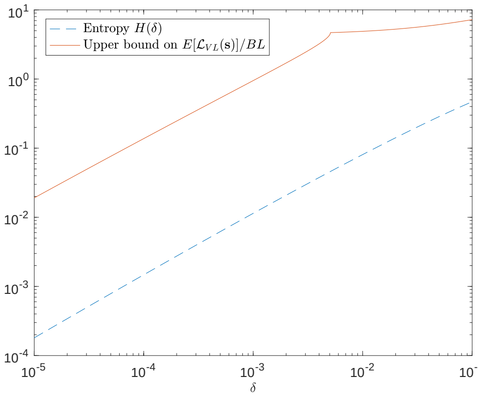


Figure 3: Upper bound on $\frac{\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]}{BL}$ and $H(\delta)$ vs the edit probability δ with $A = L = B^{1/2}$, as δ ranges from 10^{-5} to 10^{-1}.

E. Deduplication over source $\mathcal{I}_f(t)$

While [62] extended the information-theoretic analysis of data deduplication to approximate repeats, the studied model has entropy linear in the length of the uncompressed string (shown in Lemma 3) and the gain in compression is at best a constant factor. This makes compression less challenging and the distinction between the performance of compression methods less clear. In [64], we assume each source block only contains a constant number of substitutions (randomly distributed) instead of iid bit flips, leading to the entropy being of smaller order than the length of the uncompressed string and thus high compression ratio can be achieved.

The asymptotic regime that we are particularly interested in is when the source string uncertainty mainly results from substitution edits, i.e., the entropy $H(\mathcal{I}_f(t))$ is dominated by the term $B\log\binom{L}{t}$. Therefore, we assume that asymptotically $\log A = O(\log L)$ and $AL = O(B\log L)$.

In the following, we study the performance of variable-length and multi-chunk deduplication algorithms over source model $\mathcal{I}_f(t)$.

1) Variable-length deduplication

We start with a lower bound on the expected length of the compressed strings by variable-length deduplication.

Theorem 30. *If $B \le A\binom{L/2}{t}$, then the average length of the compressed strings by variable-length deduplication with optimal marker length M satisfies*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \ge \Omega\left(\frac{BL^{\frac{1}{t+1}}}{\log L}\right). \tag{163}$$

Proof: In this proof, we lower bound $\mathcal{L}_{VL}(\boldsymbol{s})$ by the total length of the distinct chunks, denoted W, plus the number of chunks C since each chunk needs one bit indicating if it has appeared before. We have C is greater than the number of non-overlapping marker strings in $\boldsymbol{s}$. So $\mathbb{E}[C] \ge \frac{BL}{M2^M}$. It follows that

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \ge \mathbb{E}[W] + \frac{BL}{M2^M}. \tag{164}$$

We lower bound $\mathbb{E}[W]$ in the following.

For each source symbol X_a, we use n_a to denote the number of its descendants among the source blocks. Moreover, we fix the positions where substitutions occur in each of its descendants. Let $\mathcal{M}$ contain the information about $\{n_a\}_{a=1}^A$, the positions of substitutions, and the lengths of source symbols $\{L_a\}_{a=1}^A$.

We first compute the expected value of $\mathcal{L}_{VL}(\boldsymbol{s})$ conditioned on $\mathcal{M}$. Let $\ell = \min\big(2^{M-5}, L/4\big)$. Partition each X_a into segments of length ℓ, i.e., for each X_a, we write

$$\mathsf{X}_a = \boldsymbol{x}_{a,1}\boldsymbol{x}_{a,2}\cdots\boldsymbol{x}_{a,c_a}\boldsymbol{x}_{a,c_a+1}, \tag{165}$$

where $|\boldsymbol{x}_{a,1}| = \cdots = |\boldsymbol{x}_{a,c_a}| = \ell, c_{a+1} = \lceil L_a/\ell\rceil$. We consider the substrings of the descendants of X_a that correspond to $\boldsymbol{x}_{a,j}$, denoted $\boldsymbol{h}_{a,j}^1, \dots, \boldsymbol{h}_{a,j}^{n_a}$ (see Figure 4). Each of $\boldsymbol{h}_{a,j}^1, \dots, \boldsymbol{h}_{a,j}^{n_a}$ results from $\boldsymbol{x}_{a,j}$ through at most t substitutions. For each $1 \le j \le c_a$, we assume without loss of generality that $\boldsymbol{h}_{a,j}^1, \dots, \boldsymbol{h}_{a,j}^{m_{a,j}}$ are distinct, where $m_{a,j}$ denotes the total number of distinct strings among $\boldsymbol{h}_{a,j}^1, \dots, \boldsymbol{h}_{a,j}^{n_a}$.

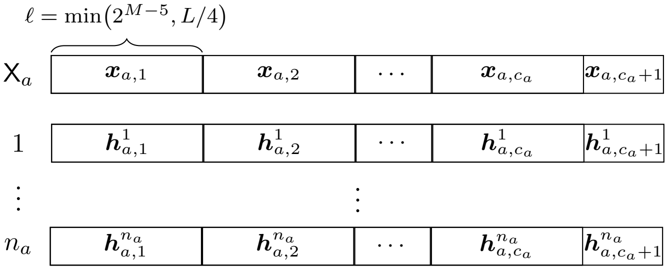

$\ell = \min\big(2^{M-5}, L/4\big)$
X_a
$\boldsymbol{x}_{a,1}$
$\boldsymbol{x}_{a,2}$
$\cdots$
$\boldsymbol{x}_{a,c_a}$
$\boldsymbol{x}_{a,c_a+1}$
1
$\boldsymbol{h}_{a,1}^1$
$\boldsymbol{h}_{a,2}^1$
$\cdots$
$\boldsymbol{h}_{a,c_a}^1$
$\boldsymbol{h}_{a,c_a+1}^1$
$\vdots$
$\vdots$
n_a
$\boldsymbol{h}_{a,1}^{n_a}$
$\boldsymbol{h}_{a,2}^{n_a}$
$\cdots$
$\boldsymbol{h}_{a,c_a}^{n_a}$
$\boldsymbol{h}_{a,c_a+1}^{n_a}$

Figure 4: A partition of X_a and its n_a descendants into segments of length ℓ.

We consider the event $\mathcal{E}_1$ that any two $\ell/2$-substrings of the source alphabet are of Hamming distance at least $2t+1$ from each other. For any two $\ell/2$ substrings of the source alphabet, regardless of whether they overlap or not, the probability of their Hamming distance being less than or equal to $2t$ is at most

$$\frac{\sum_{m=0}^{2t}\binom{\ell/2}{m}}{2^{\ell/2}} \le \binom{\ell/2}{2t}\frac{\ell/2-2t+1}{\ell/2-4t+1}\cdot\frac{1}{2^{\ell/2}} \le \frac{(\ell/2)^{2t}}{2^{\ell/2}}, \tag{166}$$

when $\ell \ge 12t$, where the first inequality follows from $\frac{\binom{N}{k-i}}{\binom{N}{k}} \le \left(\frac{k}{N-k+1}\right)^i$ for $i \le k$. Since there are less than $(2AL)^2$ pairs of $\ell/2$-substrings in the source alphabet, by the union bound, It can be shown by the union bound that

$$\Pr(\mathcal{E}_1) \geq 1 - (2AL)^2 \frac{(\ell/2)^{2t}}{2^{\ell/2}} \geq \frac{3}{4}, \tag{167}$$

when $t/3 \leq \frac{\ell/2}{12\log(\ell/2)}$ and $\log(AL) \leq \ell/8 - 2$.

Assume $\mathcal{E}_1$ holds. Then different source alphabet $\ell/2$-substrings have different descendants. Consider $\boldsymbol{h}_{a,j}^1$. The only substrings that are possible to be the same as $\boldsymbol{h}_{a,j}^1$ are $\boldsymbol{h}_{a,j}^2, \ldots, \boldsymbol{h}_{a,j}^{n_a}$. Note that if we have defined $\mathcal{E}_1$ to be the event that any two ℓ-substrings of the source alphabet are of Hamming distance at least $2t+1$ from each other, then when $\mathcal{E}_1$ holds, $\boldsymbol{h}_{a,j}^1$ is still possible to be the same as ℓ-substrings that sit across boundaries of source blocks. Therefore, we can assume without loss of generality that $\boldsymbol{h}_{a,j}^1, \ldots, \boldsymbol{h}_{a,j}^{m_{a,j}}$ are the first time such strings appear. For any $\boldsymbol{h}_{a,j}^n$, $1 \leq n \leq m_{a,j}$, if $\boldsymbol{h}_{a,j}^n$ is M-RLL, then it is fully contained in some chunk, denoted Z. So Z must have not appeared before and takes $|Z|$ bits to encode since its substring $\boldsymbol{h}_{a,j}^n$ has not appeared before. Now that consider the set of distinct descendants of all ℓ-segments in the source alphabet, i.e.,

$$\mathcal{H} = \{\boldsymbol{h}_{a,j}^n : 1 \leq a \leq A, 1 \leq j \leq c_a, 1 \leq n \leq m_{a,j}\}. \tag{168}$$

Every M-RLL string in $\mathcal{H}$ is contained in a chunk that has not appeared before. To enter these chunks into the dictionary, it takes ℓ bits for each M-RLL string in $\mathcal{H}$ since strings in $\mathcal{H}$ do not overlap.

Since the source symbol X_a is a Ber(1/2) process, each $\boldsymbol{h}_{a,j}^n$ is M-RLL with probability at least $1 - 2^{M-5} \cdot 2^{-M} = 1 - 2^{-5}$. Since it holds for every element in $\mathcal{H}$, by Markov's inequality that with probability at least $3/4$, over $7/8$ of the strings in $\mathcal{H}$ are M-RLL.

Combining the two arguments, with probability at least $1/2$, there are $7|\mathcal{H}|/8$ distinct M-RLL substrings in $\mathcal{H}$, which contribute

$$\frac{7}{8}|\mathcal{H}|\ell = \frac{7}{8}\ell \sum_{a=1}^{A} \sum_{j=1}^{c_a} m_{a,j} \tag{169}$$

bits to the total length of distinct chunks W. It follows that when $\ell \geq 8(2 + \log(AL))$,

$$\mathbb{E}[W|\mathcal{M}] \geq \frac{7}{16}\ell \sum_{a=1}^{A} \sum_{j=1}^{c_a} m_{a,j}, \tag{170}$$

and we further have

$$\mathbb{E}[W] = \mathbb{E}[\mathbb{E}[W|\mathcal{M}]] \geq \mathbb{E}\left[\frac{7}{16}\ell \sum_{a=1}^{A} \sum_{j=1}^{c_a} m_{a,j}\right] \tag{171}$$

$$= \frac{7}{16}\ell \sum_{a=1}^{A} \sum_{j=1}^{c_a} \mathbb{E}[m_{a,j}]. \tag{172}$$

Next, we compute the expected value of $m_{a,j}$. Note that $m_{a,j}$ is independent of the source alphabet. The probability of k substitutions occurring at a fixed set of positions in $\boldsymbol{x}_{a,j}$ is $\binom{L_a-\ell}{t-k}/\binom{L_a}{t}$. Hence,

$$\mathbb{E}[m_{a,j}] \geq \sum_{k=0}^{t} \binom{\ell}{k} \cdot \left(1 - \left(1 - \frac{\binom{L_a-\ell}{t-k}}{A\binom{L_a}{t}}\right)^B\right) \geq \frac{1}{2} \sum_{k=0}^{t} \binom{\ell}{k} \cdot \min\left(1, \frac{B\binom{L_a-\ell}{t-k}}{A\binom{L_a}{t}}\right) \tag{173}$$

$$\geq \frac{1}{2}\left(\binom{\ell}{0} \cdot \min\left(1, \frac{B\binom{L_a-\ell}{t-0}}{A\binom{L_a}{t}}\right) + \binom{\ell}{t} \cdot \min\left(1, \frac{B\binom{L_a-\ell}{t-t}}{A\binom{L_a}{t}}\right)\right) \tag{174}$$

$$= \frac{1}{2}\left(1 + \frac{B\binom{\ell}{t}}{A\binom{L_a}{t}}\right) \geq \frac{1}{2}\left(1 + \frac{B\binom{\ell}{t}}{A\binom{2L}{t}}\right), \tag{175}$$

where the second equality follows from

$$\frac{B\binom{L_a-\ell}{t}}{A\binom{L_a}{t}} \geq \frac{B}{A}\frac{\binom{L/4}{t}}{\binom{2L}{t}} = \frac{B}{A}\left(\frac{1}{8}\right)^t(1+o(1)) \geq 1. \tag{176}$$

Thus, by (175), (172) and (164), $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ is lower bounded by

$$\frac{7}{16}\ell\left(1+\frac{B\binom{\ell}{t}}{A\binom{2L}{t}}\right)\left(\sum_{a=1}^{A}\left\lfloor\frac{L/2}{\ell}\right\rfloor\right)I_{\ell\geq 8(2+\log(AL))}+\frac{BL}{M2^M} \tag{177}$$

$$\geq\frac{7}{64}\left(AL+BL\left(\frac{\ell}{2L}\right)^t(1+o(1))\right)I_{\ell\geq 8(2+\log(AL))}+\frac{BL}{M2^M}, \tag{178}$$

where the last inequality follows from $\left\lfloor\frac{L/2}{\ell}\right\rfloor \geq \frac{L}{4\ell}$, $\frac{\binom{\ell}{t}}{\binom{2L}{t}} = \left(\frac{\ell}{2L}\right)^t(1+o(1))$.

When $2^M = O(L^{\frac{t}{t+1}})$,

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq \frac{BL}{M2^M} = \Omega\left(\frac{BL^{\frac{1}{t+1}}}{\log L}\right). \tag{179}$$

When $2^M = \omega\left(L^{\frac{t}{t+1}}\right)$,

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \geq\frac{7}{64}\left(AL+BL\left(\frac{\ell}{2L}\right)^t(1+o(1))\right) \tag{180}$$

$$=\Theta(AL)+\omega\left(BL^{\frac{1}{t+1}}\right). \tag{181}$$

■

By the preceding theorem, if $\Omega\left(\frac{AL}{\log L}\right) \leq B \leq A\binom{L/2}{t}$, then $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ is greater than $H(\mathcal{I}_f(t))$ by at least an order of $\frac{L^{\frac{1}{t+1}}}{\log^2 L}$.

In the following, we derive an upper bound on the performance of variable-length algorithm.

Theorem 31. *The average length of the compressed strings by variable-length deduplication with optimal marker length M satisfies*

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \leq 2AL+\Theta\left(BL^{\frac{1}{2}}\log^{\frac{1}{2}}(BL)\right). \tag{182}$$

Proof: The variable-length deduplication partitions the source string $\boldsymbol{s}$ as a random number C of chunks, denoted $Z_1,\dots,Z_C$. The length of $\boldsymbol{s}$ can be encoded in at most $2\log|\boldsymbol{s}|+1$ bits by Elias gamma coding. Let T_{VL}^c denote the dictionary right after chunk Z_c is processed (T_{VL}^0 denotes the initial empty dictionary). We can write

$$\mathcal{L}_{VL}(\boldsymbol{s}) \leq\sum_{c=1}^{C}\left(I_{Z_c\in T_{VL}^{c-1}}(1+\log\left|T_{VL}^{c-1}\right|+1)+I_{Z_c\notin T_{VL}^{c-1}}(1+|Z_c|)\right)+2\log|\boldsymbol{s}|+1. \tag{183}$$

We next consider a partition of $\boldsymbol{s}$ into a random number of "edit blocks". We first split $\boldsymbol{s}$ at all the boundaries of source blocks. Each source block Y_b is further split in the following way. For all $1\leq a\leq A$, we let the first descendant of X_a be $Y_{g(a)}$, i.e., $g(a)$ is the smallest index such that $J_{g(a)}=a$ (we define $g(a)$ only for source symbols that have at least one descendant). For any other descendant Y_b of X_a, we consider the mismatches between Y_b and $Y_{g(a)}$. Suppose Y_b differs from $Y_{g(a)}$ in positions $c_1,c_2,\dots,c_m$, $0\leq c_m\leq 2t$. We break Y_b into c_m+1 segments at these points. Specifically, for each $1\leq j\leq m$, we split between the (c_j-1)-th symbol and the c_j-th symbol. The first segment is set to be empty if $c_1=1$. These segments are referred to as edit blocks. As an example, if $c_1=2, c_2=5$ and

$$Y_b=01000101, \tag{184}$$

then the edit blocks are $0, 100, 0101$.

Thus, conditioned on the differences between each Y_b and its corresponding "first descendant" source block $Y_{h_{J_b}}$, we can partition the source string $\boldsymbol{s}=Y_1Y_2\cdots Y_B$ into a random number K of edit blocks, denoted $D_1,\dots,D_K$ (the boundaries of source blocks are also breakpoints). Note that each $Y_{g(a)}$ has no mismatch with itself, so they are edit blocks by themselves, i.e., there exist $k_1, \ldots, k_A$ such that $D_{k_1} = Y_{h_1}, \ldots, D_{k_A} = Y_{g(a)}$.

We define a similar notion of interior chunks and boundary chunks as in [77, Theorem 3] but with respect to edit blocks and the substitutions. Consider chunks whose first symbols are in edit block D_k. Some of them are invariant of the neighboring source blocks and the first bit of D_k. In other words, by replacing D_{k-1}, D_{k+1} or the first bit of D_k by any other strings, the existence or content of these chunks do not change. They are referred to as "interior" chunks. We denote the set of indexes of interior chunks in D_k by $\mathcal{C}_k^\circ$. The chunks that are not interior chunks are referred to as "boundary" chunks. Their content depend on neighboring edit blocks D_{k-1}, D_{k+1} and the first bit of D_k, which corresponds to a substitution. We denote the set of indexes of all boundary chunks that start in D_k by $\partial\mathcal{C}_k$. We give examples in the following of boundary chunks (indicated by underbrackets) and interior chunks (indicated by overbrackets) when marker length $M = 3$ in various cases. Vertical bars indicate the boundaries of edit blocks. Different rows are independent examples.

$$\cdots 000\,\underbracket{10110|00}\,\underbracket{0101000}\,010\cdots \tag{185}$$

$$\cdots\,\underbracket{101100|0}\,\underbracket{000}\,\underbracket{00101000}\,010\cdots \tag{186}$$

$$\cdots 000\,\underbracket{10|11000}\,\overbracket{0101000}\,010\cdots \tag{187}$$

$$\cdots 000\,\overbracket{1011000}\,|\,\underbracket{0101000}\,010\cdots \tag{188}$$

$$\cdots 000\,\underbracket{101100|0}\,\underbracket{1000}\,\overbracket{11000}\,10\cdots \tag{189}$$

$$\cdots 000\,\underbracket{1011000}\,|\,\underbracket{1000}\,\underbracket{11000}\,10\cdots \tag{190}$$

By (183),

$$\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})] \le \mathbb{E}\left[\sum_{k=1}^{K}\left(\sum_{\substack{c\in\mathcal{C}_k^\circ\\ Z_c\notin T_{VL}^{c-1}}}|Z_c| + \sum_{\substack{c\in\partial\mathcal{C}_k\\ Z_c\notin T_{VL}^{c-1}}}|Z_c|\right) + C + \sum_{Z_c\in T_{VL}^{c-1}}\left(1+\log\left|T_{VL}^{c-1}\right|\right)\right]. \tag{191}$$

We first consider the interior chunks that appear for the first time. Consider the edit block D_k and the source block Y_b that contains D_k. If Y_b is not the first source block among the descendants of X_{J_b}, then D_k equals to the substring of $Y_{h_{J_a}}$ at the same location with the first bit flipped. It follows from the definition of interior chunks that any interior chunk of D_k must have already appeared in that substring of $Y_{h_{J_a}}$. Thus, any interior chunk of $\boldsymbol{s}$ that has not appeared in the dictionary is a substring of one of $Y_{h_1}, \ldots, Y_{g(a)}$. The total length of these chunks is hence less than the sum of lengths of $Y_{h_1}, \ldots, Y_{g(a)}$. Hence,

$$\mathbb{E}\left[\sum_{k=1}^{K}\sum_{\substack{c\in\mathcal{C}_k^\circ\\ Z_c\notin T_{VL}^{c-1}}}|Z_c|\right] \le 2AL. \tag{192}$$

Secondly, we upper bound the lengths of boundary chunks. We adopt a similar approach as [77]. We call an occurrence of 10^M *internal* to an edit block D if it starts in D but after its first (substituted) bit. For edit block D, we use $\mathrm{head}(D)$ to denote the prefix of D which ends at the last zero of the first internal 10^M in D. We use $\mathrm{tail}(D)$ to denote the suffix of D which starts at the first zero of the rightmost 0^M in D. Head or tail is defined to be D itself if D does not contain corresponding patterns. Consider a boundary chunk Z with starting position in edit block D_k and ending position in edit block D_{k+j}, $j \ge 0$. If $j = 0$, then at least one of the following two event must hold: i) the starting position of Z is in $\mathrm{tail}(D_k)$, ii) the ending position of Z is in $\mathrm{head}(D_k)$. This can be seen by noting that any chunk after the first internal 10^M is invariant of D_{k-1} and the first bit of D_k, and the rightmost 0^M contains the end of an interior chunk. For a similar reason, when $j > 0$, the starting position of Z must be in $\mathrm{tail}(D_k)$ and the ending position of Z must be in $\mathrm{head}(D_{k+j})$. Thus, the total length of boundary chunks that start in or after D_k and end in or before D_{k+j} is upper bounded by the total length of

$\mathrm{head}(D_k), \dots, \mathrm{head}(D_{k+j}), \mathrm{tail}(D_k), \dots, \mathrm{tail}(D_{k+j})$. It follows that

$$\sum_{k=1}^{K} \sum_{c \in \partial \mathcal{C}_k} |Z_c| \leq \sum_{k=1}^{K} (|\mathrm{head}(D_k)| + |\mathrm{tail}(D_k)|). \tag{193}$$

Note that every D_k by itself is a Bernoulli(1/2) process. The expected number of bits forwards until the end of the first internal 10^M is $2^{M+1}+1$ and the expected number of bits backwards until the beginning of the rightmost 0^M is $2^{M+1}-2$ [94, Chapter 8]. It follows that

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{K} \sum_{\substack{c \in \partial \mathcal{C}_k \\ Z_c \notin T_{VL}^{c-1}}} |Z_c|\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K} \sum_{c \in \partial \mathcal{C}_k} |Z_c|\right] && (194)\\
&\leq \mathbb{E}\left[\sum_{k=1}^{K} \big(2^{M+1} + 1 + 2^{M+1} - 2\big)\right] && (195)\\
&\leq (2t+1)B\big(2^{M+2} - 1\big). && (196)
\end{aligned}$$

Finally, we upper bound the remaining terms in (191). The number of chunks C is less than the number of occurrences of marker 0^M plus 1 (the last chunk). The expected number of occurrences of 0^M inside source blocks is $\frac{BL}{2^M}$. It follows that $\mathbb{E}[C] \leq \frac{BL}{2^M} + B$, where B accounts for possible occurrences of 0^M across the boundaries of source blocks. Therefore,

$$\begin{aligned}
\mathbb{E}\left[C + \sum_{Z_c \in T_{VL}^{c-1}} \big(1 + \log\big|T_{VL}^{c-1}\big|\big)\right] &\leq \mathbb{E}[C + C(1 + \log C)] \leq \mathbb{E}[C](3 + \log(BL)) && (197)\\
&\leq B\left(1 + \frac{L}{2^M}\right)(3 + \log(BL)), && (198)
\end{aligned}$$

where the second inequality follows from $C \leq 2BL$.

Thus, by plugging (192), (196) and (198) in (191), we find that $\mathbb{E}[\mathcal{L}_{VL}(\boldsymbol{s})]$ is upper bounded by

$$\begin{aligned}
&2AL + B\left((2t+1)2^{M+2} + 3 + \log(BL) - (2t+1) + \frac{(3 + \log(BL))L}{2^M}\right) && (199)\\
&= 2AL + B\left((2t+1)2^{M+2} + \frac{(3 + \log(BL))L}{2^M}\right) + \Theta(B \log(BL)). && (200)
\end{aligned}$$

The preceding upper bound is minimized at $2^M = \left(\frac{(3+\log(BL)L)}{4(2t+1)}\right)^{\frac{1}{2}}$ with minimum value

$$2AL + 4B((2t+1)(3 + \log(BL))L)^{\frac{1}{2}} = 2AL + \Theta\left(BL^{\frac{1}{2}} \log^{\frac{1}{2}}(BL)\right). \tag{201}$$

■

Note that the expected length of the source string is BL. When $\log B = o(L)$, $BL^{\frac{1}{2}} \log^{\frac{1}{2}}(BL) = o(BL)$. Therefore, under this condition, the variable-length deduplication can achieve asymptotically arbitrarily large compression ratio.

By Theorem 30 and 31, the variable-length deduplication can achieve arbitrarily large compression ratio but may also spend number of bits larger than entropy by an arbitrarily large factor over the proposed source model.

2) Multi-chunk deduplication

In the following, we derive an upper bound on the performance of multi-chunk algorithm.

Theorem 32. *The average length of the compressed strings by multi-chunk deduplication with optimal marker length M satisfies*

$$\mathbb{E}[\mathcal{L}_{MC}(\boldsymbol{s})] \leq \Theta(AL) + O(B \log(ABL)). \tag{202}$$

Proof: Same as the proof of Theorem 31, we consider the differences between each Y_b and $Y_{h_{J_b}}$, and break $\boldsymbol{s}$ into edit blocks $D_1, D_2, \dots, D_K$.

Next, we consider boundary chunks and interior chunks but with respect to multi-chunking. We give examples in the following about boundary chunks (indicated by underbrackets) and interior chunks (indicated by overbrackets) in multi-chunking when marker length $M = 3$ in various cases. Vertical bars indicate the boundaries of edit blocks. Different rows are independent examples.

$$\cdots 000\,\underbracket{10110|00}\,\underbracket{0101000}\,010\cdots \tag{203}$$

$$\cdots\,\underbracket{101100|0}\,\underbracket{0000}\,\underbracket{0101000}\,010\cdots \tag{204}$$

$$\cdots 000\,\underbracket{10|11000}\,\overbracket{0101000}\,010\cdots \tag{205}$$

$$\cdots 000\,\overbracket{1011000}\,|\,\underbracket{0101000}\,010\cdots \tag{206}$$

$$\cdots 000\,\underbracket{101100|0}\,\underbracket{1000}\,\overbracket{11000}\,10\cdots \tag{207}$$

Further, in the multi-chunk deduplication algorithm, reducing the number of jointly encoded chunks and encoding them separately increase the length of the compressed string. With this observation, we compute an upper bound on $\mathcal{L}_{ML}(\boldsymbol{s})$ in the following by assuming that every boundary chunk is encoded individually.

Let $\mathcal{E}_{in}$ denote the event that any two source alphabet substrings of length 2^{M-2} are of Hamming distance at least $2t+1$ from each other. Given $\mathcal{E}_{in}$, since chunk lengths are at least 2^{M-1}, repeated chunks may only occur at the same position of source blocks that have the same ancestor.

Assume $\mathcal{E}_{in}$ holds. We first consider the interior chunks. For all $1 \leq a \leq A$, in the edit block $D_{k_a} = Y_{g(a)}$, all interior chunks are distinct and different from any other chunk that appeared previously. Thus, they are encoded jointly and take number of bits

$$4 + \log(\left|\mathcal{C}^{\circ}_{k_a}\right|) + \left|Z_{c_a} Z_{c_a+1} \cdots Z_{c_a+\left|\mathcal{C}^{\circ}_{k_a}\right|-1}\right|, \tag{208}$$

where Z_{c_a} is the first interior chunk in D_{k_a}.

For any D_k that lies in some Y_b with $J_b = a$ and $b \neq h_{J_b}$, all of its interior chunks must have appeared in the corresponding positions in $Y_{g(a)}$. Moreover, those are the first times these chunks appeared and entered into the dictionary. Suppose the interior chunks in D_k are $Z_{c_k}, Z_{c_k+1}, \ldots, Z_{c_k+\left|\mathcal{C}^{\circ}_{k}\right|-1}$, they jointly take

$$5 + 2\log\left|\mathcal{C}^{\circ}_{k}\right| + \log\left|T^{c_k-1}_{MC}\right| \tag{209}$$

bits, where $T^{c_k-1}_{MC}$ denotes the dictionary right after chunk Z_{c_k-1} is encoded.

For the boundary chunks, as mentioned, we assume they are encoded individually. If a boundary chunk Z_c is new, then it takes at most $4 + |Z_c|$ bits. If Z_c has appeared before, it takes at most $5 + \log(\left|T^{c-1}_{MC}\right|)$ bits. If we consider $2^{M-1} \geq \log(2BL)$,

$$5 + \log(\left|T^{c-1}_{MC}\right|) \leq 5 + \log(2BL) \leq 5 + |Z_c|. \tag{210}$$

Next, if the complement of $\mathcal{E}_{in}$ holds, then by the same argument as (210), every chunk Z_c can be encoded with at most $5 + |Z_c|$ bits. It follows that

$$\mathcal{L}_{ML}(\boldsymbol{s}) \leq \sum_{c} (5 + |Z_c|) \leq 5 \cdot \frac{2BL}{2^{M-1}} + 2BL \leq 12BL, \tag{211}$$

where the term $\frac{2BL}{2^{M-1}}$ bounds the number of chunks in $\boldsymbol{s}$ from above.

Thus, by combining (208), (209) (210), (211), we get

$$\mathcal{L}_{ML}(\boldsymbol{s}) \leq \sum_{a=1}^{A}\Big(4+\log(\big|\mathcal{C}^{\circ}_{k_a}\big|)+\Big|Z_{c_a}\cdots Z_{c_a+|\mathcal{C}^{\circ}_{k_a}|-1}\Big|\Big) \tag{212}$$

$$+\sum_{k=1}^{K}\big(5+2\log|\mathcal{C}^{\circ}_{k}|+\log\big|T^{c_k-1}_{ML}\big|\big) \tag{213}$$

$$+\sum_{k=1}^{K}\sum_{c\in\partial\mathcal{C}_k}(5+|Z_c|)+I\big(\bar{\mathcal{E}}_{in}\big)\cdot 12BL \tag{214}$$

$$\leq A(4+\log(2L)+2L) \tag{215}$$

$$+B(2t+1)(5+2\log(2L)+\log(2BL)) \tag{216}$$

$$+\sum_{k=1}^{K}\sum_{c\in\partial\mathcal{C}_k}(5+|Z_c|)+I\big(\bar{\mathcal{E}}_{in}\big)\cdot 12BL, \tag{217}$$

where we upper bound both $|\mathcal{C}^{\circ}_{k}|$ and $\Big|Z_{c_a}\cdots Z_{c_a+|\mathcal{C}^{\circ}_{k_a}|-1}\Big|$ by $2L$, upper bound K by $B(2t+1)$ since there are at most $(2t+1)$ edit blocks in each source block, and upper bound the dictionary size $\big|T^{c_k-1}_{MC}\big|$ by $2BL$.

It remains to bound the lengths of boundary chunks. For edit block D, we similarly define $\mathrm{head}(D)$ and $\mathrm{tail}(D)$ as the proof of Theorem 31 but with respect to multi-chunking. We define $\mathrm{tail}(D)$ to be the suffix of D backwards until the beginning of the rightmost occurrence of 0^M plus an additional $2^{M-1}-M-1$ bits, $\mathrm{head}(D)$ to be the prefix of D forwards until the end of the first occurrence of the string $\boldsymbol{u}10^M$ such that $\boldsymbol{u}$ is an M-RLL string of length $2^{M-1}-M-1$, i.e., $\boldsymbol{u}\in R_M^{2^{M-1}-M-1}$ after (exclusive) the first bit of D. Head and tail denote the whole edit block if no such pattern appears. The total length of boundary chunks is again upper bounded by the total length of head and tail strings:

$$\sum_{k=1}^{K}\sum_{c\in\partial\mathcal{C}_k}(5+|Z_c|)\leq\sum_{k=1}^{K}(5|\partial\mathcal{C}_k|+|\mathrm{head}(D_k)|+|\mathrm{tail}(D_k)|). \tag{218}$$

Thus, $\mathbb{E}[\mathcal{L}_{ML}(\boldsymbol{s})]$ is less than

$$A(4+\log(2L)+2L) \tag{219}$$

$$+B(2t+1)(5+2\log(2L)+\log(2BL)) \tag{220}$$

$$+\sum_{k=1}^{K}(5\mathbb{E}[|\partial\mathcal{C}_k|]+\mathbb{E}[|\mathrm{head}(D_k)|]+\mathbb{E}[|\mathrm{tail}(D_k)|]) \tag{221}$$

$$+\mathbb{P}\big(\bar{\mathcal{E}}_{in}\big)\cdot 12BL. \tag{222}$$

By [77, Appendix F],

$$\mathbb{E}[|\mathrm{head}(D_k)|]\leq 2^{M+2}+1, \mathbb{E}[|\mathrm{tail}(D_k)|]\leq 2^{M+2}. \tag{223}$$

Moreover,

$$\sum_{k=1}^{K}\mathbb{E}[|\partial\mathcal{C}_k|]\leq\mathbb{E}\left[\frac{\sum_{k=1}^{K}(|\mathrm{head}(D_k)|+|\mathrm{tail}(D_k)|)}{2^{M-1}}\right]. \tag{224}$$

It follows that

$$\sum_{k=1}^{K}(5\mathbb{E}[|\partial\mathcal{C}_k|]+\mathbb{E}[|\mathrm{head}(D_k)|]+\mathbb{E}[|\mathrm{tail}(D_k)|]) \tag{225}$$

$$\leq B(2t+1)2^{M+4}+5\cdot 2^5\cdot(2t+1)B. \tag{226}$$

On the other hand, by a similar argument as (167),

$$\mathbb{P}(\bar{\mathcal{E}}_{in}) \le (2AL)^2 \frac{\sum_{m=0}^{2t+1}\binom{2^{M-2}}{m}}{2^{2^{M-2}}} \le (2AL)^2 \frac{(2^{M-2})^{2t+1}}{2^{2^{M-2}}} \tag{227}$$

$$\le (2AL)^2 \frac{1}{2^{2^{M-3}}}, \tag{228}$$

for all $M \ge 3 + 2\log(2t+1)$.

Thus, by plugging (226) and (228) into (222), we get

$$\mathbb{E}[\mathcal{L}_{ML}(\boldsymbol{s})] \le A(5 + \log L + 2L) + 48A^2BL^3 2^{-2^{M-3}} \tag{229}$$

$$+(2t+1)B\big(2^{M+4} + 168 + 2\log L + \log(BL)\big), \tag{230}$$

subject to the condition that $2^{M-1} \ge \log(2BL)$.

Pick

$$M = \big\lceil \log\log\big(\max\big(4A^2L^3, 2BL\big)\big) + 1\big\rceil. \tag{231}$$

Then $2^{M-1} \ge \log(2BL)$ is satisfied. With this choice of M, (230) is further upper bounded by

$$A(5 + \log L + 2L) + 48B + (2t+1)B\big(32\log\big(4A^2L^3 + 2BL\big) \tag{232}$$

$$+168 + 2\log L + \log(BL)) \tag{233}$$

$$=\Theta(AL) + O(B\log(ABL)). \tag{234}$$

■

By the preceding theorem, when $\log B = O(\log L)$,

$$\frac{\mathbb{E}[\mathcal{L}_{ML}(\boldsymbol{s})]}{\mathbb{E}[|\boldsymbol{s}|]} \le O(1). \tag{235}$$

Note that the $O(1)$ term depends on the substitution number t. Therefore, with the existence of substitutions, the multi-chunk algorithm can achieve a constant factor of optimal with respect to the entropy.

F. Data deduplication from the point of view of universal compression

Existing theoretic analysis of data deduplication presented in Sections II-C, II-D and II-E assume known sources and compare the expected length of the compressed strings with source entropy since entropy is the fundamental limit for any uniquely decodable representation. However, in practice the specific probability distribution underlying the data is usually unknown. For example, given a large chunk of texts or a set of images, the distribution of words/pixels is not observable. Similarly, for data storage systems, it is often hard to do estimation about the underlying distribution given the scale. Instead, a common assumption in these situations is to assume that there is a class of distributions $\mathcal{P}$ to which the true distribution belongs, but the precise distribution is unknown. Thus, analysis and evaluation of compressors must take all possible sources into account. A good compressor should have 'universality' over all possible sources instead of just having performance approaching the entropy of a certain source distribution.

Motivated by applications like data deduplication, we study compressors under low-complexity constraints in the framework of universal compression in this section. We assume that due to the complexity restriction, there are groups of input data that compressors can not distinguish. As a result, the whole data space $\mathcal{X}$ is partitioned into groups uniquely defined by the constraint and compressors must assign elements in the same group with the same probability. General results on the worst and average cases redundancies with respect to the constrained compressors are derived. In particular, we consider universal compression of patterns generated by iid sources over an alphabet of size k but with constrained compressors. We consider the constraint that compressors are only allowed to use the information about how many distinct symbols (integers) are there in the pattern sequence. In other words, patterns with the same number of distinct integers are assigned with the same probability. We compute the worst and average case redundancies for such compressors. It is shown that under this constraint, the per-symbol redundancies are at least a constant number of bits while diminishing redundancy can be achieved without any constraint. We also show that the encoding scheme presented in [77] is optimal up to the first-order term under constraint.

1) The general universal compression framework

We first give an introduction to the general universal coding framework, where the metric ‘redundancy’ for compressors is defined. Let a source X be distributed over a discrete support set $\mathcal{X}$ according to a distribution p. Every compressor of X corresponds to a probability distribution q over $\mathcal{X}$ where $x \in \mathcal{X}$ is represented by roughly $\log(1/q(x))$ bits. Shannon’s source coding theorem states that the optimal way of compressing a source X is to represent each outcome x with $\log(1/p(x))$ bits. The extra number of bits required to represent x when q is used instead of p is therefore

$$\log\frac{1}{q(x)} - \log\frac{1}{p(x)} = \log\frac{p(x)}{q(x)}. \tag{236}$$

The *worst case* redundancy of q with respect to $\mathcal{P}$ is defined as the largest number of extra bits used for any possible x and any distribution p, i.e.,

$$\hat{R}(\mathcal{P}, q) = \sup_{p\in\mathcal{P}}\sup_{x\in\mathcal{X}}\log\frac{p(x)}{q(x)} = \sup_{x\in\mathcal{X}}\log\frac{\hat{p}(x)}{q(x)}, \tag{237}$$

where $\hat{p}(x) = \sup_{p\in\mathcal{P}} p(x)$, the maximum probability of x assigned by any $p \in \mathcal{P}$.

Let $\mathcal{Q}$ be the set of all compressors (distributions) over $\mathcal{X}$. The *worst case* redundancy of $\mathcal{P}$ is defined as

$$\hat{R}(\mathcal{P}, \mathcal{Q}) = \hat{R}(\mathcal{P}) = \inf_{q\in\mathcal{Q}}\hat{R}(\mathcal{R}, q) = \inf_{q\in\mathcal{Q}}\sup_{x\in\mathcal{X}}\log\frac{\hat{p}(x)}{q(x)}, \tag{238}$$

the lowest number of extra bits in the worst case required by any compressor.

Similarly, one can define the *average case* redundancy of $\mathcal{P}$ as

$$\bar{R}(\mathcal{P}, \mathcal{Q}) = \bar{R}(\mathcal{P}) = \inf_{q\in\mathcal{Q}}\sup_{p\in\mathcal{P}}\left(\sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}\right), \tag{239}$$

the lowest number of extra bits on average required by any compressor. It can also be shown [17, Theorem 13.1.1] that

$$\bar{R}(\mathcal{P}) = \sup_{\pi\in\Pi}\inf_{q\in\mathcal{Q}}\mathbb{E}_{p\sim\pi}\left[\sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}\right], \tag{240}$$

where Π is the set of all distributions over $\mathcal{P}$. Note that it is always true that $\bar{R}$ and $\hat{R}$ are nonnegative and $\hat{R}$ is an upper bound on $\bar{R}$.

2) Universal compression of iid sources over patterns

The class of iid sources that generate length-n sequences over $\mathcal{A}$ is denoted $\mathcal{I}_k^n$. Let $\Theta_k = \{(\theta_1, \theta_2, \ldots, \theta_k) : \sum_{i=1}^k \theta_i = 1, 0 \le \theta_i \le 1\}$. Each $p_{\boldsymbol{\theta}} \in \mathcal{I}_k^n$ is then parameterized by a vector $\boldsymbol{\theta} \in \Theta_k$.

Each $p_{\boldsymbol{\theta}}$ induces a distribution over $\Psi_{\le k}^n$ as

$$p_{\boldsymbol{\theta}}(\boldsymbol{\psi}) = \sum_{x^n:\Psi(x^n)=\boldsymbol{\psi}} p_{\boldsymbol{\theta}}(x^n). \tag{241}$$

For example, let $k = 2$, $n = 2$. For $\boldsymbol{\theta} = (0.4, 0.6)$, the induced pattern distribution is

$$p_{\boldsymbol{\theta}}(11) = 0.4^2 + 0.6^2 = 0.52, \tag{242}$$

$$p_{\boldsymbol{\theta}}(12) = 2 \times 0.4 \times 0.6 = 0.48. \tag{243}$$

Note the dual use of $p_{\boldsymbol{\theta}}$ that $p_{\boldsymbol{\theta}}(x^n)$ denotes the probability of sequence x^n and $p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$ denotes the induced probability of pattern $\boldsymbol{\psi}$.

As mentioned, we are interested in universal compression of patterns generated by iid sources over alphabets of size k. Let $\mathcal{I}_\Psi^{n,k}$ denote the set of pattern distributions over $\Psi_{\le k}^n$ induced by $\mathcal{I}_k^n$. From (237), the worst case redundancy of the class $\mathcal{I}_\Psi^{n,k}$ with respect to a compressor q equals

$$\hat{R}\left(\mathcal{I}_\Psi^{n,k}, q\right) = \sup_{\boldsymbol{\psi}\in\Psi_{\le k}^n}\log\frac{\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{q(\boldsymbol{\psi})}, \tag{244}$$

where $\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi}) = \sup_{\boldsymbol{\theta}\in\Theta_k} p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$. Let $\mathcal{Q}$ denote the set of all distributions over $\Psi_{\le k}^n$. the worst case redundancy of $\mathcal{I}_\Psi^{n,k}$ with respect to a set $\mathcal{Q}$ of compressors equals

$$\hat{R}\Big(\mathcal{I}_{\Psi}^{n,k},\mathcal{Q}\Big)=\hat{R}\Big(\mathcal{I}_{\Psi}^{n,k}\Big)=\inf_{q\in\mathcal{Q}}\sup_{\boldsymbol{\psi}\in\Psi_{\le k}^{n}}\log\frac{\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{q(\boldsymbol{\psi})},\tag{245}$$

where $\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi})=\sup_{\boldsymbol{\theta}\in\Theta_k}p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$.

Let Π denote the set of all distributions over Θ_k. From (240), the average case redundancy of $\mathcal{I}_{\Psi}^{n,k}$ with respect to $\mathcal{Q}$ equals

$$\bar{R}\Big(\mathcal{I}_{\Psi}^{n,k},\mathcal{Q}\Big)=\sup_{\pi\in\Pi}\inf_{q\in\mathcal{Q}}\mathbb{E}_{\boldsymbol{\theta}\sim\pi}\left[\left(\sum_{\boldsymbol{\psi}\in\Psi_{\le k}^{n}}p_{\boldsymbol{\theta}}(\boldsymbol{\psi})\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{q(\boldsymbol{\psi})}\right)\right].\tag{246}$$

Universal compression of patterns was first introduced in [1], which leads to a lower bound on the worst case pattern redundancy over $\mathcal{I}_{\Psi}^{n,k}$ of

$$\hat{R}\Big(\mathcal{I}_{\Psi}^{n,k}\Big)\ge(1-\epsilon)(k-1)\log\frac{n}{k^3}(1+o(1)),\tag{247}$$

where $\epsilon>0$ can be made arbitrarily small. Note that this bound is useful only when $k=o(n^{\frac{1}{3}})$. Later In [82], compression of patterns generated by the class of iid sources of arbitrarily large alphabet size, i.e., $\mathcal{I}_{\Psi}^{n}=\cup_{k=1}^{\infty}\mathcal{I}_{\Psi}^{n,k}=\mathcal{I}_{\Psi}^{n,n}$, is considered. Upper and lower bounds on the worst case redundancy with respect to $\mathcal{I}_{\Psi}^{n}$ are shown as

$$\left(\frac{3}{2}\log e\right)n^{1/3}(1+o(1))\le\hat{R}(\mathcal{I}_{\Psi}^{n})\le\left(\pi\sqrt{\frac{2}{3}}\log e\right)n^{1/2}.\tag{248}$$

More recently, [2] gave the upper bound

$$\hat{R}(\mathcal{I}_{\Psi}^{n})\le n^{1/3}(\log n)^4,\tag{249}$$

and thus showed together with (248) that $\hat{R}(\mathcal{I}_{\Psi}^{n})$ is $\tilde{\Theta}\big(n^{1/3}\big)$.

Upper and lower bounds on the average case redundancy of $\mathcal{I}_{\Psi}^{n,k}$ was given in [96]. In particular, lower bounds were shown as

$$\bar{R}\Big(\mathcal{I}_{\Psi}^{n,k}\Big)\ge\begin{cases}\frac{k-1}{2n}\log\frac{n^{(1-\epsilon)}}{k^3}(1+o(1)), \text{ for } k\le\left(\frac{\pi n^{(1-\epsilon)}}{2}\right)^{1/3}\\ \left(\frac{\pi}{2}\right)^{1/3}\left(\frac{3}{2}\log e\right)n^{(1-\epsilon)/3}(1+o(1)), \text{ for } k>\left(\frac{\pi n^{1-\epsilon}}{2}\right)^{1/3}.\end{cases}\tag{250}$$

Papers [2], [3], [9], [38] considered the average case redundancy for encoding patterns with arbitrarily large alphabet size, i.e., $\bar{R}(\mathcal{I}_{\Psi}^{n})$. The tightest bounds are given in [2], [3] as

$$0.3n^{1/3}\le\bar{R}(\mathcal{I}_{\Psi}^{n})\le n^{1/3}(\log n)^{4/3}.\tag{251}$$

3) Universality of constrained compressors

In this section, we present general results for computing the redundancies of constrained compressors.

We consider constraints resulting from complexity restrictions. Compressors thus do not have enough resources to fully process the information, thus causing some data inputs to be indistinguishable. With this intuition, we assume that every constraint $\mathcal{C}$ uniquely defines a partition of the support set $\mathcal{X}$ as

$$\mathcal{X}=\cup_{j=1}^{K}C_j.\tag{252}$$

Under $\mathcal{C}$, elements in the same partition set are indistinguishable, i.e., they must be assigned with the same probability. So we use

$$\mathcal{Q}_c=\{q:q(x_1)=q(x_2)\text{ if }x_1\sim x_2\}\tag{253}$$

to denote the set of allowed compressors under $\mathcal{C}$, where $\sim$ denotes the equivalence relation that x_1 and x_2 belong to the same partition set.

For a generic distribution p over $\mathcal{X}$, we use $\tilde{p}$ to denote the distribution over the partition sets $\{C_j\}_{j=1}^K$ induced by p, i.e.,

$$\tilde{p}(j) = \sum_{x \in C_j} p(x), \quad j = 1, 2, \dots, K. \tag{254}$$

For a family of distributions $\mathcal{P}$, let $\tilde{\mathcal{P}} = \{\tilde{p} : p \in \mathcal{P}\}$.

Moreover, if we distribute $\tilde{p}(j)$ back evenly to all $x \in C_j$, we get the flatten distribution of p, denoted $\bar{p}$, as

$$\bar{p}(x) = \frac{\tilde{p}(j)}{|C_j|} = \frac{\sum_{x \in C_j} p(x)}{|C_j|}, \quad \text{for } x \in C_j. \tag{255}$$

In the following, we present two lemmas about the minimum redundancy that compressors in $\mathcal{Q}_c$ can achieve, in both worst and average cases.

Lemma 33. *The worst case redundancy of $\mathcal{P}$ with respect to the set of constrained compressors $\mathcal{Q}_c$ satisfies*

$$\hat{R}(\mathcal{P}, \mathcal{Q}_c) = \log\left(\sum_{j=1}^K \left(|C_j| \cdot \sup_{x \in C_j} \hat{p}(x)\right)\right), \tag{256}$$

where $\hat{p}(x) = \sup_{p \in \mathcal{P}} p(x)$.

Proof: From (238),

$$\hat{R}(\mathcal{P}, \mathcal{Q}_c) = \inf_{q \in \mathcal{Q}_c} \sup_{x \in \mathcal{X}} \log \frac{\hat{p}(x)}{q(x)}. \tag{257}$$

Since every $q \in \mathcal{Q}_c$ assigns the same probability to elements in the same partition set, we replace $q(x)$ with q_j for all $x \in C_j$.

For any q,

$$\sup_{x \in \mathcal{X}} \log \frac{\hat{p}(x)}{q(x)} = \sup_{j=1,\dots,K} \sup_{x \in C_j} \log \frac{\hat{p}(x)}{q_j} \tag{258}$$

$$= \sup_{j=1,\dots,K} \log \frac{\sup_{x \in C_j} \hat{p}(x)}{q_j}. \tag{259}$$

Similar to Shtarkov's result [100], (259) is minimized at $q_j^* \propto \sup_{x \in C_j} \hat{p}(x)$, i.e.,

$$q_j^* = \frac{\sup_{x \in C_j} \hat{p}(x)}{\sum_{j=1}^K \sum_{x \in C_j} \sup_{x \in C_j} \hat{p}(x)}. \tag{260}$$

The redundancy thus equals log of the normalizer, i.e.,

$$\log\left(\sum_{j=1}^K \sum_{x \in C_j} \sup_{x \in C_j} \hat{p}(x)\right) = \log\left(\sum_{j=1}^K \left(|C_j| \cdot \sup_{x \in C_j} \hat{p}(x)\right)\right). \tag{261}$$

■

It can be seen from the proof that the lowest redundancy in worst case is achieved by assigning each x with probability proportional to the largest maximum probability in the same partition set. Note that when there is no constraint, i.e., the corresponding partition of $\mathcal{X}$ is $\mathcal{X} = \cup_x \{x\}$, $\hat{R}(\mathcal{P}, \mathcal{Q})$ is thus reduced to $\log(\sum_x \hat{p}(x))$, which was given in [100].

Lemma 34. *The average case redundancy of $\mathcal{P}$ with respect to the set of constrained compressors $\mathcal{Q}_c$ satisfies*

$$L(\mathcal{P}, \mathcal{Q}_c) \le \bar{R}(\mathcal{P}, \mathcal{Q}_c) \le U(\mathcal{P}, \mathcal{Q}_c), \tag{262}$$

where

$$L(\mathcal{P}, \mathcal{Q}_c) = \max\left(\sup_{p \in \mathcal{P}} D(p||\bar{p}), \bar{R}\left(\tilde{\mathcal{P}}\right)\right), \tag{263}$$

$$U(\mathcal{P}, \mathcal{Q}_c) = \sup_{p \in \mathcal{P}} D(p||\bar{p}) + \bar{R}\left(\tilde{\mathcal{P}}\right). \tag{264}$$

Proof: From (240),

$$\bar{R}(\mathcal{P}, \mathcal{Q}_c) = \sup_{\pi \in \Pi} \inf_{q \in \mathcal{Q}_c} \mathbb{E}_{p \sim \pi}\left[\sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\right] \tag{265}$$

$$\inf_{q \in \mathcal{Q}_c} \sup_{p \in \mathcal{P}} \left(\sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\right). \tag{266}$$

Let Π be the set of probability distributions over $\mathcal{P}$. It can be shown [17, Theorem 13.1.1] that

$$\inf_{q \in \mathcal{Q}_c} \sup_{p \in \mathcal{P}} \left(\sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\right) = \sup_{\pi \in \Pi} \inf_{q \in \mathcal{Q}_c} \mathbb{E}_{p \sim \pi}\left[\sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\right]. \tag{267}$$

Further,

$$\sup_{\pi \in \Pi} \inf_{q \in \mathcal{Q}_c} \mathbb{E}_{p \sim \pi}\left[\sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\right] \tag{268}$$

$$= \sup_{\pi \in \Pi} \inf_{q \in \mathcal{Q}_c} \mathbb{E}_{p \sim \pi}\left[\sum_{j=1}^{K} \sum_{x \in C_j} \left(p(x) \log \frac{p(x)}{\bar{p}(x)}\right) + \sum_{j=1}^{K} \tilde{p}(j) \log \frac{\tilde{p}(j)}{\tilde{q}(j)}\right] \tag{269}$$

$$= \sup_{\pi \in \Pi} \left(\mathbb{E}_{p \sim \pi}[D(p||\bar{p})] + \inf_{q \in \mathcal{Q}_c} \mathbb{E}_{p \sim \pi}[D(\tilde{p}||\tilde{q})]\right), \tag{270}$$

where the second equality follows from $\frac{\bar{p}(x)}{p(x)} = \frac{\tilde{p}(j)}{\tilde{q}(j)}$ for all $x \in C_j$.

Further, we have

$$\sup_{\pi \in \Pi} \mathbb{E}_{p \sim \pi}[D(p||\bar{p})] = \sup_{p \in \mathcal{P}} D(p||\bar{p}), \tag{271}$$

and by definition,

$$\sup_{\pi \in \Pi} \inf_{q \in \mathcal{Q}_c} \mathbb{E}_{p \sim \pi}[D(\tilde{p}||\tilde{q})] = \bar{R}\left(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}_c\right) = \bar{R}\left(\tilde{\mathcal{P}}\right). \tag{272}$$

The desired inequalities thus follow from the fact that for any set $\mathcal{Y}$,

$$\sup_{y \in \mathcal{Y}}(f_1(y) + f_2(y)) \geq \max\left(\sup_{y \in \mathcal{Y}} f_1(y), \sup_{y \in \mathcal{Y}} f_2(y)\right), \tag{273}$$

$$\sup_{y \in \mathcal{Y}}(f_1(y) + f_2(y)) \leq \sup_{y \in \mathcal{Y}} f_1(y) + \sup_{y \in \mathcal{Y}} f_2(y). \tag{274}$$

■

Lemma 34 shows that the lowest average case redundancy under constraint is determined by two terms, $\sup_{p \in \mathcal{P}} D(p||\bar{p})$ and $\bar{R}\left(\tilde{\mathcal{P}}\right)$. The former is the maximum KL-divergence between source p and its flattened distribution $\tilde{p}$ for all $p \in \mathcal{P}$. The latter is the average case redundancy of the set of induced distribution $\tilde{P}$. Note that when there is no constraint, $p, \bar{p}$ and $\tilde{p}$ are identical, so both upper and lower bounds are reduced to $\bar{R}\left(\tilde{\mathcal{P}}\right) = \bar{R}(\mathcal{P})$.

4) Universal compression of patterns under constraint $\mathcal{C}_1$

In this section, we consider a specific set of constrained pattern compressors and present lower bounds on the worst and average case redundancies for encoding patterns generated by iid sources over an alphabet of size k. The constraint $\mathcal{C}_1$ is motivated by data deduplication algorithms and it requires that compressors encode patterns only according to the number of distinct integers. In other words, patterns that contain the same number of distinct integers should be assigned with the same probability.

We find lower bounds on redundancies for compressing patterns only according to the number of distinct index integers. The partition of $\Psi^n_{\leq k}$ defined by $\mathcal{C}_1$ is

$$\Psi^n_{\leq k} = \cup_{m=1}^{k} \Psi^n_m. \tag{275}$$

The set of allowed compressors is

$$\mathcal{Q}_1 = \{q : q(\boldsymbol{\psi}_1) = q(\boldsymbol{\psi}_2) \text{ if } N(\boldsymbol{\psi}_1) = N(\boldsymbol{\psi}_2)\}. \tag{276}$$

Theorem 35. *As $n \to \infty$, $\hat{R}\Big(\mathcal{I}_{\Psi}^{n,k}, \mathcal{Q}_1\Big)$ is greater than or equal to*

$$\begin{cases} (n \log k - k \log n)(1 + o(1)), & \textit{for } k \leq \frac{n}{\ln n}, \\ n(\log n - \log \log n)(1 + o(1)), & \textit{for } k > \frac{n}{\ln n}. \end{cases} \tag{277}$$

Proof: By Lemma 33,

$$\hat{R}\Big(\mathcal{I}_{\Psi}^{n,k}, \mathcal{Q}_1\Big) = \log\Bigg(\sum_{m=1}^{k}\Bigg(|\Psi_m^n| \cdot \sup_{\boldsymbol{\psi} \in \Psi_m^n} \hat{p}_{\Psi}(\boldsymbol{\psi})\Bigg)\Bigg), \tag{278}$$

where $\hat{p}_{\Psi}(\boldsymbol{\psi}) = \sup_{\boldsymbol{\theta} \in \Theta_k} p_{\boldsymbol{\theta}}(\boldsymbol{\psi})$ is the maximum probability of pattern $\boldsymbol{\psi}$.

For a pattern $\boldsymbol{\psi}$ with profile $\Phi(\boldsymbol{\psi}) = (\varphi_1, \dots, \varphi_n)$, it was pointed out in [82] that the maximum probability assigned by any iid distribution satisfies

$$\hat{p}_{\Psi}(\boldsymbol{\psi}) \geq \sum_{\mu=1}^{n} \varphi_{\mu}!\Big(\frac{\mu}{n}\Big)^{\mu\varphi_{\mu}}. \tag{279}$$

Consider any $m < n$. Let $\bar{\boldsymbol{\psi}}^m$ be any pattern sequence in Ψ_m^n such that integer 1 appears $n - m + 1$ times and each of the integers $2, 3, \dots, m$ appears only once.

We lower bound $\sup_{\boldsymbol{\psi} \in \Psi_m^n} \hat{p}_{\Psi}(\boldsymbol{\psi})$ by $\hat{p}_{\Psi}\big(\bar{\boldsymbol{\psi}}^m\big)$. The profile of $\bar{\boldsymbol{\psi}}^m$ equals $\Phi\big(\bar{\boldsymbol{\psi}}^m\big) = (\bar{\varphi}_1^m, \dots, \bar{\varphi}_n^m)$ where

$$\bar{\varphi}_{n-m+1}^m = 1, \quad \bar{\varphi}_1^m = m - 1, \tag{280}$$

and $\bar{\varphi}_i^m = 0$ for all other values of i. By (279),

$$\hat{p}_{\Psi}\big(\bar{\boldsymbol{\psi}}^m\big) \geq (m-1)!\frac{(n-m+1)^{n-m+1}}{n^n} \tag{281}$$

$$\geq \sqrt{\frac{2\pi}{m}}\frac{m^m}{e^m}\frac{n^{n-m+1}\big(1 - \frac{m-1}{n}\big)^{n-m+1}}{n^n} \tag{282}$$

$$= \Big(\frac{m}{n}\Big)^{m-1}\frac{\sqrt{2\pi m}}{e^m}\Big(1 - \frac{m-1}{n}\Big)^{n-m+1}, \tag{283}$$

where the second inequality follows from Feller's bounds on Stirling's approximation [37] that for any $m \geq 1$,

$$m! \geq \sqrt{2\pi m}\Big(\frac{m}{e}\Big)^m. \tag{284}$$

Next, we compute $|\Psi_m^n|$. There is a one-to-one correspondence between Ψ_m^n and the set of unordered m-partitions of $[n]$. The number of m-partitions of $[n]$ is known as the stirling number of the second kind and is lower bounded in [89] by

$$\frac{1}{2}\big(m^2 + m + 1\big)m^{n-m-1} - 1, \tag{285}$$

for $1 \leq m \leq n - 1$.

Plugging (283) and (285) into (278) gives

$$\hat{R}\Big(\mathcal{I}_{\Psi}^{n,k},\mathcal{Q}_1\Big) = \log\left(\sum_{m=1}^{k}\left(|\Psi_m^n|\cdot\sup_{\boldsymbol{\psi}\in\Psi_m^n}\hat{p}_{\Psi}(\boldsymbol{\psi})\right)\right) \tag{286}$$

$$\geq \log\left(\sum_{m=1}^{\min(n-1,k)}|\Psi_m^n|\cdot\hat{p}_{\Psi}\big(\bar{\boldsymbol{\psi}}^m\big)\right) \tag{287}$$

$$\geq \log\left(\sum_{m=1}^{\min(n-1,k)}\left(\left(\frac{1}{2}\big(m^2+m+1\big)m^{n-m-1}-1\right)\right.\right. \tag{288}$$

$$\left.\left.\cdot\left(\frac{m}{n}\right)^{m-1}\frac{\sqrt{2\pi m}}{e^m}\left(1-\frac{m-1}{n}\right)^{n-m+1}\right)\right) \tag{289}$$

$$\geq \log\left(\left(\frac{1}{2}\big(m^2+m+1\big)m^{n-m-1}-1\right)\right. \tag{290}$$

$$\left.\cdot\left(\frac{m}{n}\right)^{m-1}\frac{\sqrt{2\pi m}}{e^m}\left(1-\frac{m-1}{n}\right)^{n-m+1}\right)\Bigg|_{m=\min\left(\left\lfloor\frac{n}{\ln n}\right\rfloor,k\right)} \tag{291}$$

$$= (n\log m - m\log n)(1+o(1))\Big|_{m=\min\left(\left\lfloor\frac{n}{\ln n}\right\rfloor,k\right)} \tag{292}$$

$$= \begin{cases}(n\log k - k\log n)(1+o(1)), & \text{for } k\leq\frac{n}{\ln n}\\ n(\log n-\log\log n)(1+o(1)), & \text{for } k>\frac{n}{\ln n}.\end{cases} \tag{293}$$

■

Theorem 36. *Fix an arbitrarily small* $\epsilon>0$. *As* $n\to\infty$, $\bar{R}\Big(\mathcal{I}_{\Psi}^{n,k},\mathcal{Q}_1\Big)$ *is greater than*

$$\begin{cases}\Big(n\log k-(\log e)k(\ln n)^2\Big)(1+o(1)), \ \textit{for } k<\left(\frac{n}{\ln n}\right)^{1-\epsilon},\\ (1-\epsilon)n(\log n-\log\log n)(1+o(1)), \ \textit{for } k\geq\left(\frac{n}{\ln n}\right)^{1-\epsilon}.\end{cases} \tag{294}$$

Proof: By Lemma 34, $\bar{R}\Big(\mathcal{I}_{\Psi}^{n,k},\mathcal{Q}_1\Big)$ is lower bounded by[5]

$$\max\left(\sup_{\boldsymbol{\theta}\in\Theta_k}D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}),\bar{R}\Big(\tilde{\mathcal{I}}_{\Psi}^{n,k}\Big)\right)\geq\sup_{\boldsymbol{\theta}\in\Theta_k}D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}). \tag{295}$$

Recall that $\bar{p}_{\boldsymbol{\theta}}$ is the flattened distribution of $p_{\boldsymbol{\theta}}$ with respect to the partition $\cup_{m=1}^{k}\Psi_m^n$ and $\tilde{\mathcal{I}}_{\Psi}^{n,k}$ is the set of distributions over $[k]$ induced by $\mathcal{I}_{\Psi}^{n,k}$.

To find a lower bound on $\sup_{\boldsymbol{\theta}\in\Theta_k}D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}})$, we write

$$D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}) = \sum_{\boldsymbol{\psi}\in\Psi_{\leq k}^n}p_{\boldsymbol{\theta}}(\boldsymbol{\psi})\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{\bar{p}_{\boldsymbol{\theta}}(\boldsymbol{\psi})} \tag{296}$$

$$= \sum_{m=1}^{k}\sum_{\boldsymbol{\psi}\in\Psi_m^n}p_{\boldsymbol{\theta}}(\boldsymbol{\psi})\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{\psi})}{\frac{1}{|\Psi_m^n|}\sum_{\boldsymbol{\psi}'\in\Psi_m^n}p_{\boldsymbol{\theta}}(\boldsymbol{\psi}')} \tag{297}$$

$$= \sum_{m=1}^{k}\sum_{\boldsymbol{\psi}\in\Psi_m^n}\left(p_{\boldsymbol{\theta}}(\boldsymbol{\psi})\log p_{\boldsymbol{\theta}}(\boldsymbol{\psi})+p_{\boldsymbol{\theta}}(\boldsymbol{\psi})\log|\Psi_m^n|+p_{\boldsymbol{\theta}}(\boldsymbol{\psi})\log\frac{1}{\sum_{\boldsymbol{\psi}'\in\Psi_m^n}p_{\boldsymbol{\theta}}(\boldsymbol{\psi}')}\right) \tag{298}$$

$$= -H_{\boldsymbol{\theta}}(\boldsymbol{\psi})+\sum_{m=1}^{k}p_{\boldsymbol{\theta}}(m)\log|\Psi_m^n|+\sum_{m=1}^{k}p_{\boldsymbol{\theta}}(m)\log\frac{1}{p_{\boldsymbol{\theta}}(m)}, \tag{299}$$

where $H_{\boldsymbol{\theta}}(\boldsymbol{\psi})$ is the entropy of the pattern distribution parameterized by $\boldsymbol{\theta}$ and $p_{\boldsymbol{\theta}}(m)=\sum_{\boldsymbol{\psi}\in\Psi_m^n}p_{\boldsymbol{\theta}}(\boldsymbol{\psi})=\Pr(\boldsymbol{\psi}\in\Psi_m^n|\boldsymbol{\theta})$.

[5] $\bar{R}\Big(\tilde{\mathcal{I}}_{\Psi}^{n,k}\Big)$ can be shown to be upper bounded by $\log k$, which can be seen later to be negligible. So it suffices to only consider $\sup_{\boldsymbol{\theta}\in\Theta_k}D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}})$.

Consider $J = \min\Big(k, \Big\lfloor\big(\frac{n}{\ln n}\big)^{1-\epsilon}\Big\rfloor\Big)$ for a small $\epsilon > 0$ and vector $\boldsymbol{\theta}_J = (\theta_1, \theta_2, \ldots, \theta_k) \in \Theta_k$ where

$$\theta_j = \begin{cases} 1-(J-1)\frac{\ln n}{n}, & j = 1, \\ \frac{\ln n}{n}, & j = 2, 3, \ldots, J, \\ 0, & \text{otherwise}. \end{cases} \tag{300}$$

Note that since $J \le \big(\frac{n}{\ln n}\big)^{1-\epsilon}$, $\theta_1 \ge 1 - \big(\frac{n}{\ln n}\big)^{-\epsilon}$.

We bound $\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}})$ from below by $D(p_{\boldsymbol{\theta}_J}||\bar{p}_{\boldsymbol{\theta}_J})$, which by (299) equals

$$-H_{\boldsymbol{\theta}_J}(\boldsymbol{\psi}) + \sum_{m=1}^{k} p_{\boldsymbol{\theta}_J}(m)\log|\Psi_m^n| + \sum_{m=1}^{k} p_{\boldsymbol{\theta}_J}(m)\log\frac{1}{p_{\boldsymbol{\theta}_J}(m)}. \tag{301}$$

Since the distributions over patterns are induced by the iid distributions over sequences,

$$H_{\boldsymbol{\theta}_J}(\boldsymbol{\psi}) \le H_{\boldsymbol{\theta}_J}(x^n) = n\Bigg(\bigg(1-(J-1)\frac{\ln n}{n}\bigg)\log\frac{1}{1-(J-1)\frac{\ln n}{n}} \tag{302}$$

$$+ (J-1)\frac{\ln n}{n}\log\frac{n}{\ln n}\Bigg) \tag{303}$$

$$< J\ln n\log\frac{ne}{\ln n}, \tag{304}$$

where the last inequality follows $(1-x)\log\frac{1}{1-x} < x\log(e)$ for all $0 < x < 1$.

For the second term in (301), we show that in the original sequence x^n, all of the J symbols with positive probabilities will appear with high probability, i.e., $p_{\boldsymbol{\theta}_J}(J) \approx 1$, thus leading to a lower bound approximately $\log|\Psi_J^n|$. Rigorously, given $\boldsymbol{\theta}_J$, in the original sequence x^n, the probability that any symbol does not appear is less than or equal to

$$\left(1-\frac{\ln n}{n}\right)^n \le \frac{1}{n}. \tag{305}$$

By the union bound, the probability that all J symbols appear is greater than or equal to

$$1-\frac{J}{n} \ge 1-\Big(\frac{n}{\ln n}\Big)^{-\epsilon}\frac{1}{\ln n}. \tag{306}$$

So $p_{\boldsymbol{\theta}_J}(J) \ge 1-\big(\frac{n}{\ln n}\big)^{-\epsilon}\frac{1}{\ln n}$ and

$$\sum_{m=1}^{k} p_{\boldsymbol{\theta}_J}(m)\log|\Psi_m^n| \ge \left(1-\Big(\frac{n}{\ln n}\Big)^{-\epsilon}\frac{1}{\ln n}\right)\log|\Psi_J^n| \tag{307}$$

$$\ge \left(1-\Big(\frac{n}{\ln n}\Big)^{-\epsilon}\frac{1}{\ln n}\right)\log\left(\frac{1}{2}\big(J^2+J+1\big)J^{n-J-1}-1\right), \tag{308}$$

where the last inequality follows again from (285).

Combining (301), (304), (308) and trivially lower bounding the last term in (301) by 0 give

$$\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}) \ge D(p_{\boldsymbol{\theta}_J}||\bar{p}_{\boldsymbol{\theta}_J}) \tag{309}$$

$$> -J\ln n\log\frac{ne}{\ln n} \tag{310}$$

$$+ \left(1-\Big(\frac{n}{\ln n}\Big)^{-\epsilon}\frac{1}{\ln n}\right)\log\left(\frac{1}{2}\big(J^2+J+1\big)J^{n-J-1}-1\right) \tag{311}$$

$$= \big(-\log e\cdot J(\ln n)^2 + (n-J+1)\log J\big)(1+o(1)) \tag{312}$$

$$= \begin{cases} \Big(n\log k - (\log e)k(\ln n)^2\Big)(1+o(1)), & \text{for } k < \big(\frac{n}{\ln n}\big)^{1-\epsilon}, \\ (1-\epsilon)n(\log n - \log\log n)(1+o(1)), & \text{for } k \ge \big(\frac{n}{\ln n}\big)^{1-\epsilon}. \end{cases} \tag{313}$$

■

Theorems 35 and 36 show that if compressors can only use the information about the number of distinct integers in the pattern sequence, the worst and average redundancies are close and greater than $\Theta\left(n\log\left(\min\left(k, \frac{n}{\log n}\right)\right)\right)$. The per-symbol redundancy thus goes to infinity as the alphabet size k increases. On the other hand, it was shown in [2] that the redundancies are upper bounded by $\Theta\left(n^{1/3}\right)$ for all $k \leq n$ when there is no constraint, i.e., diminishing per-symbol redundancy can be achieved. The discrepancy results from the fact that pattern probabilities is determined by its profile, but under $\mathcal{C}_1$, compressors do not know this information. Therefore, to compress only knowing the number of distinct integers is not efficient.

5) Universal compression of patterns under constraint $\mathcal{C}_2$

In this section, we consider another constraint $\mathcal{C}_2$ and present lower bounds on the worst and average case redundancies for encoding patterns generated by iid sources over an alphabet of size k. Notice that sequential compressors are often more favorable and they perform encoding one symbol at a time. The encoding of i-th symbol is determined by the content of the previous $i-1$ symbols that have appeared. Therefore, we extend $\mathcal{C}_1$ in a natural sense. We consider constraint $\mathcal{C}_2$ that compressors only know how many distinct index integers are there in the first i symbols, for all i. In other words, for two patterns, if their length-i prefixes have the same number of distinct integers for all i, then they should be assigned with the same probability. It is clear that $\mathcal{C}_1$ is more restrictive than $\mathcal{C}_2$ and any compressor that satisfies $\mathcal{C}_2$ must also satisfies $\mathcal{C}_1$. We will show later $\mathcal{C}_2$ is equivalent of encoding by innovation vectors. Note that although $\mathcal{C}_2$ is motivated by sequential compression algorithms, compressors that satisfy $\mathcal{C}_2$ are not necessarily sequential.

The number of distinct integers is determined by the occurrences of new symbols. Therefore, if the corresponding prefixes of two patterns have the same number of distinct integers, then the innovation vector of the two patterns must be the same. The partition of $\Psi^n_{\leq k}$ defined by $\mathcal{C}_2$ is thus

$$\Psi^n_{\leq k} = \cup_{\boldsymbol{\lambda}\in\Lambda^n_{\leq k}} \Psi^n(\boldsymbol{\lambda}), \tag{314}$$

where $\Psi^n(\boldsymbol{\lambda}) = \{\boldsymbol{\psi} : \Lambda(\boldsymbol{\psi}) = \boldsymbol{\lambda}\}$. The set of allowed compressors is

$$\mathcal{Q}_2 = \{q : q(\boldsymbol{\psi}_1) = q(\boldsymbol{\psi}_2) \text{ if } \Lambda(\boldsymbol{\psi}_1) = \Lambda(\boldsymbol{\psi}_2)\}. \tag{315}$$

Theorem 37. *The worst case redundancy of $\mathcal{I}^{n,k}_{\Psi}$ with respect to $\mathcal{Q}_2$ is the same as that with respect to $\mathcal{Q}_1$, i.e.,*

$$\hat{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_2\right) = \hat{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_1\right) \tag{316}$$

Proof: The proof simply follows from the fact that for each $\boldsymbol{\lambda} \in \Lambda^n_m$, there exists a pattern $\boldsymbol{\psi}^\circ$ that $\Lambda(\boldsymbol{\psi}^\circ) = \boldsymbol{\lambda}$ and $\hat{p}_\Psi(\boldsymbol{\psi}^\circ) = \sup_{\boldsymbol{\psi}'\in\Psi^n_m} \hat{p}_\Psi(\boldsymbol{\psi}')$. ■

The preceding theorem states that although $\mathcal{C}_2$ allows compressors to acquire more information compared with $\mathcal{C}_1$, the worst case redundancy does not decrease. This also is as expected since the compressors do not have the information about profiles.

Theorem 38. *Fix arbitrarily small real numbers $\epsilon, \delta > 0$. As $n \to \infty$, $\bar{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_2\right)$ is greater than*

$$\begin{cases} \left((1-\delta)n\log k - \frac{\log e}{\delta}k(\ln n)^2\right)(1+o(1)), & \textit{for } k < \left(\frac{n}{\ln n}\right)^{1-\epsilon}, \\ (1-\delta-\epsilon)n(\log n - \log\log n)(1+o(1)), & \textit{for } k \geq \left(\frac{n}{\ln n}\right)^{1-\epsilon}. \end{cases} \tag{317}$$

Proof: Fix some small $\epsilon, \delta > 0$. Let $T = \min\left(k, \left\lfloor\left(\frac{n}{\ln n}\right)^{1-\epsilon}\right\rfloor\right)$ and let the vector $\boldsymbol{\theta}_T = (\theta_1, \theta_2, \ldots, \theta_k) \in \Theta_k$ be

$$\theta_t = \begin{cases} 1-(T-1)\frac{\ln n}{\delta n}, & t = 1, \\ \frac{\ln n}{\delta n}, & t = 2, 3, \ldots, T, \\ 0, & \text{otherwise.} \end{cases} \tag{318}$$

Similar to the proof of Theorem 36, we can write

$$\bar{R}\left(\mathcal{I}^{n,k}_{\Psi}, \mathcal{Q}_2\right) \geq \max\left(\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}), \bar{R}\left(\tilde{\mathcal{I}}^{n,k}_{\Psi}\right)\right) \tag{319}$$

$$\geq \sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}) \geq D(p_{\boldsymbol{\theta}_T}||\bar{p}_{\boldsymbol{\theta}_T}) \tag{320}$$

$$= -H_{\boldsymbol{\theta}_T}(\boldsymbol{\psi}) + \sum_{\boldsymbol{\lambda}\in\Lambda^n_{\leq k}} p_{\boldsymbol{\theta}_T}(\boldsymbol{\lambda}) \log|\Psi^n(\boldsymbol{\lambda})|. \tag{321}$$

The pattern entropy is at most the sequence entropy, i.e.,

$$
\begin{aligned}
H_{\boldsymbol{\theta}_T}(\boldsymbol{\psi}) &\le H_{\boldsymbol{\theta}_T}(x^n) && (322)\\
&= n\Bigg((T-1)\frac{\ln n}{\delta n}\log\frac{\delta n}{\ln n} && (323)\\
&\quad + \left(1-(T-1)\frac{\ln n}{\delta n}\right)\log\frac{1}{1-(T-1)\frac{\ln n}{\delta n}}\Bigg) && (324)\\
&< T\frac{\ln n}{\delta}\log\frac{e\delta n}{\ln n}. && (325)
\end{aligned}
$$

For the second term in (321), we show that in the original sequence, the T symbols with positive probabilities will all appear with high probability in the first δn positions. The probability that any symbol does not appear is less than or equal to

$$
\left(1-\frac{\ln n}{\delta n}\right)^{\delta n} \le \frac{1}{n}. \tag{326}
$$

Therefore, by the union bound, the probability that all T symbols appear is greater than or equal to

$$
1-\frac{T}{n} \ge 1-\left(\frac{n}{\ln n}\right)^{-\epsilon}\frac{1}{\ln n}. \tag{327}
$$

If there are T distinct symbols in the first δn positions, then the innovation vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_k)$ must satisfy $\lambda_T \le \delta n$. Moreover, the number of patterns whose innovation vector satisfies this condition is greater than or equal to

$$
T^{n-\delta n} = T^{(1-\delta)n}. \tag{328}
$$

It follows that $\sum_{\boldsymbol{\lambda}\in\Lambda^n_{\le k}} p_{\boldsymbol{\theta}_T}(\boldsymbol{\lambda})\log|\Psi^n(\boldsymbol{\lambda})|$ is greater than or equal to

$$
\begin{aligned}
&\left(1-\left(\frac{n}{\ln n}\right)^{-\epsilon}\frac{1}{\ln n}\right)\log\left(T^{(1-\delta)n}\right) && (329)\\
&= (1-\delta)n(\log T)(1+o(1)). && (330)
\end{aligned}
$$

Summing up (325) and (330) gives

$$
\begin{aligned}
\sup_{\boldsymbol{\theta}\in\Theta_k} D(p_{\boldsymbol{\theta}}||\bar{p}_{\boldsymbol{\theta}}) &> (1-\delta)n(\log T)(1+o(1)) - T\frac{\ln n}{\delta}\log\frac{e\delta n}{\ln n} && (331)\\
&\ge \begin{cases} \left((1-\delta)n\log k - \frac{\log e}{\delta}k(\ln n)^2\right)(1+o(1)), & \text{for } k < \left(\frac{n}{\ln n}\right)^{1-\epsilon},\\ (1-\delta-\epsilon)n(\log n - \log\log n)(1+o(1)), & \text{for } k \ge \left(\frac{n}{\ln n}\right)^{1-\epsilon}. \end{cases} && (332)
\end{aligned}
$$

■

6) A Low-complexity sequential compressor

In this section, we consider the data deduplication algorithms in [77] which encode patterns as follows. For a pattern $\boldsymbol{\psi} = \iota_1\iota_2\cdots\iota_n$, the compressor $Q_D \in \mathcal{Q}_2$ assigns probability sequentially as

$$
Q_D(\boldsymbol{\psi}) = \prod_{i=1}^{n} Q_D\left(\iota_i|\iota_1^{i-1}\right), \tag{333}
$$

where

$$
Q_D\left(\iota_i|\iota_1^{i-1}\right) = \begin{cases} \frac{1}{|M_{i-1}|}\cdot\frac{1}{2} & \text{if } \iota_i \in M_{i-1},\\ \frac{1}{2} & \text{if } \iota_i \notin M_{i-1}, \end{cases} \tag{334}
$$

where M_{i-1} is the set of all index integers in ι_1^{i-1}, i.e., $M_{i-1} = \{\iota_1, \iota_2, \dots, \iota_{i-1}\}$. It is clear that $\mathcal{Q}_D$ satisfies constraint 2 (and thus also constraint 1).

Lemma 39. *For any length-n pattern $\boldsymbol{\psi}$ with m distinct index integers, the maximum probability assigned by any iid source* *satisfies*

$$\hat{p}_\Psi(\boldsymbol{\psi}) \le \sqrt{\frac{2\pi m(n-m+1)}{n}}\,\frac{(n-m+1)^{n-m+1}m^m}{n^n}. \tag{335}$$

Proof: For a pattern $\boldsymbol{\psi}$ with profile $\Phi(\boldsymbol{\psi}) = (\varphi_1, \varphi_2, \ldots, \varphi_n)$, it was shown in [82] that the maximum probability assigned by any iid distribution satisfies

$$\hat{p}_\Psi\big(\bar{\boldsymbol{\psi}}\big) \le \frac{\prod_{\mu=1}^n (\mu!)^{\varphi_\mu} \cdot \varphi_\mu!}{n!}. \tag{336}$$

Note that if $\boldsymbol{\psi}$ contains exactly m distinct index integers, then $\sum_{\mu=1}^n \varphi_\mu = m$, and

$$\prod_{\mu=1}^n (\mu!)^{\varphi_\mu} = \prod_{i=1}^m n_i, \tag{337}$$

where n_i denotes the number of times integer i appears in $\boldsymbol{\psi}$. It follows that

$$\begin{aligned}\frac{\prod_{\mu=1}^n (\mu!)^{\varphi_\mu} \cdot \varphi_\mu!}{n!} &\le \frac{(n-m+1)!m!}{n!} && (338)\\ &\le \frac{\sqrt{2\pi m(n-m+1)}(n-m+1)^{n-m+1}m^m}{\sqrt{n}n^n}, && (339)\end{aligned}$$

where the second inequality follows from that if $a_1 + a_2 + \cdots + a_s = t$, then

$$\prod_{i=1}^s a_i! \le \begin{cases} t! \quad \text{if } 0 \le a_i \le t, \\ (t-s+1)! \quad \text{if } 1 \le a_i \le t, \end{cases} \tag{340}$$

and the last inequality follows from Feller's bounds on Stirling's approximation [37] that for any $m \ge 1$,

$$m! \le \sqrt{2\pi m}\Big(\frac{m}{e}\Big)^m e^{\frac{1}{12m}}. \tag{341}$$

■

Theorem 40. *Let $n \to \infty$. The compressor Q_D satisfies*

$$\hat{R}\Big(\mathcal{I}_\Psi^{n,k}, Q_D\Big) \le \begin{cases} \Big(n\log k - k\log n + n + \frac{(\log e)k^2}{n-k}\Big)(1+o(1)), & \textit{for } k < \frac{n}{\log n}, \\ \Big(n\log\frac{n}{\log n} + \frac{(\log e)n}{(\log n)^2}\Big)(1+o(1)), & \textit{for } k \ge \frac{n}{\log n}. \end{cases} \tag{342}$$

Proof: Let $\boldsymbol{\psi}$ be a pattern in Ψ_m^n with profile

$$\Phi(\boldsymbol{\psi}) = (\varphi_1, \varphi_2, \ldots, \varphi_n) \tag{343}$$

and innovation vector

$$\Lambda(\boldsymbol{\psi}) = (\lambda_1, \lambda_2, \ldots, \lambda_m). \tag{344}$$

Let $\nu_{m+1} = n+1$. The compressor Q_D assigns $\boldsymbol{\psi}$ with probability

$$\begin{aligned} Q_D(\boldsymbol{\psi}) &= \prod_{i=1}^n Q_D\big(\iota_i|\iota_1^{i-1}\big) && (345)\\ &= \Big(\frac{1}{2}\Big)^n \prod_{i=1}^m \Big(\frac{1}{i}\Big)^{\lambda_{i+1}-\lambda_i-1} && (346)\\ &\ge \Big(\frac{1}{2}\Big)^n \Big(\frac{1}{m}\Big)^{n-m}, && (347)\end{aligned}$$

where the equality holds when $\lambda_i = i$ for all $i = 1, 2, \ldots, m$.

By Lemma 39, for any pattern $\boldsymbol{\psi}$ with m distinct index integers,

$$\log\frac{1}{Q_D(\boldsymbol{\psi})}-\log\frac{1}{\hat{p}_\Psi(\boldsymbol{\psi})}\leq\log\bigl(2^n m^{n-m}\bigr) \tag{348}$$

$$+\log\left(\sqrt{\frac{2\pi m(n-m+1)}{n}}\frac{(n-m+1)^{n-m+1}m^m}{n^n}\right) \tag{349}$$

$$=n\log m-m\log n+n\log\left(1-\frac{m-1}{n}\right) \tag{350}$$

$$-m\log\left(1-\frac{m-1}{n}\right)+n+\frac{1}{2}\log\left(\frac{2\pi m}{n}\right) \tag{351}$$

$$+\frac{3}{2}\log(n-m+1) \tag{352}$$

$$\leq n\log m-m\log n+n+(\log e)\left(\frac{m(m-1)}{n-m+1}-m+1\right) \tag{353}$$

$$+O(\log n). \tag{354}$$

Therefore, the worst case redundancy for encoding patterns generated by iid sources using Q_D equals

$$\max_{\boldsymbol{\psi}\in\Psi^n_{\leq k}}\left(\log\frac{1}{Q_D(\boldsymbol{\psi})}-\log\frac{1}{\hat{p}_\Psi(\boldsymbol{\psi})}\right) \tag{355}$$

$$=\max_{m\in[k]}\max_{\boldsymbol{\psi}\in\Psi^n_m}\left(\log\frac{1}{Q_D(\boldsymbol{\psi})}-\log\frac{1}{\hat{p}_\Psi(\boldsymbol{\psi})}\right) \tag{356}$$

$$\leq\max_{m\in[k]}\left(n\log m-m\log n+n+(\log e)\left(\frac{m(m-1)}{n-m+1}-m+1\right)+O(\log n)\right) \tag{357}$$

$$=\begin{cases}\left(n\log k-k\log n+n+\frac{(\log e)k^2}{n-k}\right)(1+o(1)), & \text{for } k<\frac{n}{\log n},\\ \left(n\log\frac{n}{\log n}+\frac{(\log e)n}{(\log n)^2}\right)(1+o(1)), & \text{for } k\geq\frac{n}{\log n}.\end{cases} \tag{358}$$

■

G. Experiments

In this section, we present experimental results related to deduplication algorithms. In our experiment, we consider a real-world dataset and several synthetic datasets. The variable-length chunking scheme is used to split the datasets into multiple chunks. After that, we apply several simple encoding schemes to the chunks and compute the size of the datasets after compression. By setting different marker lengths in the chunking process, we are able to control the average length of the chunks and thus study experimentally how chunk lengths affect the compression ratio. Further, we also compare the effectiveness of different encoding schemes for compressing patterns.

1) Datasets

In our experiment, we consider both real-world and synthetic datasets.

For the real-world dataset, we choose the source code of the GNU program bash[6]. It contains the source code of the bash shell from version 1.14.0 to version 5.2, with an uncompressed total size of 940MB. This dataset serves as a representative of backup/primary storage that contains multiple edited versions of the same file.

We also synthesize datasets according to the source models $\mathcal{I}_b(\delta)$. In particular, we pick a set of values $A=2^{10}, B=2^{15}$ and $L=2^{15}$ bytes. The values of A, B, L are chosen based on the observation made in [77] that experiments suggested reasonable numbers for L range from a few kB to a few MB. Further, we choose the values of B such that the whole data file is of size approximately 1GB. Our synthetic dataset can be viewed as a small sample from a larger and more general dataset. It is also suggested by [77] that the number of distinct source symbols A should be somewhere in the range $0.01B$ to B. Therefore, in our experiment, we pick A to be $B/2^5$. In the $I_b(\delta)$ model, we pick the edit probability $\delta\in\{0,10^{-5}\}$ to simulate situations where we have no edit and small edit probability. Given A, B, L and δ, we use the entropy upper bound provided in 3 as an approximate. For both $\delta=0,10^{-5}$, the entropy is approximately 32.8MB. Note that in our analysis, we consider

[6] https://ftp.gnu.org/gnu/bash/

δ to be a constant and let B, L increase. There, entropy is dominated by the term $BLH(\delta)$ determined by the uncertainty from edits. However, in our experiment, the 32.8MB of entropy is almost fully contributed by the A distinct source symbols. It therefore leaves us an interesting open problem that if our assumption is reasonable for real-world datasets. The actual size of the synthetic dataset is approximately 1.34GB.

2) Rabin-based chunking

The files are split according to the “Rabin-fingerprint” [10], [86], with a sliding window of size 64 bytes. In this method, the sliding window will move from the beginning of the file to the end, one byte at a time. For every 64 bytes in the current window, a “Rabin-fingerprint” will be computed, and if it satisfies a certain condition, a chunk breaking point is made. In our experiment, we set the fingerprint to be of length 53 bits. The fingerprint is obtained by modulo the degree-255 polynomial obtained from the sliding window by a fixed degree-53 polynomial. The polynomials are with coefficients in $\mathbb{Z}_2$. A common way of checking for chunk breaking point is to see whether the least significant bits of the fingerprint are all zero. If the input bytes are random, the fingerprints can also roughly be viewed as random. Therefore, if we check for the least a bits of the fingerprint, then every time we move the sliding window, there is a roughly 2^{-a} probability that a chunk break point is found, i.e., the expected length of the chunks is approximately 2^a bytes. Furthermore, we also set upper and lower limits on the chunk lengths to reduce chunk length variance.

Table 1 shows the average length of chunks in the bash dataset and in the synthetic dataset with edit probability $\delta = 0$.

number of check bits	3	4	5	6	7	8	9	10	11	12
synthetic with $\delta = 0$	7.8	15.6	31.2	62.4	124.8	251.0	500.4	997.1	1,999.4	4,019.2
bash dataset	7.6	15.5	28.0	64.5	129.7	251.2	493.2	965.1	1,860.5	3,449.5

Table 1: Average length of chunks (bytes) of the bash and synthetic datasets under different number of check bits. The upper and lower bound on chunk lengths are not specified.

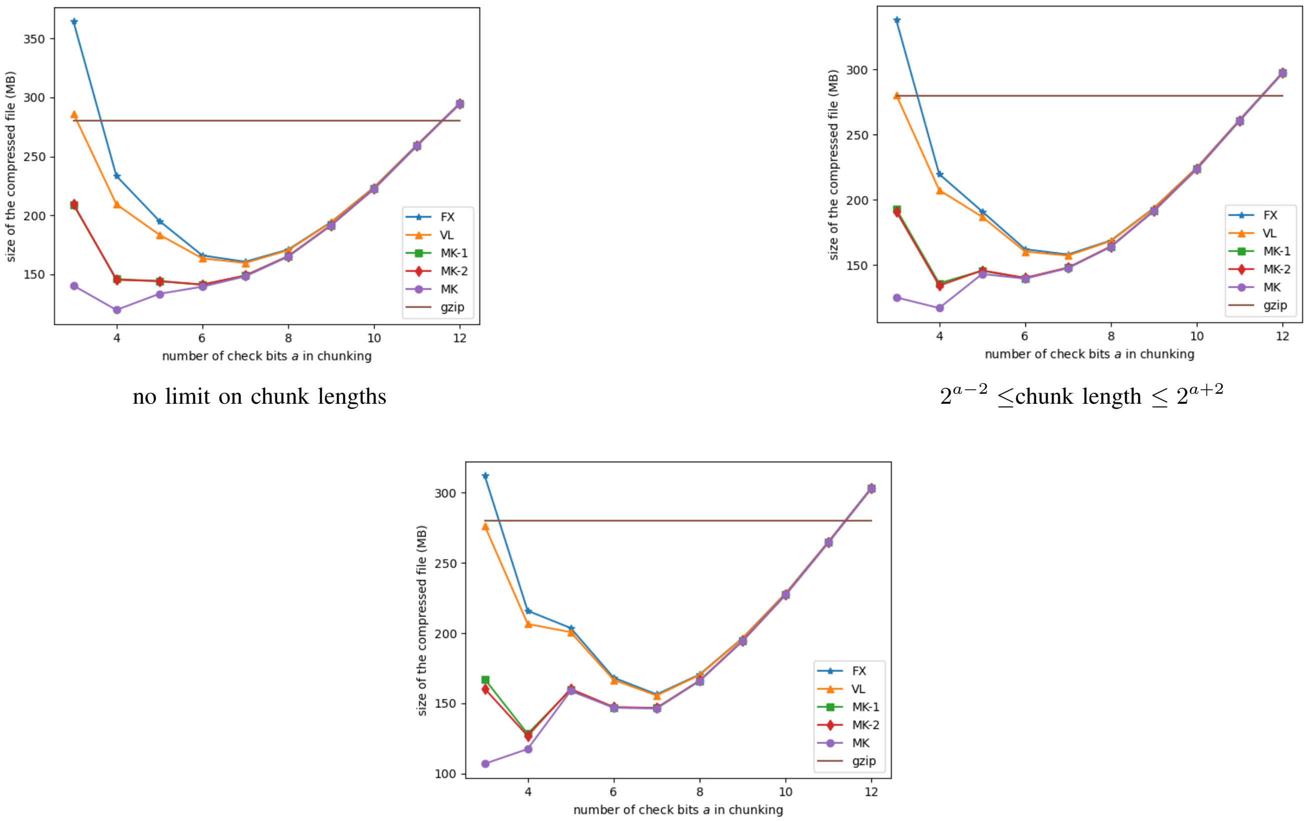
3) Encoding schemes

In our experiment, we consider several types of dictionary-based encoding methods after splitting data files into chunks.

The simple fixed-size pointer encoding is adopted by the fixed-length and variable-length deduplication algorithms provided in Section II-B3. When a chunk has appeared before, it is encoded using a pointer whose length approximately equals log of the total number of distinct chunks processed so far. We denote this encoding scheme by FX. The FX encoding also belongs to the set of pattern compressors that satisfy constraint $\mathcal{C}_2$ in Section II-F. It was shown there that if all iid sources are considered, then the compression is very limited. Given this fact, we also explore a variant of the FX encoding scheme by allowing our dictionary to store not only the chunks that have appeared before, but also the frequency of the appeared chunks. In [80], it is shown that by taking into account the frequencies of symbols/chunks, $o(1)$ per-symbol redundancy can be achieved. We denote this encoding by VL. In the VL scheme, if a chunk z_c has appeared before, it will be replaced by a pointer of length at most $-\log(n_{z_c}/N_{z_c}) + 1$ bits, where n_{z_c} is the number of times z_c has appeared so far, and N_{z_c} is the total number of chunks appeared so far.

Furthermore, in deduplication, some chunks frequently appear together, i.e., if chunk z_1 is followed by chunk z_2, then the next time z_1 appears, z_2 might also follow. This is due to the fact that a repeating content can be partitioned into several chunks. Based on this observation, we also consider encoding chunks based on the context. In particular, we consider an order-1 Markov model, which encodes the chunk z_2 following z_1 in the following way (ignoring indicator bits): i) z_2 is encoded in full if it has not appeared before, ii) z_2 is encoded by a pointer of length at most $-\log(n_{z_2}/N_{z_2}) + 1$ bits if it has appeared before but the chunk pair z_1z_2 has not appeared before, iii) else, z_2 is encoded by a pointer of length at most $-\log(n_{z_1z_2}/n_{z_1}) + 1$ bits, i.e., the number of times z_1 is followed by z_2 divided by the total number of times z_1 appears. We denote this encoding scheme MK.

Note that to apply MK, we need to store the number of times every chunk pair $z_{c_1}z_{c_2}$ appears. Therefore, the memory consumed will be roughly squared in the worst case. To avoid such a large memory overhead, we consider a simplification of the MK encoding scheme. Fix k. For every chunk z_c, we store the first k distinct chunk pairs $z_cz_{c'}$. The encoding of a chunk z_2 following z_1 is given by: i) z_2 is encoded in full if it has not appeared before, ii) z_2 is encoded by a pointer of length at most $-\log(n_{z_2}/N_{z_2}) + 1$ bits if it has appeared before but the chunk pair z_1z_2 is not one of the k chunk pairs associated with z_1, iii) z_2 is encoded by a pointer of length at most $-\log(n_{z_1z_2}/\sum_{i=1}^{k} n_{z_1z_{c'_i}}) + 1$ bits, where chunks $z_{c'_i}$ are the first



no limit on chunk lengths

2^{a-2} $\leq$chunk length $\leq 2^{a+2}$

$2^{a-1} \leq$ chunk length$\leq 2^{a+1}$

Figure 5: Compressed file sizes of the bash dataset vs. number of check bits a for different encoding schemes and chunk length (bytes) constraints.

k chunks that appear right after z_1. We denote this encoding scheme by MK-k. In out experiment, we consider $k = 1$ and $k = 2$. Note that with this simplification, the memory consumption reduces to be linear in the number of distinct chunks.

4) Experiment results and discussion

We first show our experiment on the real-world bash dataset. Figure 5 shows the compressed file sizes of the bash dataset for the five different encoding schemes, different number of check bits a and different upper and lower bounds on chunk lengths. Experiment results for more general settings can be found on Appendix B.

The performance of the deduplication algorithms depends on the number of check bits, i.e., the average length of the chunks. The performance of the FX scheme, which is also the variable-length deduplication algorithm, achieves the optimal when check bits equal to 7. This is consistent with our analysis in Section II-D that the chunk lengths can not be chosen too large or too small. Furthermore, we can observe that by utilizing the frequency information of the chunks, the VL encoding scheme does not show a big advantage over the simple FX encoding scheme. One possible reason could be due to the chunking method, different chunks appear for a similar number of times. The encoding schemes MK, MK-1, MK-2 achieve a much higher compression ratio compared with FX and VL when chunk lengths are small. This is due to the fact that pointers play a more important role when chunk length is small. We thus observe that at the cost of complexity, Markov models provide more robustness in terms of the choice of the expected chunk length. We can also notice that encoding schemes MK-1 and MK-2 have similar performance, which implies that in most cases, chunks are followed by the same chunks.

Figure 6 shows the size of pointers when applying different encoding schemes to the bash dataset. The chunks are limited to lengths between 2^{a-1} and 2^{a+1} bytes. As we can see, as the number of check bits, i.e., the chunk lengths, gets larger, there are more distinct chunks and pointers become less significant. The encoding schemes MK, MK-1 and MK-2 show advantage over the simple encoding schemes FX and VL only for small values of a.

Next, we consider the synthetic dataset, for which the entorpy is known and given in the plots. Figure 7 shows the size

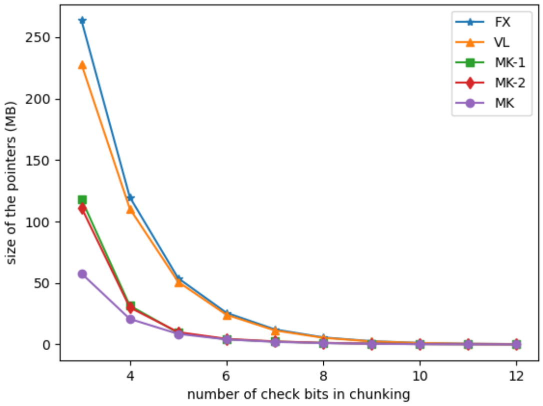


Figure 6: Pointer sizes of the bash dataset vs. number of check bits for different encoding schemes. The chunk lengths are limited to $[2^{a-1}, 2^{a+1}]$ bytes, where a is the number of check bits.

of the compressed file after applying different encoding schemes to the synthetic data with $\delta = 10^{-5}$. Similar to the bash dataset, we can also observe that for encoding schemes FX and VL, the compression ratio first increases and then decreases as the number of check bits increases. This again agrees with our theoretical analysis that neither small chunk lengths nor large chunk lengths can provide good deduplication. Further, for encoding schemes MK, MK-1 and MK-2, both Figures 5 and 7 show that the best compression is achieved with chunk length. One of the reasons could be that after considering the context information, as long as we can identify the first chunk in every repeating block, we can identify all subsequent ones. It can also be observed that MK-2 has almost the same performance as MK, which suggests that on the synthetic dataset, it is enough to store the first two chunks that appear following every chunk. It can also be seen that all deduplication schemes can get close to entropy with the proper choice of the parameters.

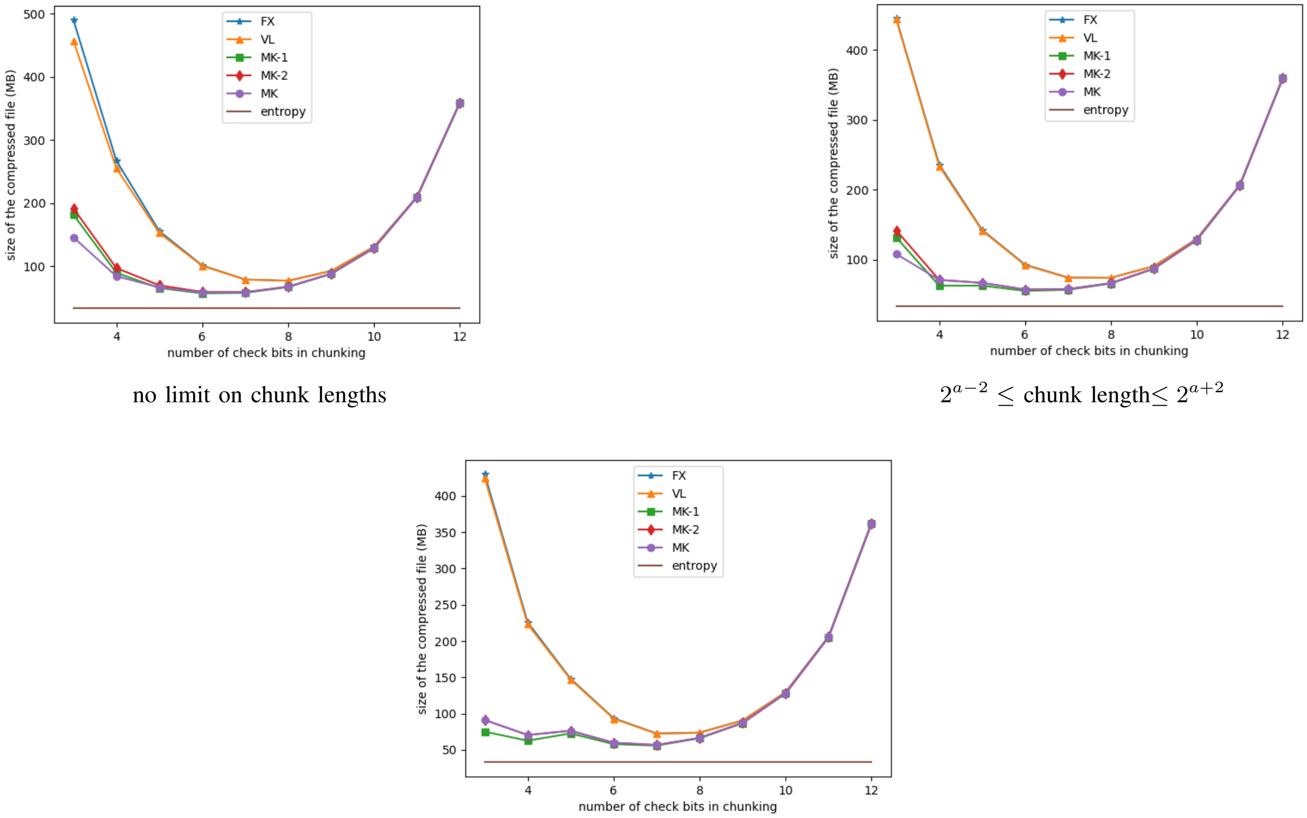
III. Analysis of Genomic Sequence Data via an Evolutionary Model

Due to advances in DNA sequencing, vast amounts of biological sequence data are available nowadays. Developing efficient methods for the analysis and storage of this type of data will benefit from gaining a better mathematical understanding of the structure of these sequences. Biological sequences are formed by genomic mutations, which alter the sequence in each generation to create a new sequence in the next generation. These processes can be viewed as stochastic string editing operations that shape the statistical properties of sequence data.

To gain a better understanding of the evolution of sequences under random mutations, we represented the evolutionary process as a stochastic system in which an arbitrary initial string evolves through random mutation events [61], [63]. In such systems, we studied the evolution of the frequencies of words of length k, i.e., k-mers, as the sequence evolves. The analysis of k-mers has various applications, including identifying functions and evolutionary features [102]. Alignment-free sequence comparison also relies on k-mer frequencies [116]. Their analysis is also of interest because other statistical properties can be computed from k-mer frequencies.

In [61], we studied the asymptotic behavior of k-mer frequencies of the string under mutation, through which we also provided bounds on the entropy of the stochastic evolution system. From an information-theoretic point of view, stochastic sequence generation process through mutation can be viewed as a *source* of information. The entropy determines how well the sequences it produces can be compressed, which is an increasingly important problem given the growth of biological data. Entropy also represents the complexity of sequences generated by the source. Sequence complexity measures, including entropy, have been used to determine the origin and/or the role of DNA sequences [33], [83], [112], for example to classify protein-coding and non-coding regions of a genome.

In previous work, the related problem of finding the combinatorial capacity of duplication systems has been studied. The combinatorial capacity is related to entropy but is defined based on the size of the set of sequences that can be generated by the system, without considering their probabilities. The combinatorial capacity is studied by [36], [48], for duplication systems



no limit on chunk lengths

$2^{a-2} \leq$ chunk length$\leq 2^{a+2}$

$2^{a-1} \leq$ chunk lengths $\leq 2^{a+1}$

Figure 7: Compressed file size of the synthetic dataset with $\delta = 10^{-5}$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.

(without allowing other types of mutations) and by [47] for systems with both tandem duplication and substitution. Compared to combinatorial capacity, entropy provides a more accurate measure of the complexity and compressibility of sequences generated by the system. For duplication systems and duplication/substitution systems, entropy has been studied by [28]. While this work considers a wider range of systems, it only allows duplications involving single symbols. Furthermore, it does not study k-mer frequencies. The stochastic-approximation framework has been used for estimation of model parameters in tandem duplication systems [35]. Estimating the entropy of DNA sequences has been studied in [33], [60], [92]. However these works focus on estimating the entropy from a given sequence, rather than computing the entropy of a stochastic sequence generation system that models evolution. Duplication systems have also been studied in the context of designing error-correcting codes [14], [23], [49], [55].

In [63], we focused on the finite-time behavior of the k-mer frequencies. We studied the first and second moment trajectories of k-mer frequencies, and provided bounds on the waiting time of the k-mers. The waiting time for a given string $\boldsymbol{u}$ in an evolutionary system is the first time index in which $\boldsymbol{u}$ appears as a substring of the evolving sequence. Waiting time problems are of interest since appearances of new patterns in DNA sequences lead to new biological functions and changes in physical attributes [105]. Furthermore, accumulated alterations in certain types of genes, including oncogenes, tumor suppressor genes and genetic instability genes, are known to be responsible for tumorigenesis [109]. Thus, understanding the time scales in which such events take place is of importance in explaining evolutionary trends and the study of diseases such as cancer [26].

Several types of mutations exist in genomic data, including substitution, duplication, insertion, and deletion. Substitution refers to changing a symbol in the sequence, e.g., $\mathsf{ACGTCT} \to \mathsf{ACG\underline{C}CT}$. Duplication mutations refer to the process where a segment of DNA (called the template) is copied and inserted elsewhere in the genome. In the models studied in [61], [63], only tandem deduplication and substitution are included. In tandem duplication, the copy is inserted immediately after the template. For example, from ACGTCT, we may obtain $\mathsf{AC\overline{GT}\underline{GT}CT}$, where the template is overlined and the copy is underlined. Tandem duplication is generally thought to be caused by slipped-strand mispairings [74], where during DNA synthesis, one strand in a DNA duplex becomes misaligned with the other. Tandem duplications and substitutions, along with other mutations, lead to tandem repeats, i.e., stretches of DNA in which the same pattern is repeated many times. Tandem repeats are known to cause important phenomena such as chromosome fragility [106].

From a broader perspective, information theory has natural applications in biology since the processing and transmission of information are ubiquitous in living organisms, from genetic to ecological inheritance mechanisms [108]. Research towards the intersection of information theory and biology can be traced back to the paper "The information content and error rate of living things" [18] in 1949 (just one year after Shannon's seminal paper on information theory). Since then, efforts have been made to address many problems in biology with information-theoretic methods, and have been successful in areas such as predicting the correlation between DNA mutations and disease, identifying protein binding sequences in nucleic acids, and analyzing neural spike trains and higher functionalities of cognitive systems [71]. Recently, due to the symbolism of biological sequences, information theory has found various applications in molecular biology, regarding which [4], [6], [42] serve as excellent surveys. For example, [57] introduced a universal sequence distance based on the information theoretical concept of Kolmogorov complexity and applied it in constructing genome phylogeny; [42] studied the possibility of using mutual information for gene mapping and marker clustering; and [73] studied the the minimum number of reads required for an assembly DNA sequencing algorithm to reconstruct the original sequence. Moreover, two essential areas of information theory, data compression and channel coding, both have direct and practical applications in biology. Compressing biological data has become an inevitable need as the amount of biological sequencing data grows explosively. Many compression algorithms have been designed targeting DNA/RNA sequences [12], [13], [15], [16], [51], [84]. On the other hand, DNA storage is also attracting increased attention due to the longevity and enormous information density of DNA. With challenges arising from the existence of diverse error types in DNA synthesis, replication, and sequencing, many techniques in information theory, especially coding schemes, have been studied and used to enhance the reliability of DNA storage system [43], [50], [54], [56], [87], [104].

In the following, we first give notation and formally define the string evolution system and k-mer frequencies. After that, we present results that are derived in [61] and [63].

A. Preliminaries and Notation

For a positive integer m, let $[m] = \{1, \dots, m\}$. For a finite alphabet Σ, the set of all finite strings over Σ is denoted Σ^*, and the set of all finite non-empty strings is denoted Σ^+. Also, let Σ^k denote the set of k-mers, i.e., length-k strings, over Σ. We let Σ^k be alphabetically ordered, where k-mer $\boldsymbol{u}$ has index $i_{\boldsymbol{u}}$. For instance, let $\Sigma = \{0, 1\}$, then the 2-mers $00, 01, 10, 11$ have orders $i_{00} = 1, i_{01} = 2, i_{10} = 3, i_{11} = 4$. For a string $\boldsymbol{u} \in \Sigma^*$, the elements in $\boldsymbol{u}$ are indexed starting from 1, e.g., $\boldsymbol{u} = u_1 u_2 \cdots u_m$, where $|\boldsymbol{u}| = m$ is the length of $\boldsymbol{u}$. We use $\boldsymbol{u}_{i,j}$ to denote the length-j substring of $\boldsymbol{u}$ starting at u_i. For two positive integers a and b, $\boldsymbol{u}_a^b$ denotes the substring $u_a u_{a+1} \cdots u_b$. Furthermore, the concatenation of two strings $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted by $\boldsymbol{uv}$. For a non-negative integer j, and $\boldsymbol{u} \in \Sigma^*$, $\boldsymbol{u}^j$ is a concatenation of j copies of $\boldsymbol{u}$. Vectors and strings are denoted by boldface letters such as $\boldsymbol{x}$, while scalars and symbols by normal letters such as x. We use $\tau_{\boldsymbol{u}}(m)$ to denote the smallest n such that the sequence $\boldsymbol{s}_n$ contains m occurrences of $\boldsymbol{u}$ and, as shorthand, let $\tau_{\boldsymbol{u}} = \tau_{\boldsymbol{u}}(1)$.

The set of strings at Hamming distance d from $\boldsymbol{w}$ is denoted $\mathcal{B}_d(\boldsymbol{w})$, e.g., $\mathcal{B}_1(00) = \{01, 10\}$. For $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$, we define the indicator function $I(\boldsymbol{u}, \boldsymbol{v})$ as,

$$I(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} 1, & \text{if } \boldsymbol{u} = \boldsymbol{v} \\ 0, & \text{otherwise} \end{cases} .$$

We also provide an informal review of some concepts from probability theory that will be of use. For further detail, we refer readers to [19]. For a sequence of random variables $y_n, n = 0, 1, 2, \dots$, the filtration $\mathcal{F}_n$ associated with the process represents the information provided by $y_0, \dots, y_n$. Formally, $\mathcal{F}_n$ is the sigma-algebra $\sigma(y_0, \dots, y_n)$. The process y_n is a martingale if $\mathbb{E}[y_{n+1} | \mathcal{F}_n] = y_n$. Intuitively, this says that given knowledge of what has happened so far, the expected value of y in the future is equal to its current value. The process y_n is called a martingale difference sequence if $\mathbb{E}[y_{n+1} | \mathcal{F}_n] = 0$. Moreover, we introduce two important results about martingales in the following, Doob's convergence theorem and the Hoeffding-Azuma inequality. Doob's martingale convergence theorem states that if a martingale y_n satisfies $\sup_n \mathbb{E}[|y_n|] < \infty$, then almost surely $y_\infty = \lim_n y_n$ exists and is finite in expectation. The Hoeffding-Azuma inequality states that for a martingale y_n, if $|y_n - y_{n-1}| \leq c_n$ almost surely, then for all positive integers N and all positive reals λ,

$$\Pr(|y_n - y_0| \geq \lambda) \leq 2\exp\left(\frac{-\lambda^2}{2\sum_{n=1}^{N} c_n^2}\right).$$

B. Stochastic String System

A stochastic string system is composed of an initial string $\boldsymbol{s}_0$ and a discrete-time process where in each step a random string edit operation, or 'mutation', is applied to $\boldsymbol{s}_n$, resulting in $\boldsymbol{s}_{n+1}$. To avoid the complications arising from boundaries, we assume the strings $\boldsymbol{s}_n$ are circular, with a given origin and direction. Let the length of $\boldsymbol{s}_n$ be denoted by L_n and let $\ell_n = L_n - L_{n-1}$. For a string $\boldsymbol{u} \in \Sigma^*$, denote the number of appearances of $\boldsymbol{u}$ in $\boldsymbol{s}_n$ as $\mu_n^{\boldsymbol{u}}$, and its frequency as $x_n^{\boldsymbol{u}}$, where $x_n^{\boldsymbol{u}} = \mu_n^{\boldsymbol{u}}/L_n$. For example, if $\boldsymbol{s}_n = \mathsf{ACGAC}$, then $\mu_n^{\mathsf{AC}} = 2, x_n^{\mathsf{AC}} = \frac{2}{5}$. Furthermore, we define $\boldsymbol{\mu}_n(k) = (\mu_n^{\boldsymbol{u}})_{\boldsymbol{u}\in\Sigma^k}$, and $\boldsymbol{x}_n(k) = (x_n^{\boldsymbol{u}})_{\boldsymbol{u}\in\Sigma^k}$, with a lexicographic ordering over Σ^k. We will omit k and directly write $\boldsymbol{\mu}_n$ and $\boldsymbol{x}_n$ when there is no ambiguity. Thus $\boldsymbol{\mu}_n$ is a vector representing the number of appearances of k-mers in the string $\boldsymbol{s}$ at time n and $\boldsymbol{x}_n$ is the normalized version of $\boldsymbol{\mu}_n$.

In [61], we considered the **tandem duplication and substitution** (**TDS**) system, i.e., in each step, the possible mutations are limited to tandem duplications of different lengths or a substitution. In a tandem duplication, a randomly chosen substring of the sequence is duplicated and inserted in tandem. Formally, for $\ell > 0$, the tandem duplication $\mathcal{T}_\ell : \Sigma^* \to \Sigma^*$ is defined as

$$\mathcal{T}_\ell(\boldsymbol{w}) = \boldsymbol{uaav}, \qquad \text{for all } \boldsymbol{w} \in \Sigma^*, \tag{359}$$

where $\boldsymbol{a}$ is a substring of $\boldsymbol{w}$ of length ℓ chosen uniformly at random and $\boldsymbol{w} = \boldsymbol{uav}$. We assign $\mathcal{T}_\ell$ with probability q_ℓ. In a substitution, a position is chosen at random and the symbol in that position is changed to one of the other symbols. We use $\mathcal{T}_0$ to denote the substitution, defined as

$$\mathcal{T}_0(\boldsymbol{w}) = \boldsymbol{u}a'\boldsymbol{v}, \qquad \text{for all } \boldsymbol{w} \in \Sigma^*, \tag{360}$$

where a' is uniformly chosen at random from $\Sigma \setminus a$ and a is a symbol in $\boldsymbol{w}$ chosen also uniformly at random with $\boldsymbol{w} = \boldsymbol{u}a\boldsymbol{v}$. We denote the probability of substitution with q_0. We assume there exists M such that $q_\ell = 0$ for all $\ell \geq M$. Hence, we have $\sum_{\ell=0}^{M-1} q_\ell = 1$. Therefore in TDS systems, the length increment ℓ_n is random, specifically, ℓ_n equals k with probability q_k.

In [63], we considered the **noisy tandem duplication** (**NTD**) system. Noisy tandem duplication is a variant of tandem duplication and substitution, where a randomly chosen substring of the sequence is duplicated with substitution errors and the approximate copy is inserted in tandem. For integers $d \geq 0$ and $\ell \geq 1$, the *noisy duplication* $\mathcal{N}_\ell^d : \Sigma^* \to \Sigma^*$ is defined as

$$\mathcal{N}_\ell^d(\boldsymbol{w}) = \boldsymbol{uaa'v}, \qquad \text{for all } \boldsymbol{w} \in \Sigma^*,$$

where $\boldsymbol{a}$ is a substring of $\boldsymbol{w}$ of length ℓ chosen uniformly at random, $\boldsymbol{u}$ and $\boldsymbol{v}$ are strings such that $\boldsymbol{w} = \boldsymbol{uav}$, and $\boldsymbol{a'} \in \mathcal{B}_d(\boldsymbol{a})$, chosen uniformly at random. In an NTD system with duplication length ℓ, the set of permitted mutations is $\mathcal{M} = \{\mathcal{T}_\ell^d : 0 \leq d \leq \ell\}$, where $\ell \in \mathbb{Z}_{>0}$. In step n, $\mathcal{T}_\ell^d$ occurs with probability p_d, independently of other steps. Hence we have $\sum_{d=0}^{\ell} p_d = 1$. In NTD systems, the length increment ℓ_n in each step is fixed and equals ℓ.

C. Stochastic Approximation for Duplication Systems

In this section, we present an overview of the application of stochastic approximation in the analysis of duplication systems. By using stochastic approximation, our goal is to study how the k-mer frequencies vector $\boldsymbol{x}_n$ changes with n by finding a differential equation whose solution approximates $\boldsymbol{x}_n$.

1) Preliminaries

We start by providing the definitions used in this section. For any positive integer d, a subset of $\mathbb{R}^d$ is said to be *closed* if it contains its boundary, and is said to be *compact* if it is both closed and bounded. Moreover, a subset of $\mathbb{R}^d$ is *connected* if it is not a union of two nonempty separated sets [91]. A set A is an *invariant* set of an ODE $d\boldsymbol{z}_t/dt = \boldsymbol{f}(\boldsymbol{z}_t)$ if it is closed and $\boldsymbol{z}_{t'} \in A$ for some $t' \in \mathbb{R}$ implies that $\boldsymbol{z}_t \in A$ for all $t \in \mathbb{R}$. The invariant set A is *internally chain transitive* with respect to the ODE $d\boldsymbol{z}_t/dt = \boldsymbol{f}(\boldsymbol{z}_t)$, provided that for every $\boldsymbol{y}, \boldsymbol{y'} \in A$ and positive reals T and ϵ, there exist $N \geq 1$ and a sequence

$\boldsymbol{y}_0, \dots, \boldsymbol{y}_N$ with $\boldsymbol{y}_i \in A$, $\boldsymbol{y}_0 = \boldsymbol{y}$, and $\boldsymbol{y}_N = \boldsymbol{y}'$ such that for $0 \le i < n$, if $\boldsymbol{z}_0 = \boldsymbol{y}_i$, then for some $t \ge T$, $\boldsymbol{z}_t$ is in the ϵ-neighborhood of $\boldsymbol{y}_{i+1}$ [8].

We will also make use of the following theorem, which enables studying the behavior of a discrete dynamical system through a system of differential equations.

Theorem 41. *(Stochastic Approximation Theorem [8, Theorem 2].) Let $\{\boldsymbol{z}_n, n \ge 0\}$ be a bounded discrete stochastic process in $\mathbb{R}^d$ with*

$$\boldsymbol{z}_{n+1} = \boldsymbol{z}_n + a(n)[\boldsymbol{h}(\boldsymbol{z}_n) + \boldsymbol{M}_{n+1}], \quad n \ge 0,$$

where $\{\boldsymbol{M}_n, n \ge 0\}$ is a bounded martingale difference sequence in $\mathbb{R}^d$ with $\mathbb{E}[\boldsymbol{M}_{n+1}|\boldsymbol{z}_m, \boldsymbol{M}_m, m \le n] = 0$ almost surely, $\boldsymbol{h} : \mathbb{R}^d \to \mathbb{R}^d$ is a Lipschitz map, and $\{a(n), n \ge 0\}$ are positive scalars satisfying $\sum_n a(n) = \infty, \sum_n a(n)^2 < \infty$. Then $\{\boldsymbol{z}_n, n \ge 0\}$ converges almost surely to a compact connected internally chain transitive invariant set of the ODE

$$\dot{\boldsymbol{z}}_t = \boldsymbol{h}(\boldsymbol{z}_t), \quad t \ge 0.$$

Note the dual use of the symbol $\boldsymbol{z}$; the meaning is however clear from the subscript.

2) Stochastic Approximation in Duplication Systems

We present a set of conditions that will allow us to adapt duplication systems to the stochastic approximation framework, described in Theorem 41. Let $\mathbb{E}_\ell[\ \cdot\ ]$ denote the expected value conditioned on the fact that the length of the duplicated substring is ℓ and let $\boldsymbol{\delta}_\ell = \mathbb{E}_\ell\big[\boldsymbol{\mu}_{n+1}|\mathcal{F}_n\big] - \boldsymbol{\mu}_n$. In the case of substitution, we let $\ell = 0$. We consider the following conditions.

(A1) There exists $M \in \mathbb{N}$ such that $q_i = 0$ for $i \ge M$.

(A2) $\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n$, and thus $\boldsymbol{\delta}_\ell$, are bounded.

(A3) $\boldsymbol{x}_n$ is bounded.

(A4) For each ℓ, $\boldsymbol{\delta}_\ell$ is a function of $\boldsymbol{x}_n$ only, so we can write $\boldsymbol{\delta}_\ell = \boldsymbol{\delta}_\ell(\boldsymbol{x}_n)$.

(A5) The function $\boldsymbol{\delta}_\ell(\boldsymbol{x}_n)$ is Lipschitz.

(A1) holds by assumption. From this follows (A2) since for each k-mer, a mutation can create or eliminate a bounded number of occurrences. Additionally, (A3) holds because each element of $\boldsymbol{x}_n$ is between 0 and 1. The correctness of (A4) and (A5) will be shown for each system.

To understand how $\boldsymbol{x}_n$ varies, our starting point is its difference sequence $\boldsymbol{x}_{n+1} - \boldsymbol{x}_n$. We note that

$$\boldsymbol{x}_{n+1} - \boldsymbol{x}_n = \mathbb{E}[\boldsymbol{x}_{n+1} - \boldsymbol{x}_n|\mathcal{F}_n] + (\boldsymbol{x}_{n+1} - \mathbb{E}[\boldsymbol{x}_{n+1}|\mathcal{F}_n]).$$

For the first term of the right side of (III-C2), we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_{n+1} - \boldsymbol{x}_n|\mathcal{F}_n] &= \sum_{\ell=0}^{M-1} q_\ell(\mathbb{E}_\ell[\boldsymbol{x}_{n+1}|\mathcal{F}_n] - \boldsymbol{x}_n) \\
&= \sum_{\ell=0}^{M-1} q_\ell\left(\frac{\boldsymbol{\mu}_n + \boldsymbol{\delta}_\ell(\boldsymbol{x}_n)}{L_n + \ell} - \frac{\boldsymbol{\mu}_n}{L_n}\right) \\
&= \sum_{\ell=0}^{M-1} q_\ell \frac{L_n \delta_\ell(\boldsymbol{x}_n) - \ell\boldsymbol{\mu}_n}{L_n(L_n + \ell)} \\
&= \sum_{\ell=0}^{M-1} q_\ell \frac{\delta_\ell(\boldsymbol{x}_n) - \ell\boldsymbol{x}_n}{L_n + \ell} \\
&= \frac{1}{L_n}\sum_{\ell=0}^{M-1} q_\ell \boldsymbol{h}_\ell(\boldsymbol{x}_n)\big(1 + O\big(L_n^{-1}\big)\big) \\
&= \frac{1}{L_n}\boldsymbol{h}(\boldsymbol{x}_n)\big(1 + O\big(L_n^{-1}\big)\big),
\end{aligned} \tag{361}$$

where $\boldsymbol{h}_\ell(\boldsymbol{x}_n) = \boldsymbol{\delta}_\ell(\boldsymbol{x}_n) - \ell\boldsymbol{x}_n$, $\boldsymbol{h}(\boldsymbol{x}_n) = \sum_{\ell=0}^{M-1} q_\ell \boldsymbol{h}_\ell(\boldsymbol{x}_n)$, and where we have used $1/(L_n + \ell) = \big(1 + O\big(L_n^{-1}\big)\big)/L_n$, which follows from the boundedness of ℓ (see (A1)).

Furthermore, for the second term of the right side of (III-C2), we have

$$\begin{aligned}\boldsymbol{x}_{n+1} - \mathbb{E}[\boldsymbol{x}_{n+1}|\mathcal{F}_n] &= \frac{\boldsymbol{\mu}_{n+1}}{L_{n+1}} - \mathbb{E}\left[\frac{\boldsymbol{\mu}_{n+1}}{L_{n+1}}\middle|\mathcal{F}_n\right] \\ &= \frac{1+O\big(L_n^{-1}\big)}{L_n}\big(\boldsymbol{\mu}_{n+1} - \mathbb{E}[\boldsymbol{\mu}_{n+1}|\mathcal{F}_n]\big) \\ &= \frac{1}{L_n}\big(1+O\big(L_n^{-1}\big)\big)M_{n+1}, \end{aligned} \tag{362}$$

where $\boldsymbol{M}_{n+1} = \boldsymbol{\mu}_{n+1} - \mathbb{E}\big[\boldsymbol{\mu}_{n+1}|\mathcal{F}_n\big]$. Note that $\boldsymbol{M}_n$ is a bounded martingale difference sequence.

From (III-C2), (361), and (362), we find

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \frac{1}{L_n}\big(\boldsymbol{h}(\boldsymbol{x}_n) + \boldsymbol{M}_{n+1} + O\big(L_n^{-1}\big)\big),$$

where we have used the fact that $\boldsymbol{h}(\boldsymbol{x}_n)\big(1+O\big(L_n^{-1}\big)\big) = \boldsymbol{h}(\boldsymbol{x}_n) + O\big(L_n^{-1}\big)$. This follows from the boundedness of $\boldsymbol{h}(\boldsymbol{x}_n)$, which in turn follows from the boundedness of $\boldsymbol{\delta}_\ell(\boldsymbol{x}_n)$ for all $0 \le \ell < M$. We note that $\boldsymbol{h}$ determines the overall expected behavior of the system.

In the following, the element of $\boldsymbol{\delta}_\ell(\boldsymbol{x}_n)$ that corresponds to $\boldsymbol{u}$ is denoted by $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}_n)$. More precisely, $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}_n) = \mathbb{E}_\ell[\mu_{n+1}^{\boldsymbol{u}} - \mu_n^{\boldsymbol{u}}|\mathcal{F}_n]$. This notation also extends to $\boldsymbol{h}$.

An additional condition requires $\sum_n 1/|\boldsymbol{s}_n| = \infty$ and $\sum_n 1/|\boldsymbol{s}_n|^2 < \infty$, which can be proven using the Borel-Cantelli lemma [41] if $q_0 < 1$. Given these and our discussion above, the following theorem, which relates the discrete system describing $\boldsymbol{x}_n$ to a continuous system, follows directly from Theorem 41.

Theorem 42. *The vector of k-mer frequencies $\boldsymbol{x}_n$ converges almost surely to a compact connected internally chain transitive invariant set of the ODE $d\boldsymbol{x}_t/dt = \boldsymbol{h}(\boldsymbol{x}_t)$.*

D. Asymptotic Analysis of k-mer Frequencies and Entropy in Tandem Duplication and Substitution Systems

In this subsection, we present the results derived in [61]. The analysis starts with considering the evolution of k-mer frequencies in TDS systems. The k-mer frequency vector $\boldsymbol{s}_n$ is formulated in the form of a recurrence. Stochastic-approximation method was then applied on the recurrence, converting the discrete string system to a corresponding continuous system described by an **ordinary differential equation** (**ODE**). With this approach, it was shown that k-mer frequencies converge to a limit point which is a function of model parameters. These results then provide bounds on the entropy of sequences generated in the TDS systems.

1) Frequencies of 1-mers in the TDS system

Before proceeding to the analysis of k-mer frequencies, we present two results for the evolution of symbol frequencies (1-mers) in the TDS system. These results can be viewed as extensions of results for Pólya urn models [66]. In such models, a random ball is chosen from an urn containing balls of different colors. The chosen ball is returned to the urn, along with a predetermined number of balls of the same color. It is known that, conditioned on the present state, the expected ratio of the balls of each color (equivalent to symbol frequencies) in the future is equal to the present value and therefore by definition is a martingale and converges almost surely. While strings are more complex objects than urns, we describe similar results that are valid for any duplication process in which for each i, all i-substring of $\boldsymbol{s}$ have the same chance of being duplicated.

Theorem 43. *In a TDS system with $q_0 = 0$, the random variables x_n^a, $a \in \Sigma$, are martingales and converge almost surely.*

Proof: Suppose $a \in \Sigma$. We have

$$\begin{aligned}\mathbb{E}\big[x_{n+1}^a|\mathcal{F}_n\big] = \mathbb{E}\left[\frac{\mu_{n+1}^a}{L_{n+1}}\middle|\mathcal{F}_n\right] &= \mathbb{E}\left[\mathbb{E}\left[\frac{\mu_{n+1}^a}{L_{n+1}}\middle|\mathcal{F}_n,\ell\right]\middle|\mathcal{F}_n\right] \\ &= \mathbb{E}\left[\frac{\mu_n^a + \ell x_n^a}{L_n+\ell}\middle|\mathcal{F}_n\right] = x_n^a.\end{aligned}$$

We thus have $\mathbb{E}\big[x_{n+1}^a|\mathcal{F}_n\big] = x_n^a$ and so x_n^a is a martingale. Since it is nonnegative, by the martingale convergence theorem, it converges almost surely. ∎

Remark 44. The above theorem does not in fact require the distribution q to be constant and bounded. Under our assumption that q is so, we can in addition obtain the following result on the probability of x_n^a deviating from its starting value.

Theorem 45. *For all $a \in \Sigma$ and $n \geq 1$ we have*

$$\Pr(|x_n^a - x_0^a| \geq \lambda) \leq 2e^{-\lambda^2 L_0/(2M^2)} .$$

Theorem 45 is proved in Appendix C1. The preceding theorem implies that it is unlikely for the composition of a long DNA sequence to change dramatically through random duplication events of bounded length. Such changes, if observed, are likely the result of context-dependent duplications or other biased mutations. Unfortunately, this simple martingale argument does not extend to $x_n^{\boldsymbol{u}}$ when $|\boldsymbol{u}| > 1$. Therefore, for analyzing such cases, we use the more flexible technique of stochastic approximation as described in the sequel.

Next, we study in detail the behavior of a system that allows tandem duplication and substitution mutations. First, we will determine the limits of the frequencies of k-mers. Then, after presenting a theorem relating the limits to entropy, we find bounds on the entropy of these systems.

Let $U = \Sigma^k$, so $\boldsymbol{\mu}_n$ is the vector of all k-mer occurrences, and $\boldsymbol{x}_n$ is the vector of all k-mer frequencies. From Section III-C we know that we can use the differential equation $d\boldsymbol{x}_t/dt = \boldsymbol{h}(\boldsymbol{x}_t)$ to determine the limit of k-mer frequencies. To find the differential equation, in Theorem 50, we determine $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}_n)$ for ℓ with $q_\ell > 0$ and $\boldsymbol{u} \in U$, where it can be observed that (A.4) and (A.5) hold in our model

In the next subsection, we will give some necessary definitions. We will then prove that $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}_n)$ is a linear function of $\boldsymbol{x}_n$, which leads to a linear first-order differential equation. This linear form facilitates determining the asymptotic behavior of the k-mer frequencies. We will then show that the entropy of stochastic string systems can be related to the capacity of semiconstrained systems defined by the limit set of the k-mer frequencies. Leveraging the simple form of the limits for systems with tandem duplications and substitutions, we will provide bounds on the entropy of these systems.

2) Definitions

The following definitions will be useful for finding $\boldsymbol{\delta}_\ell(\boldsymbol{x}_n)$.

Definition 1. For $\boldsymbol{u} \in \Sigma^*$ and $m \in \mathbb{N}^+$, define $\varphi_m(\boldsymbol{u})$ to be a sequence of length $|\boldsymbol{u}|$ whose i-th element is determined by whether the symbol in position i of $\boldsymbol{u}$ equals the symbol in position $i - m$. More specifically, the i-th element of $\varphi_m(\boldsymbol{u})$ is

$$\varphi_m(\boldsymbol{u})_i = \begin{cases} 0, & m+1 \leq i \leq |\boldsymbol{u}|, u_i = u_{i-m} \\ \mathsf{X}, & \text{otherwise} \end{cases}$$

where X is a dummy variable. Let the lengths of the maximal runs of 0s immediately after the initial X^m and at the end of $\varphi_m(\boldsymbol{u})$ be denoted by $l_m^{\boldsymbol{u}}$ and $r_m^{\boldsymbol{u}}$, respectively.

Note that either of $l_m^{\boldsymbol{u}}$ or $r_m^{\boldsymbol{u}}$ may be equal to 0. If $\varphi_m(\boldsymbol{u}) = \mathsf{X}^m 0^{|\boldsymbol{u}|-m}$, then $l_m^{\boldsymbol{u}} = r_m^{\boldsymbol{u}} = |\boldsymbol{u}| - m$. Otherwise, we have $\varphi_m(\boldsymbol{u}) = \mathsf{X}^m 0^{l_m^{\boldsymbol{u}}} Y 0^{r_m^{\boldsymbol{u}}}$, for some Y that starts and ends with X.

Example 11. For $\Sigma = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$, we have

$$\begin{aligned} \boldsymbol{u} &= \mathsf{ACA\ A\ CC\ ACC\ AA\ CAAC}, \\ \varphi_3(\boldsymbol{u}) &= \mathsf{XXX}\,0\ 0\,\mathsf{X}\,0\ 0\,0\,0\,\mathsf{X}\,0\ 0\,0\,0, \end{aligned}$$

and $l_m^{\boldsymbol{u}} = 2$, and $r_m^{\boldsymbol{u}} = 4$.

Remark 46. A duplication of length m is equivalent to inserting m zeros into $\varphi_m(\boldsymbol{u})$. In the above example, $\boldsymbol{u}$ may come from $\boldsymbol{u}' = \mathsf{ACA}\overline{\mathsf{ACC}}\mathsf{AACAAC}$ after a length 3 tandem duplication with the overlined substring as the template and $\varphi_3(\boldsymbol{u})$ can be viewed as the result of inserting 3 zeros into $\varphi_3(\boldsymbol{u}') = \mathsf{XXX}00\overline{\mathsf{X}0\mathsf{X}}0000$ between the two overlined symbols.

To enable us to succinctly represent the results, we then define several functions. These functions relate $\boldsymbol{u}$ to the frequencies of other substrings that can generate $\boldsymbol{u}$ via appropriate duplication events. For example, consider the sequence $\boldsymbol{u} = \mathsf{ACACAGAG}$, for which $\varphi_2(\boldsymbol{u}) = \mathsf{XX}000\mathsf{X}00$. This sequence can be created through duplications of length 2 from ACAGAG (in two ways) and from ACACAG. These correspond to runs of 0 of length 2 in $\varphi_2(\boldsymbol{u})$.

Definition 2. For a sequence $\boldsymbol{u}$ and positive integers m, z with $m + z \leq |\boldsymbol{u}| + 1$, define

$$D_{z,m}(\boldsymbol{u}) = \boldsymbol{u}_{1,z-1}\boldsymbol{u}_{z+m,|\boldsymbol{u}|+1-z-m},$$

the sequence obtained from $\boldsymbol{u}$ by removing the subsequence $\boldsymbol{u}_{z,m}$, i.e., by removing symbols in positions $z, \ldots, z+m-1$.

Example 12. For $\boldsymbol{u} = \mathsf{ACGTA}, z = 3, m = 2$, we have $\boldsymbol{u}_{3,2} = \mathsf{GT}$ and $D_{3,2}(\mathsf{ACGTA}) = \mathsf{ACA}$.

Definition 3. For a string $\boldsymbol{u}$ and positive integers m, z with $m + z \leq |\boldsymbol{u}| + 1$, define

$$G_m^{\boldsymbol{u}}(\boldsymbol{x}) = \sum_z x^{D_{z,m}(\boldsymbol{u})}, \tag{363}$$

where the sum is over all z that are the indices of the start of (not necessarily maximal) runs of 0s in $\boldsymbol{\varphi}_m(\boldsymbol{u})$, i.e., $(\boldsymbol{\varphi}_m(\boldsymbol{u}))_{z,m} = 0^m$.

Example 13. For $\boldsymbol{u} = \mathsf{GACCACCA}, m = 3$, we have $\boldsymbol{\varphi}_3(\boldsymbol{u}) = \mathsf{XXXX0000}$ and $(\boldsymbol{\varphi}_3(\boldsymbol{u}))_{5,3} = (\boldsymbol{\varphi}_3(\boldsymbol{u}))_{6,3} = 0^3$. Therefore $G_3^{\boldsymbol{u}}(\boldsymbol{x}) = 2x^{\mathsf{GACCA}}$.

There is a slight abuse of notation in the definition of G above (as well as the definitions of F and M below). While the argument of G is $\boldsymbol{x} = (x^{\boldsymbol{v}})_{\boldsymbol{v}\in\mathcal{A}^k}$, on the right side of (363), $x^{\boldsymbol{w}}$ for sequences $\boldsymbol{w}$ with $|\boldsymbol{w}| < k$ may appear. We note however that $x^{\boldsymbol{w}}$ can be obtained from $\boldsymbol{x}$ by summing over the elements of $\boldsymbol{x}$ corresponding to strings that include $\boldsymbol{w}$ as a prefix.

New occurrences of $\boldsymbol{u}$ can also be generated from strings that are not of the form $D_{z,m}(\boldsymbol{u})$. For example, consider the sequence $\boldsymbol{u} = \mathsf{ACGACTG}$, for which $\boldsymbol{\varphi}_3(\boldsymbol{u}) = \mathsf{XXX00XX}$. This sequence can be created through a length-3 tandem duplication from $\overline{\mathsf{CGA}}\mathsf{CTG}$ and $\overline{\mathsf{GAC}}\mathsf{TG}$, where the part that is to be duplicated is overlined. The following definitions will be of use in the analysis of this type of duplication.

Definition 4. For a sequence $\boldsymbol{u}$ and a positive interger m, define

$$F_{m,l}^{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{i=1}^{\min(l_m^{\boldsymbol{u}},m-1)} x^{\boldsymbol{u}_{i+1,|\boldsymbol{u}|-i}},$$

$$F_{m,r}^{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{i=1}^{\min(r_m^{\boldsymbol{u}},m-1)} x^{\boldsymbol{u}_{1,|\boldsymbol{u}|-i}}.$$

In the special case where $\boldsymbol{\varphi}_m(\boldsymbol{u}) = \mathsf{X}^m 0^{|\boldsymbol{u}|-m}$ and $|\boldsymbol{u}| \leq 2m-2$, we will benefit from the following definition.

Definition 5. For a sequence $\boldsymbol{u}$ and a positive integer m s.t. $\boldsymbol{\varphi}_m(\boldsymbol{u}) = \mathsf{X}^m 0^{|\boldsymbol{u}|-m}$ and $|\boldsymbol{u}| \leq 2m-2$, define

$$M_m^{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{b=|\boldsymbol{u}|-m+1}^{m-1} x^{\boldsymbol{u}_{b+1,m-b}\boldsymbol{u}_{1,b}}.$$

We define $M_m^{\boldsymbol{u}}(\boldsymbol{x}) = 0$ if $\boldsymbol{\varphi}_m(\boldsymbol{u}) \neq \mathsf{X}^m 0^{|\boldsymbol{u}|-m}$.

3) Evolution of k-mer Frequencies

We first find $\boldsymbol{\delta}_\ell(\boldsymbol{x}) = (\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}))_{\boldsymbol{u}\in U}$ for $\ell > 0$ (duplication) and then for $\ell = 0$ (substitution). When analyzing $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x})$, we only consider substrings $\boldsymbol{u}$ of length $|\boldsymbol{u}| > \ell$, which simplifies the derivation. The frequencies of shorter substrings can be found by summing over the frequencies of longer substrings.

We first analyze the case in which $\ell > 0$. We present three lemmas and then use them to prove a general form for $\boldsymbol{\delta}_\ell(\boldsymbol{x}), \ell > 0$. Suppose a duplication of length ℓ occurs in $\boldsymbol{s}_n$, resulting in $\boldsymbol{s}_{n+1}$. The number of occurrences of $\boldsymbol{u}$ may change due to the duplication event. To study this change, we consider the k-substrings of $\boldsymbol{s}_n$ that are eliminated (do not exist in $\boldsymbol{s}_{n+1}$) and the k-substrings of $\boldsymbol{s}_{n+1}$ that are new (do not exist in $\boldsymbol{s}_n$). Any new k-substring must intersect with both the template and the copy in $\boldsymbol{s}_{n+1}$. Likewise, an eliminated k-substring must include symbols on both sides of the template in $\boldsymbol{s}_n$, i.e., the template must be a strict substring of the k-substring that includes neither its leftmost symbol nor its rightmost symbol. As an example,

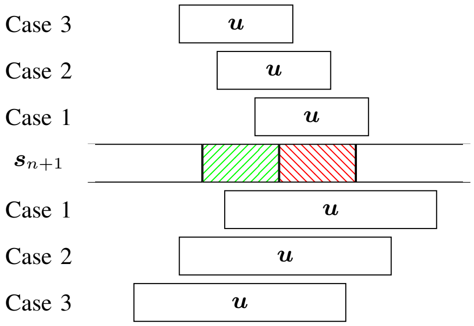


Figure 8: Possible cases for new occurrences of u in s_{n+1}. Cases above and below s_{n+1} correspond to $\ell+1 \leq k < 2\ell$ and $k \geq 2\ell$, respectively. The hatched boxes, from left to right, are the template and the copy.

suppose

$$s_n = v\mathsf{ACGTAGAT}w, \tag{364}$$

$$s_{n+1} = v\mathsf{ACG}\overline{\mathsf{TAG}}\underline{\mathsf{TAG}}\mathsf{AT}w, \tag{365}$$

where $\ell = 3$, the (new) copy is underlined and the template is overlined, and $v, w \in \Sigma^*$. Let $k = 5$, the new 5-substrings are GTAGT, TAGTA, AGTAG, GTAGA and the eliminated substring is GTAGA. Note that here the two GTAGA substrings are counted as different. Formally, let

$$s_n = a_1 \cdots a_i a_{i+1} \cdots a_{i+\ell} a_{i+\ell+1} \cdots a_{|s_n|},$$

$$s_{n+1} = a_1 \cdots a_i a_{i+1} \ldots a_{i+\ell} a_{i+1} \ldots a_{i+\ell} a_{i+\ell+1} \ldots a_{|s_n|},$$

where the substring $a_{i+1} \cdots a_{i+\ell}$ is duplicated. The new k-substrings created in s_{n+1} are

$$y_b = a_{i+\ell+1-b} a_{i+\ell+2-b} \ldots a_{i+\ell} a_{i+1} a_{i+2} \ldots a_{i+k-b},$$

for $1 \leq b \leq k-1$. Note that we have defined y_b such that the first element of the copy, a_{i+1}, is at position $b+1$ in y_b. The k-substrings eliminated from s_n are $a_{i-c+1} \cdots a_{i+k-c}$, for $1 \leq c \leq k-\ell-1$.

For a given u, let Y_b denote the indicator random variable for the event that $y_b = u$, that is, the duplication creates a new occurrence of u in s_{n+1} in which the first symbol of the copy is in position $b+1$. In example denoted by (365), if $u = \mathsf{TAGTA}$, then $y_3 = u$ and thus $Y_3 = 1$.

Furthermore, let W denote the number of occurrences of u that are eliminated. We have

$$\begin{aligned}\delta_\ell^{u}(x) &= \Big(\sum_{b=1}^{k-1} \mathbb{E}_\ell[Y_b|\mathcal{F}_n]\Big) - \mathbb{E}_\ell[W|\mathcal{F}_n]\\ &= \Big(\sum_{b=1}^{k-1} \mathbb{E}_\ell[Y_b|\mathcal{F}_n]\Big) - (k-\ell-1)x^{u}, \end{aligned} \tag{366}$$

where the second equality follows from the fact that each of the $k-\ell-1$ eliminated k-substrings are equal to u with probability x^{u}.

To find δ_ℓ^{u}, it suffices to find $\mathbb{E}_\ell[Y_b|\mathcal{F}_n]$, or equivalently, $\Pr(Y_b = 1|\mathcal{F}_n, \ell)$. We consider different cases based on the value of b, which determines how u overlaps with the template and the copy. These cases are illustrated in Figure 8 and are considered in Lemmas 47–49, whose proofs are given in the Appendix C2.

Lemma 47 (Case 1)**.** *For* $1 \leq b < \min(\ell, k-\ell+1)$,

$$\mathbb{E}_\ell[Y_b|\mathcal{F}_n] = x^{u_{b+1,k-b}} I(u_{1,b}, u_{1+\ell,b}).$$

Lemma 48 (Case 2)**.** *Suppose* $\min(\ell, k-\ell+1) \leq b < \max(k-\ell+1, \ell)$*. If* $k \geq 2\ell$*, then*

$$\mathbb{E}_\ell[Y_b|\mathcal{F}_n] = x^{u_{1,b-\ell} u_{b+1,k-b}} I(u_{b-\ell+1,\ell}, u_{b+1,\ell}),$$

and if $\ell+1 \leq k \leq 2\ell-2$, then

$$\mathbb{E}_\ell[Y_b|\mathcal{F}_n] = x^{\boldsymbol{u}_{b+1,\ell-b}\boldsymbol{u}_{1,b}} I(\boldsymbol{u}_{1,k-\ell}, \boldsymbol{u}_{\ell+1,k-\ell}).$$

Lemma 49 (Case 3)**.** *For $\max(k-\ell+1,\ell) \leq b \leq k-1$,*

$$\mathbb{E}_\ell[Y_b|\mathcal{F}_n] = x^{\boldsymbol{u}_{1,b}} I(\boldsymbol{u}_{b-\ell+1,k-b}, \boldsymbol{u}_{b+1,k-b}).$$

Based on Lemmas 47–49, we then prove the following Theorem. We will use the three lemmas above to break the summation of (366) into three parts and then simplify them to get a generalized expression.

Theorem 50. *For an integer $\ell > 0$ and a string $\boldsymbol{u} = u_1 u_2 \cdots u_k$, if $\ell+1 \leq k < 2\ell$, then*

$$\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) = F_{\ell,l}^{\boldsymbol{u}}(\boldsymbol{x}) + F_{\ell,r}^{\boldsymbol{u}}(\boldsymbol{x}) + M_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - (k-1-\ell)x^{\boldsymbol{u}},$$

and if $k \geq 2\ell$,

$$\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) = F_{\ell,l}^{\boldsymbol{u}}(\boldsymbol{x}) + F_{\ell,r}^{\boldsymbol{u}}(\boldsymbol{x}) + G_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - (k-1-\ell)x^{\boldsymbol{u}}. \tag{367}$$

Theorem 50 is proved in Appendix C3.

In the case of $\ell = 0$, $\boldsymbol{\delta}_\ell(\boldsymbol{x})$ is given by the following theorem.

Theorem 51. *For a string $\boldsymbol{u}$ of length k, we have*

$$\delta_0^{\boldsymbol{u}}(\boldsymbol{x}) = \frac{1}{|\Sigma|-1} \sum_{\boldsymbol{v} \in \mathcal{B}_1(\boldsymbol{u})} x^{\boldsymbol{v}} - kx^{\boldsymbol{u}}. \tag{368}$$

Before proving the theorem, we give an example for $\Sigma = \{1,2,3\}$:

$$\delta_0^{123}(\boldsymbol{x}) = \frac{1}{2}(x^{223} + x^{323} + x^{113} + x^{133} + x^{121} + x^{122}) - 3x^{123}$$

Proof: A new occurrence of $\boldsymbol{u}$ results from an appropriate substitution in some $\boldsymbol{v} \in \mathcal{B}_1(\boldsymbol{u})$, which has probability $x^{\boldsymbol{v}}/(|\Sigma|-1)$. On the other hand, an occurrence of $\boldsymbol{u}$ is eliminated if a substitution occurs in any of its k positions. So the expected number occurrences that vanish is $kx^{\boldsymbol{u}}$. ■

4) ODE and the Limits of Substring Frequencies

Theorems 50 and 51 provide expressions for $\boldsymbol{\delta}_\ell(\boldsymbol{x})$ for $0 \leq \ell \leq M-1$. With these results in hand, we can formulate an ordinary differential equation (ODE) whose limits are the same as those of the substring frequencies of interest, $\boldsymbol{x} = (x^{\boldsymbol{u}})_{\boldsymbol{u} \in \Sigma^k}$, where $k \geq M$.

We first show that $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x})$ can be written as a linear combination of the elements of $\boldsymbol{x}$, i.e., a linear combination of $x^{\boldsymbol{v}}, \boldsymbol{v} \in \Sigma^k$. To see this, note that on the right side in expressions for $\delta_\ell^{\boldsymbol{u}}$ in Theorems 50 and 51, terms of the form $x^{\boldsymbol{w}}$ appear where $|\boldsymbol{w}| \leq k$. We can replace $x^{\boldsymbol{w}}$ with $\sum_{\boldsymbol{v}} x^{\boldsymbol{v}}$, where the summation is over all strings $\boldsymbol{v}$ of length k such that $\boldsymbol{w}$ is a prefix of $\boldsymbol{v}$. For example, consider the alphabet $\{1,2,3\}$ and $k=3$. From Theorem 50, we have

$$\begin{aligned}\delta_2^{121}(\boldsymbol{x}) &= x^{12} + x^{21} \\ &= x^{121} + x^{122} + x^{123} + x^{211} + x^{212} + x^{213}.\end{aligned}$$

For $0 \leq \ell < M$, let A_ℓ be the matrix satisfying $\boldsymbol{\delta}_\ell(\boldsymbol{x}) - \ell\boldsymbol{x} = A_\ell \boldsymbol{x}$. Based on the argument above, such a matrix exists and can be computed from Theorems 50 and 51. Furthermore, let

$$A = \sum_{\ell=0}^{M-1} q_\ell A_\ell. \tag{369}$$

Note that $\boldsymbol{h}_\ell(\boldsymbol{x}) = A_\ell \boldsymbol{x}$ and $\boldsymbol{h}(\boldsymbol{x}) = \sum_\ell q_\ell \boldsymbol{h}_\ell(\boldsymbol{x}) = A\boldsymbol{x}$.

For example, consider $q_0 = \alpha$, $q_1 = 1-\alpha$, $\Sigma = \{0,1\}$, and $\boldsymbol{x} = (x^{00}, x^{01}, x^{10}, x^{11})$. From Theorems 50 and 51, it can be shown that

$$A_0 = \begin{pmatrix} -2 & 1 & 1 & 0 \\ 1 & -2 & 0 & 1 \\ 1 & 0 & -2 & 1 \\ 0 & 1 & 1 & -2 \end{pmatrix}, \qquad A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

and

$$A = \begin{pmatrix} -2\alpha & 1 & \alpha & 0 \\ \alpha & -(1+\alpha) & 0 & \alpha \\ \alpha & 0 & -(1+\alpha) & \alpha \\ 0 & \alpha & 1 & -2\alpha \end{pmatrix}. \tag{370}$$

Theorem 52. *Consider a tandem duplication and substitution system with distribution $q = (q_\ell)_{0\le\ell<M}$ over these mutations, with $q_0 < 1$, and let A be the matrix defined for this system by* (369)*. The frequencies of substrings $\boldsymbol{u}$ of length $k \ge M$, $(x^{\boldsymbol{u}})_{\boldsymbol{u}\in\Sigma^k}$, converge almost surely to the null space of the matrix A.*

Theorem 52 is proved in Appendix C4.

For the matrix A of (370), for $0 < \alpha < 1$, the vector in the null space whose elements sum to 1, and thus the limit of $\boldsymbol{x}_n$, is

$$\frac{1}{2(1+3\alpha)}(\alpha+1, 2\alpha, 2\alpha, \alpha+1)^T. \tag{371}$$

If we let $\alpha = \frac{1}{4}$ as an example, the limit of $\boldsymbol{x}_n$ then is

$$\lim_{n\to\infty}(x_n^{00}, x_n^{01}, x_n^{10}, x_n^{11})^T = (\frac{5}{14}, \frac{1}{7}, \frac{1}{7}, \frac{5}{14})^T. \tag{372}$$

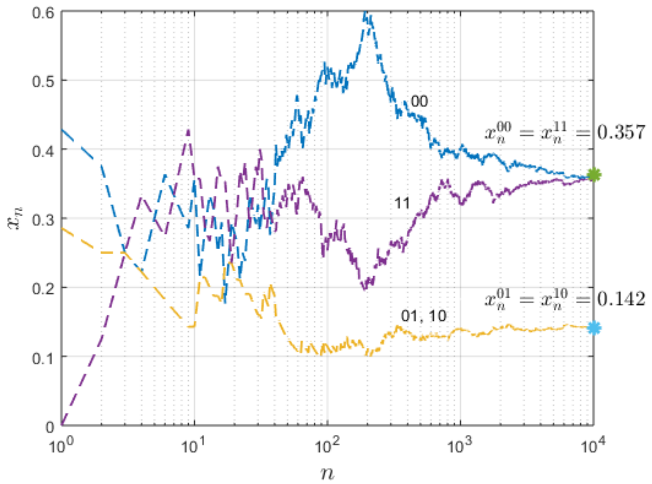


Figure 9: 2-mer frequencies vs the number of mutations in a tandem duplication and substitution system, with $\Sigma = \{0,1\}, \boldsymbol{s}_0 = 0100010, q_0 = \frac{1}{4},$ and $q_1 = \frac{3}{4}$.

Figure 9 shows the result of simulation of the above TDS system, where $\Sigma = \{0,1\}$, $\boldsymbol{s}_0 = 0100010, q_0 = \frac{1}{4}$ and $q_1 = \frac{3}{4}$. As the number n of mutations increases, the frequency vector $\boldsymbol{x}_n$ converges to the analytical result (372). Note that the limits do not depend on the initial sequence $\boldsymbol{s}_0$.

Let us consider the two extreme cases. As $\alpha \to 1$, all four 2-substrings become equally likely, each with probability $1/4$. Note however that our analysis is not applicable to $q_0 = \alpha = 1$ since the condition $\sum_n 1/|\boldsymbol{s}_n|^2 < \infty$ is not satisfied. On the other hand, for a small probability of substitution, $0 < \alpha \ll 1$, almost all 2-substrings are either 00 or 11, as expected. For $\alpha = 0$, the null space is spanned by $\boldsymbol{z}_1 = (1,0,0,0)^T$ and $\boldsymbol{z}_2 = (0,0,0,1)^T$ and the limit set is $\{a\boldsymbol{z}_1 + (1-a)\boldsymbol{z}_2 : 0 \le a \le 1\}$. In this case, the asymptotic behavior of k-mer frequencies will depend on the initial sequence $\boldsymbol{s}_0$.

5) Bounds on Entropy

We now turn to provide upper bounds on the entropy. We first formally define the entropy, and then argue that the entropy is upper bounded by the capacity of an appropriately defined semiconstrained system [29]–[31].

Consider the string $\boldsymbol{s}_n$, obtained from $\boldsymbol{s}_0$ by n rounds of mutations, as described previously. Its expected length is $\mathbb{E}[|\boldsymbol{s}_n|] = |\boldsymbol{s}_0| + n\sum_{\ell=1}^{M-1} \ell q_\ell$. We define the entropy after n rounds as

$$\begin{aligned}\mathcal{H}_n &= \frac{1}{\mathbb{E}[|\boldsymbol{s}_n|]} \cdot H(\boldsymbol{s}_n) \\ &= -\frac{1}{\mathbb{E}[|\boldsymbol{s}_n|]} \sum_{\boldsymbol{w}\in\Sigma^*} \Pr(\boldsymbol{s}_n = \boldsymbol{w}) \log_{|\Sigma|} \Pr(\boldsymbol{s}_n = \boldsymbol{w}),\end{aligned} \tag{373}$$

and the entropy $\mathcal{H}_\infty = \limsup_{n\to\infty} \mathcal{H}_n$. We note that $H(\boldsymbol{s}_n)$ is the usual entropy of $\boldsymbol{s}_n$ (except for the fact that we use base-$|\Sigma|$ logarithms instead of the usual base-2 logarithms).

It is common to define the entropy of DNA sequences based on the limit of block entropies [44], [59], [92]. Specifically, let $h_k = -\sum_{\boldsymbol{u}\in\Sigma^k} p_{\boldsymbol{u}} \log p_{\boldsymbol{u}}$, where $p_{\boldsymbol{u}}$ is the probability of observing $\boldsymbol{u}$. Entropy is then obtained as $h_{k+1} - h_k$ for $k \to \infty$. This definition may lead to misleading results. For example, consider a string system in which $\boldsymbol{s}_n$ is the De Bruijn sequence of order n (which contains all strings of length n precisely once), obtained according to some deterministic algorithm. Based on block entropies, the entropy of the system can be shown to equal $\log|\Sigma|$, while the system is in fact deterministic. The definition in (373) gives the correct entropy, i.e., 0, since there is only one possibility for $\boldsymbol{s}_n$ for each n.

Let us recall some definitions concerning semiconstrained systems (see [30]). Fix k and let $\mathcal{P}(\Sigma^k)$ denote the set of all probability measures on Σ^k. A *semiconstrained system* is defined by $\Gamma_k \subseteq \mathcal{P}(\Sigma^k)$. The set of the admissible words of the semiconstrained system, denoted $\mathcal{B}(\Gamma_k)$, contains exactly all finite words over the alphabet Σ whose k-mer distribution is in Γ_k. Let $\mathcal{B}_n(\Gamma_k) = \mathcal{B}(\Gamma_k) \cap \Sigma^n$. An expansion of Γ_k by $\epsilon > 0$ is defined as

$$\mathbb{B}_\epsilon(\Gamma_k) = \left\{ \boldsymbol{\xi} \in \mathcal{P}(\Sigma^k) \ : \ \inf_{\boldsymbol{\nu}\in\Gamma_k} \|\boldsymbol{\nu} - \boldsymbol{\xi}\|_{\mathrm{TV}} \leq \epsilon \right\},$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the total-variation norm. Thus, $\mathbb{B}_\epsilon(\Gamma_k)$ contains all the measures in Γ_k as well as those which are ϵ-close to some measure in Γ_k. The capacity of Γ_k is then defined as

$$\mathsf{cap}(\Gamma_k) = \lim_{\epsilon\to 0^+} \limsup_{n\to\infty} \frac{1}{n} \log_{|\Sigma|} |\mathcal{B}_n(\mathbb{B}_\epsilon(\Gamma_k))|,$$

which intuitively measures the information per symbol in strings whose k-mer distribution is in (or “almost” in) Γ_k.

Theorem 53. *For the mutation process described above, for $k \in \mathbb{N}^+$, if the vector of the frequencies $\boldsymbol{x}$ of strings of length k converges almost surely to a set Γ_k, then $\mathcal{H}_\infty \leq \mathsf{cap}(\Gamma_k)$.*

Theorem 53 is proved in Appendix C5.

Remark 54. We comment that if $\Gamma_k = \{\boldsymbol{\xi}_k\}$, i.e., Γ_k contains a single shift-invariant measure[7], then $\mathsf{cap}(\Gamma_k)$ has a nice form for all $k \in \mathbb{N}^+$ (see [30], [31]):

$$\mathsf{cap}(\Gamma_k) = -\sum_{a_1\dots a_k\in\Sigma^k} \boldsymbol{\xi}_k^{a_1\dots a_k} \log_{|\Sigma|} \frac{\boldsymbol{\xi}_k^{a_1\dots a_k}}{\bar{\boldsymbol{\xi}}_k^{a_1\dots a_{k-1}}},$$

where $\bar{\boldsymbol{\xi}}_k$ is the marginal of $\boldsymbol{\xi}_k$ on the first $k-1$ coordinates, i.e., $\bar{\boldsymbol{\xi}}_k^{a_1\dots a_{k-1}} = \sum_{b\in\Sigma} \boldsymbol{\xi}_k^{a_1\dots a_{k-1}b}$. Furthermore, for all $k \in \mathbb{N}^+$,

$$\mathsf{cap}(\Gamma_k) \geq \mathsf{cap}(\Gamma_{k+1}),$$

which follows from the fact that $\mathsf{cap}(\Gamma_k)$ can be viewed as the conditional entropy of a symbol given the $k-1$ previous symbols in a stationary process.

Using the preceding remark and Theorem 53, we can find a series of upper bounds on a given system:

$$\mathsf{cap}(\Gamma_1) \geq \mathsf{cap}(\Gamma_2) \geq \cdots \geq \mathsf{cap}(\Gamma_k) \geq \cdots \geq \mathcal{H}_\infty,$$

[7] A shift-invariant measure $\boldsymbol{\xi}_k \in \mathcal{P}(\Sigma^k)$ is a measure that satisfies $\sum_{a\in\Sigma} \boldsymbol{\xi}_k^{a\boldsymbol{w}} = \sum_{a\in\Sigma} \boldsymbol{\xi}_k^{\boldsymbol{w}a}$ for all $\boldsymbol{w} \in \Sigma^{k-1}$. The k-mer distributions of cyclic strings are always shift invariant, and thus a converging sequence of such measures also converges to a shift-invariant measure.

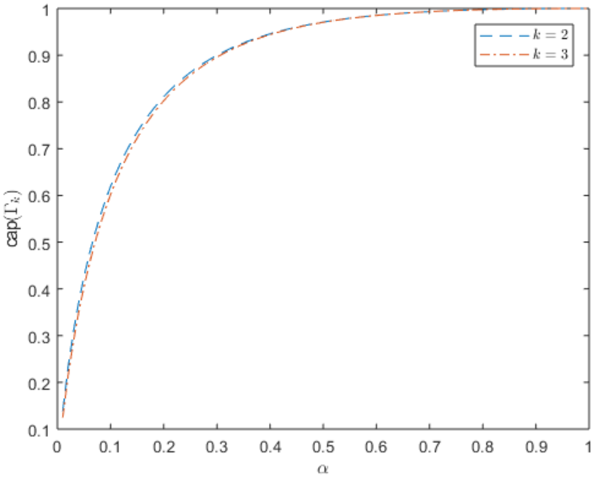


Figure 10: Entropy bound vs the probability of substitution, with $\Sigma = \{0, 1\}$.

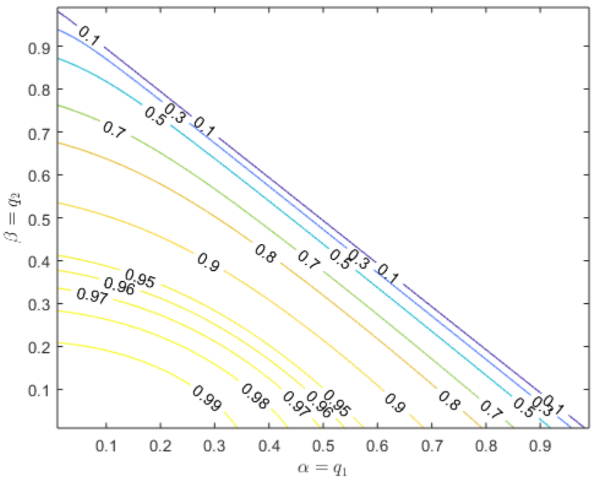


Figure 11: Contour plot of entropy bounds, with $\Sigma = \{0, 1\}$, $k = 3$, $q_0 = 1 - \alpha - \beta$, $q_1 = \alpha$, $q_2 = \beta$.

with Γ_k being the limit of $(x^{\boldsymbol{u}})_{\boldsymbol{u} \in \Sigma^k}$.

In particular, for the system whose limit is given by (371), we have $\boldsymbol{\xi}^0 = \boldsymbol{\xi}^1 = {}^1\!/_2, \boldsymbol{\xi}^{00} = \boldsymbol{\xi}^{11} = {}^{(\alpha+1)}\!/_{2(1+3\alpha)}, \boldsymbol{\xi}^{01} = \boldsymbol{\xi}^{10} = {}^{\alpha}\!/_7$. It then follows that for this system $\mathcal{H}_\infty \leq H_2\left(\frac{2\alpha}{1+3\alpha}\right) = \mathsf{cap}(\Gamma_2)$. We can also compute $\mathsf{cap}(\Gamma_k)$ for $k = 3, 4, \ldots$. Figure 10 shows the entropy bound we find using 2-mer and 3-mer frequencies. The two bounds are close, which suggests that we may be close to the exact entropy values. However, in the absence of a lower bound, this conjecture cannot be verified. The figure shows that when there is only one possible duplication length, the source of diversity is substitution, as may be expected. As $\alpha \to 1$, the relative number of substitutions increases, causing Γ_k to be close to the uniform distribution, and the entropy tends to 1. On the other hand, as $\alpha \to 0$, only duplications occur. This leads to the generation of low complexity sequences that consists of long runs of 0s and 1s, and thus entropy that is close to 0.

Figure 11 shows the entropy bound computed using 3-mer frequencies for the case in which $\Sigma = \{0, 1\}$, $q_1 = \alpha, q_2 = \beta$ and $q_0 = 1 - \alpha - \beta$. So in this system, duplications of lengths 1 and 2 are both possible. It can be seen that similar to Figure 10, even a small probability of substitution leads to relatively high values of entropy. Furthermore, we note that, as may be expected, longer duplications lead to a smaller value of entropy.

E. Finite-time Analysis of k-mer Frequencies and Waiting Time in Noisy Tandem Duplication Systems

In this subsection, we present the results derived in [63]. In contrast with the results in the previous section, finite-time behavior of k-mer frequencies in NTD systems is studied. Bounds on the expected trajectories of k-mers are established and the rate of convergence is provided. The analysis then extends to the second-moment of the k-mer trajectories, characterizing the variation around expected paths, with which waiting times are estimated.

1) Evolution of k-mer frequencies

We first provide a simple analysis of how a noisy duplication $\mathcal{T}_\ell^d$ affects the occurrences of k-mers. Consider the evolution from $\boldsymbol{s}_n$ to $\boldsymbol{s}_{n+1}$ under a noisy duplication $\mathcal{T}_\ell^d$:

$$\boldsymbol{s}_n = \cdots a_i a_{i+1} \cdots a_{i+\ell} a_{i+\ell+1} \cdots \tag{374}$$

$$\boldsymbol{s}_{n+1} = \cdots a_i \overline{a_{i+1} \cdots a_{i+\ell}} \underline{b_{i+1} \cdots b_{i+\ell}} a_{i+\ell+1} \cdots , \tag{375}$$

where a_i is the i-th symbol of $\boldsymbol{s}_n$ and $b_{i+1} \cdots b_{i+\ell}$ is the approximate copy of $a_{i+1} \cdots a_{i+\ell}$ created by $\mathcal{T}_\ell^d$. Assume $k \leq |\boldsymbol{s}_n|$. From $\boldsymbol{s}_n$ to $\boldsymbol{s}_{n+1}$, a number $\ell + k - 1$ of k-substrings in $\boldsymbol{s}_{n+1}$ are newly created, denoted $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{\ell+k-1}$. Specifically, $\boldsymbol{y}_j$ is the k-substring of $\boldsymbol{s}_{n+1}$ whose last symbol is b_{i+j} if $j \leq \ell$, and is a_{i+j} if $j > \ell$. Similarly, a number $k - 1$ of k-substrings in $\boldsymbol{s}_n$ are eliminated and do not appear in $\boldsymbol{s}_{n+1}$. We denote them $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k-1}$, where $\boldsymbol{z}_l$ is the k-substring of $\boldsymbol{s}_n$ whose last symbol is $a_{i+\ell+l}$. For instance, consider $\boldsymbol{s}_n = \mathsf{GATAC}$. A noisy duplication $\mathcal{T}_3^1$ on $\boldsymbol{s}_n$ could result in $\boldsymbol{s}_{n+1} = \mathsf{G}\overline{\mathsf{ATA}}\underline{\mathsf{CTA}}\mathsf{C}$, where the duplicated 3-mer is overlined and the copy is underlined with the first symbol substituted. For $k = 2$, the altered (created or eliminated) 2-mers are $\boldsymbol{y}_1 = \mathsf{AC}, \boldsymbol{y}_2 = \mathsf{CT}, \boldsymbol{y}_3 = \mathsf{TA}, \boldsymbol{y}_4 = \mathsf{AC}, \boldsymbol{z}_1 = \mathsf{AC}$.

Thus, given $\mathcal{T}_\ell^d$, the number of occurrences of a k-mer $\boldsymbol{u}$ changes by

$$\mu_{n+1}^{\boldsymbol{u}} - \mu_n^{\boldsymbol{u}} = \sum_{j=1}^{\ell+k-1} I\big(\boldsymbol{y}_j, \boldsymbol{u}\big) - \sum_{l=1}^{k-1} I(\boldsymbol{z}_l, \boldsymbol{u}). \tag{376}$$

We will refer to equation (376) frequently in the rest of the paper.

The following theorem puts $\mathbb{E}[\boldsymbol{x}_n]$ in the form of a recurrence equation similar to the TDS system.

Theorem 55. *Consider the noisy duplication string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$. If the length L_0 of the initial string $\boldsymbol{s}_0$ is greater than ℓ, then for any $\ell < k \leq L_0$, the k-mer frequency vector $\boldsymbol{x}_n$ satisfies*[8]

$$\mathbb{E}[\boldsymbol{x}_{n+1}] - \mathbb{E}[\boldsymbol{x}_n] = \frac{A_k}{L_{n+1}} \mathbb{E}[\boldsymbol{x}_n] \tag{377}$$

for some constant matrix $A_k \in \mathbb{R}^{|\Sigma^k| \times |\Sigma^k|}$ determined by $\boldsymbol{q}$, k, ℓ and independent of any other quantities. Further, all eigenvalues of A_k have non-positive real parts.

With a little abuse of notation, we redefine matrix A_k as the *characteristic matrix* of the NTD system for k-mers. In Section III-D and [34], this theorem is used as part of a stochastic approximation framework to find almost-sure limit sets/points for $\boldsymbol{x}_n$ as $n \to \infty$. The proof and the construction of matrix A_k were provided in Section III-D for TDS systems and we give a sketched proof for NTD systems in Appendix D1.

2) First-moment trajectories of k-mer frequencies

In this section, we study the expected trajectories of k-mer occurrences $\boldsymbol{\mu}_n$ and frequencies $\boldsymbol{x}_n$ as $\boldsymbol{s}_n$ evolves under noisy duplications. The k-mer frequency vector $\boldsymbol{x}_n$ will be represented in a basis composed of eigenvectors of the characteristic matrix A_k. The coefficients of this basis representation are bounded from below and above. Our analysis in this paper is limited to cases in which A_k has only real eigenvalues. In all examples that we have studied, the eigenvalues of A_k are indeed real. We conjecture that these hold for all noisy duplication systems. Recall that eigenvalues of A_k all have non-positive real part, so they are non-positive real numbers.

[8] Note that it suffices to consider $k > \ell$ since substring frequencies of smaller lengths are linear functions of substring frequencies of larger lengths. The assumption $k \leq L_0$ is to avoid complications of defining k-substrings in strings of lengths less than k.

Let $m = |\Sigma|^k$. When A_k is diagonalizable, we choose an eigenbasis V of A_k, i.e., V contains m linearly independent eigenvectors that form a basis for $\mathbb{R}^m$. Write $V = \{\boldsymbol{v}_s : 1 \leq s \leq m\}$ and let λ_s be the corresponding eigenvalue of $\boldsymbol{v}_s$, $1 \leq s \leq m$. For every $n \geq 0$, we represent the k-mer frequency vector $\boldsymbol{x}_n$ as $\boldsymbol{x}_n = \sum\limits_{s=1}^{m} \alpha_n^s \boldsymbol{v}_s$. The next theorem provides bounds on the expected values of coefficients α_n^s.

Theorem 56. *Consider the noisy duplication system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$ with characteristic matrix A_k and k-mer frequency vectors $\boldsymbol{x}_n = \sum_{s=1}^m \alpha_n^s \boldsymbol{v}_s$. If A_k is diagonalizable, and all eigenvalues of A_k are real and no smaller than $-\frac{L_0}{2}$,*

1) *For $1 \leq s \leq m$ such that $\lambda_s = 0$ or $\alpha_0^s = 0$,*

$$\mathbb{E}[\alpha_n^s] = \alpha_0^s \quad \text{for all } n \in \mathbb{N}.$$

2) *For $1 \leq s \leq m$ such that $\lambda_s \neq 0$ and $\alpha_0^s \neq 0$,*

$$T_n^s < \frac{\mathbb{E}[\alpha_n^s]}{\alpha_0^s} < U_n^s, \tag{378}$$

where

$$U_n^s = \left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right)^{\frac{\lambda_s}{\ell}} e^{\lambda_s^2/(L_1 \ell)} \left(1 + \frac{\lambda_s}{L_n}\right),$$

$$T_n^s = \left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right)^{\frac{\lambda_s}{\ell}} e^{-\lambda_s^2/(L_n \ell)} \left(1 + \frac{\lambda_s}{L_1}\right),$$

and $L_n = L_0 + n\ell$.

The proof of Theorem 56 is reported in Appendix D2. Recall that $\lambda_s \leq 0$ for all s. The preceding theorem states that the behaviors of both U_n^s and T_n^s are dominated by the factor $(\frac{\lambda_s + L_n}{\lambda_s + L_1})^{\frac{\lambda_s}{\ell}}$, which is of order $\Theta(n^{\frac{\lambda_s}{\ell}})$. This implies that when $\mathbb{E}[\boldsymbol{x}_n]$ is represented in the eigenbasis, for an eigenvector whose corresponding eigenvalue λ is not 0, its component in $\mathbb{E}[\boldsymbol{x}_n]$ converges to 0 at the rate of $n^{\lambda/\ell}$, while for eigenvectors whose corresponding eigenvalues are 0, their components remain unchanged and determine the expected value of the limit of $\boldsymbol{x}_n$. Among all nonzero eigenvalues, the largest one determines the convergence rate, since its corresponding components vanish at the slowest rate.

Example 14. We demonstrate the bounds provided in Theorem 56 for a simple noisy duplication system over the alphabet $\{0, 1\}$ with $\ell = 1$ and $\boldsymbol{q} = (q_1^0, q_1^1) = (1 - \delta, \delta)$. Specifically, in each step, a random symbol a is chosen uniformly from the evolving string, and a symbol b is inserted immediately after a, where $q_1^0 = \Pr(b{=}a) = 1 - \delta$ and $q_1^1 = \Pr(b{\neq}a) = \delta$. The vector $\boldsymbol{x}_n = (x_n^{00}, x_n^{01}, x_n^{10}, x_n^{11})^T$ denotes the frequencies of 2-mers. The characteristic matrix of this system for 2-mers can be shown to be

$$A_2 = \begin{bmatrix} -2\delta & 1-\delta & \delta & 0 \\ \delta & -1 & 0 & \delta \\ \delta & 0 & -1 & \delta \\ 0 & \delta & 1-\delta & -2\delta \end{bmatrix}. \tag{379}$$

For $\delta < 1/2$, A_2 is diagonalizable with four distinct eigenvalues: $\lambda_1 = 0, \lambda_2 = -2\delta, \lambda_3 = -1, \lambda_4 = -2\delta - 1$. So we pick a basis of $\mathbb{R}^4$ which is composed of 4 eigenvectors $\boldsymbol{v}_i, 1 \leq i \leq 4$, corresponding to $\lambda_i, 1 \leq i \leq 4$, respectively:

$$\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix} = \begin{bmatrix} 1/(2+4\delta) & -1 & -1 & 1 \\ \delta/(1+2\delta) & 0 & 1 & -1 \\ \delta/(1+2\delta) & 0 & -1 & -1 \\ 1/(2+4\delta) & 1 & 1 & 1 \end{bmatrix}. \tag{380}$$

Since $\lambda_1 = 0$, the limit of $\mathbb{E}[\boldsymbol{x}_n]$ is $\boldsymbol{v}_1$ (it can be shown that $\boldsymbol{x}_n$ converges to $\boldsymbol{v}_1$ almost surely) and the other components vanish at the corresponding rates. Figure 12 shows the vanish of the coefficient α_n^2 when $q_1^1 = 0.1, q_1^0 = 0.9$, and $\boldsymbol{s}_0 = 1101100111$. The figure illustrates the trajectory of the ratio between the expected value of α_n^2 and α_0^2 as well as the upper and lower bounds given in Theorem 56 as n ranges from 0 to 50. The average value for α_n^2/α_0^2 from 5000 independent trials of the process is also given.

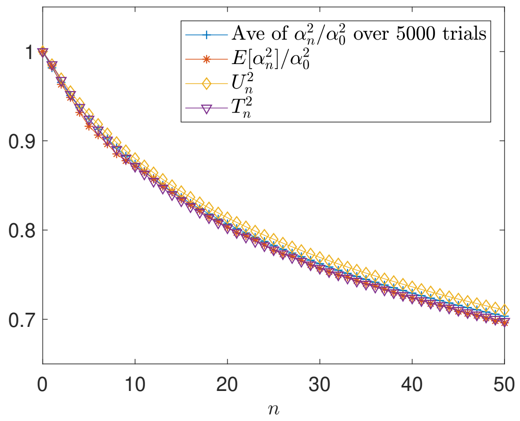


Figure 12: Vanish of the coefficient of $\boldsymbol{v}_2$ in $\mathbb{E}[\boldsymbol{x}_n]$ vs the number of mutations in a noisy duplication string system with $\Sigma = \{0, 1\}$, $\boldsymbol{s}_0 = 1101100111$, $q_1^1 = 0.9$, $q_1^0 = 0.1$.

Theorem 56 characterizes the k-mer frequencies for cases when A_k is diagonalizable. In all examples considered in this paper, the characteristic matrices are indeed diagonalizable. When A_k is not diagonalizable, similar but more complex results can be obtained. We provide this analysis in Appendix D3.

3) Second-moment trajectories of k-mer frequencies

In this section, we provide a second-moment analysis of the k-mer frequencies. We show that similar to Theorem 55, the expected values of products of k-mer frequencies, i.e., $\mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}]$, $\boldsymbol{v}, \boldsymbol{w} \in \Sigma^k$, can also be expressed as a recurrent equation. To show this, we first present the following lemma, which expresses the difference between $\mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}]$ and $\mathbb{E}\left[x_{n+1}^{\boldsymbol{v}} x_{n+1}^{\boldsymbol{w}}\right]$ as a linear function of k-mer and $(2k-2)$-mer frequencies.

Lemma 57. *Consider the noisy string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$ with characteristic matrix A_k for k-mers. For any two k-mers $\boldsymbol{v}, \boldsymbol{w}$ (not necessarily distinct),*

$$\mathbb{E}[x_{n+1}^{\boldsymbol{v}} x_{n+1}^{\boldsymbol{w}}] - \left(\frac{L_n}{L_{n+1}}\right)^2 \mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}] = \frac{\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T}{(L_{n+1})^2} \mathbb{E}[\boldsymbol{x}_n(2k-2)] \tag{381}$$

$$+ \frac{L_n}{(L_{n+1})^2} (\mathbb{E}[x_n^{\boldsymbol{w}} \cdot H_{i_{\boldsymbol{v}}} \cdot \boldsymbol{x}_n(k)] + \mathbb{E}[x_n^{\boldsymbol{v}} \cdot H_{i_{\boldsymbol{w}}} \cdot \boldsymbol{x}_n(k)]), \tag{382}$$

where H is the matrix $A_k + \ell I_{|\Sigma^k|}$ and H_m denotes the m-th row of H, $\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}$ is a constant vector of length $|\Sigma|^{2k-2}$ determined by $\boldsymbol{v}$, $\boldsymbol{w}$, $\boldsymbol{q}$, k, ℓ and independent of any other quantities. Note that $\boldsymbol{x}_n(0)$ is defined to be the zero vector for all n.

The proof of Theorem 57 is reported in Appendix D4.

We next consider the vector containing all products of the form $x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}$. Let Σ_2^k denote the set of all unordered pairs of k-mers, i.e.,

$$\Sigma_2^k = \{(\boldsymbol{v}, \boldsymbol{w}) : \boldsymbol{v}, \boldsymbol{w} \in \Sigma^k\}.$$

Define $\boldsymbol{r}_n(k) = (x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}})_{(\boldsymbol{v},\boldsymbol{w}) \in \Sigma_2^k}$, i.e., the vector of products of frequencies of every two k-mers in $\boldsymbol{s}_n$. The elements in $\boldsymbol{r}_n(k)$ are lexicographically ordered according to the $2k$-mer $\boldsymbol{v}\boldsymbol{w}$.

Theorem 58. *In the noisy string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$ with characteristic matrix A_k for k-mers and characteristic matrix A_{2k-2} for $(2k-2)$-mers,*

$$\begin{pmatrix} \mathbb{E}[\boldsymbol{r}_{n+1}(k)] \\ \mathbb{E}[\boldsymbol{x}_{n+1}(2k-2)] \end{pmatrix} = \begin{bmatrix} \left(\frac{L_n}{L_{n+1}}\right)^2 \left(I + \frac{G}{L_n}\right) & \frac{D}{(L_{n+1})^2} \\ \boldsymbol{0} & I + \frac{A_{2k-2}}{L_{n+1}} \end{bmatrix} \begin{pmatrix} \mathbb{E}[\boldsymbol{r}_n(k)] \\ \mathbb{E}[\boldsymbol{x}_n(2k-2)] \end{pmatrix}, \tag{383}$$

where G, L are both constant matrices determined by $\boldsymbol{q}, k, \ell$ and independent of any other quantities.

Proof: Consider the matrix H given by Lemma 57. For any two k-mers $\boldsymbol{v}$ and $\boldsymbol{w}$,

$$x_n^{\boldsymbol{w}} \cdot H_{i_{\boldsymbol{v}}} \boldsymbol{x}_n(k) = x_n^{\boldsymbol{w}} \cdot \sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{v}}, i_{\boldsymbol{u}}} x_n^{\boldsymbol{u}} = \sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{v}}, i_{\boldsymbol{u}}} \cdot x_n^{\boldsymbol{w}} x_n^{\boldsymbol{u}}, \tag{384}$$

$$x_n^{\boldsymbol{v}} \cdot H_{i_{\boldsymbol{w}}} \boldsymbol{x}_n(k) = x_n^{\boldsymbol{v}} \cdot \sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{w}}, i_{\boldsymbol{u}}} x_n^{\boldsymbol{u}} = \sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{w}}, i_{\boldsymbol{u}}} \cdot x_n^{\boldsymbol{v}} x_n^{\boldsymbol{u}}, \tag{385}$$

where H_{m_1, m_2} denotes the element in the m_1-th row and m_2-th column of matrix H.

It follows that

$$\mathbb{E}[x_{n+1}^{\boldsymbol{v}} x_{n+1}^{\boldsymbol{w}}] = \left(\frac{L_n}{L_{n+1}}\right)^2 \left(\mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}] + \frac{1}{L_n} \left(\sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{v}}, i_{\boldsymbol{u}}} \cdot \mathbb{E}[x_n^{\boldsymbol{w}} x_n^{\boldsymbol{u}}] + \sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{w}}, i_{\boldsymbol{u}}} \cdot \mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{u}}] \right) \right) \tag{386}$$

$$+ \frac{\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T}{(L_{n+1})^2} \mathbb{E}[\boldsymbol{x}_n(2k-2)] \tag{387}$$

$$= \left(\frac{L_n}{L_{n+1}}\right)^2 \left(\mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}] + \frac{1}{L_n} \sum_{(\boldsymbol{u},\boldsymbol{u}') \in \Sigma_2^k} (g_{\boldsymbol{u},\boldsymbol{u}'}^1 + g_{\boldsymbol{u},\boldsymbol{u}'}^2) \mathbb{E}\left[x_n^{\boldsymbol{u}} x_n^{\boldsymbol{u}'}\right] \right) + \left(\frac{1}{L_{n+1}}\right)^2 \boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T \mathbb{E}[\boldsymbol{x}_n(2k-2)] \tag{388}$$

$$= \left(\frac{L_n}{L_{n+1}}\right)^2 \left(\mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}] + \frac{1}{L_n} \boldsymbol{g}_{\boldsymbol{v},\boldsymbol{w}}^T \boldsymbol{r}_n(k) \right) + \left(\frac{1}{L_{n+1}}\right)^2 \boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T \mathbb{E}[\boldsymbol{x}_n(2k-2)], \tag{389}$$

where $g_{\boldsymbol{u},\boldsymbol{u}'}^1$ follows from the term $\sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{v}}, i_{\boldsymbol{u}}} \cdot \mathbb{E}[x_n^{\boldsymbol{w}} x_n^{\boldsymbol{u}}]$, $g_{\boldsymbol{u},\boldsymbol{u}'}^2$ follows from the term $\sum_{\boldsymbol{u} \in \Sigma^k} H_{i_{\boldsymbol{w}}, i_{\boldsymbol{u}}} \cdot \mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{u}}]$ and

$$g_{\boldsymbol{u},\boldsymbol{u}'}^1 = \begin{cases} H_{i_{\boldsymbol{v}}, i_{\boldsymbol{u}'}} & \text{if } \boldsymbol{u} = \boldsymbol{w}, \\ H_{i_{\boldsymbol{v}}, i_{\boldsymbol{u}}} & \text{if } \boldsymbol{u} \neq \boldsymbol{w}, \boldsymbol{u}' = \boldsymbol{w}, \\ 0, & \text{otherwise}, \end{cases} \quad g_{\boldsymbol{u},\boldsymbol{u}'}^2 = \begin{cases} H_{i_{\boldsymbol{w}}, i_{\boldsymbol{u}'}} & \text{if } \boldsymbol{u} = \boldsymbol{v}, \\ H_{i_{\boldsymbol{w}}, i_{\boldsymbol{u}}} & \text{if } \boldsymbol{u} \neq \boldsymbol{v}, \boldsymbol{u}' = \boldsymbol{v}, \\ 0, & \text{otherwise}, \end{cases} \tag{390}$$

$g_{\boldsymbol{v},\boldsymbol{w}}$ is the vector $\left(g_{\boldsymbol{u},\boldsymbol{u}'}^1 + g_{\boldsymbol{u},\boldsymbol{u}'}^2\right)_{(\boldsymbol{u},\boldsymbol{u}') \in \Sigma_2^k}$.

Since $\boldsymbol{r}_n(k)$ is the vector containing $x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}$ for all $(\boldsymbol{v}, \boldsymbol{w})$ in Σ_k^2, we can write (389) in the matrix form and get

$$\mathbb{E}[\boldsymbol{r}_{n+1}(k)] = \left(\frac{L_n}{L_{n+1}}\right)^2 \left(\mathbb{E}[\boldsymbol{r}_n(k)] + \frac{G}{L_n} \boldsymbol{r}_n(k) \right) + \frac{D}{(L_{n+1})^2} \mathbb{E}[\boldsymbol{x}_n(2k-2)], \tag{391}$$

where G is the matrix with rows $\boldsymbol{g}_{\boldsymbol{v},\boldsymbol{w}}^T$ and D is the matrix with rows $\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T$ for all $(\boldsymbol{v}, \boldsymbol{w}) \in \Sigma_k^2$. The remaining of the desired result thus follows by applying Theorem 55 on the $(2k-2)$-mers. ■

Similar to the characteristic matrix A_k, matrices G and D characterize the second-moment behavior of the k-mer frequencies. Specifically, Theorem 58 provides a way of computing expected values of products of k-mer frequencies, which can be used to find variances of k-mer frequencies together with the expected values.

Example 15. Figure 13 compares the variance of x_n^{12} computed using (383) with the sample variance of 10000 independent trials in the system with parameters $\ell = 1, \boldsymbol{q} = \left(q_1^0, q_1^1\right) = (1-\delta, \delta)$, $\Sigma = \{0, 1, 2\}$, $\boldsymbol{s}_0 = 0000000000$, and $\delta = 0.2$. Note that since x_n^{12} is bounded, the variance cannot increase unbounded. Indeed, if $x_n^{\boldsymbol{u}}$ converges to a single point, the variance vanishes.

4) Bounding waiting times by first- and second-moment trajectories

In this section, we study applications of the first- and second-moment trajectories on estimating the waiting times. This analysis also enables us to quantify the effect of mutation probabilities on waiting times.

Let $\boldsymbol{u}$ be the string of interest and let $\hat{n}_{\boldsymbol{u}}$ be such that $\mathbb{E}[\mu_n^{\boldsymbol{u}}] \simeq 1$ for $n = \hat{n}_{\boldsymbol{u}}$. If the expected frequency $\mathbb{E}[x_n^{\boldsymbol{u}}]$ is increasing, for $n \ll \hat{n}_{\boldsymbol{u}}$, the probability of the existence of $\boldsymbol{u}$ in $\boldsymbol{s}_n$ is small and thus we can view $\hat{n}_{\boldsymbol{u}}$ as a rough estimate for the waiting time $\tau_{\boldsymbol{u}}$.

Example 16. Consider a noisy duplication system with parameters $\ell = 1$, $\boldsymbol{q} = (q_1^0, q_1^1) = (1-\delta, \delta)$ and over the alphabet $\Sigma = \{0, 1, 2\}$. Let $\boldsymbol{s}_0$ be 0000000000. We show a comparison of the expected waiting times $\mathbb{E}[\tau_{\boldsymbol{u}}]$ and $\hat{n}_{\boldsymbol{u}}$ for 2-mers 11 and 12 in Figure 14, where expected waiting times are obtained by averaging over 1000 independent simulation trials and $\hat{n}_{11}$ and $\hat{n}_{12}$ are calculated using (377). It can be seen that $\hat{n}_{12}$ is a better estimate for τ_{12} and $\hat{n}_{11}$ has a larger relative error. This is

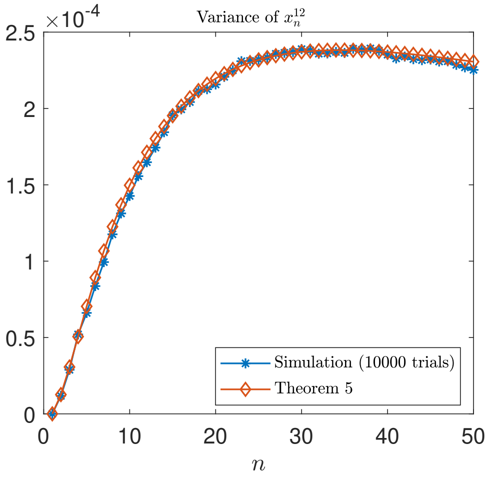


Figure 13: Variance of x_n^{12} vs the number of mutations.

due to the higher variance of μ_n^{11}, which is discussed further in this section.

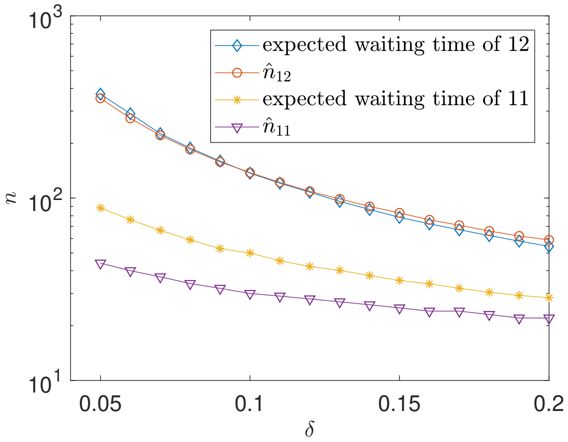


Figure 14: Expected waiting times for 11 and 12 and $\hat{n}_{11}, \hat{n}_{12}$ vs δ for $\Sigma = \{0, 1, 2\}, \boldsymbol{s}_0 = 0000000000, q_1^0 = 1 - \delta, q_1^1 = \delta$.

More generally, the cdf of $\tau_{\boldsymbol{u}}(m)$, the first time $\boldsymbol{u}$ appears m times, can be shown to be upper bounded by a constant times $\mathbb{E}[\mu_n^{\boldsymbol{u}}]$ in the next theorem.

Theorem 59. *Consider the noisy string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$. For $\boldsymbol{u} \in \Sigma^k$, if $\mathbb{E}[x_n^{\boldsymbol{u}}]$ is non-decreasing in n,*

$$\Pr(\tau_{\boldsymbol{u}}(m) \le n) \le \left(\frac{1}{m} + \frac{\max(|\boldsymbol{u}| - \ell - 1, 0)}{\ell} + \left(1 - q_\ell^0\right)\right)\mathbb{E}[\mu_n^{\boldsymbol{u}}].$$

The proof of Theorem 59 is reported in Appendix D5. We demonstrate an application of the theorem via the following example.

Example 16 (continued)**.** We study analytically how the waiting times for 11 and 12, i.e, τ_{11} and τ_{12}, vary as $\delta \to 0$ with the help of Theorem 56 and 59. After finding the characteristic matrix A_2 for this system, it can be shown by a similar proof to Theorem 56 that the expected frequencies of $11, 12$ are non-decreasing. Theorem 59 then gives that for $\boldsymbol{u} = 11, 12$,

$$\Pr(\tau_{\boldsymbol{u}} \le n) \le (1 + \delta)\mathbb{E}[\mu_n^{\boldsymbol{u}}]. \tag{392}$$

By the eigenbasis analysis and bounding coefficients of the eigenbasis representation using Theorem 56, we can get

$$\mathbb{E}[x_n^{11}] < \frac{1}{3} + \frac{2L_1\delta}{n} - \frac{1}{12}n^{-\frac{3}{2}\delta}, \tag{393}$$

$$\mathbb{E}[x_n^{12}] < \frac{\delta}{3} + \frac{2L_1}{n}\delta^2 - \frac{1}{12}\delta n^{-\frac{3}{2}\delta}. \tag{394}$$

The derivation processes are reported in Appendix D6.

For any constant $M > 1$, let $\hat{n} = \hat{\tau}_M^{11} = \frac{c}{aW\left(\frac{ce^{-b/a}}{a}\right)}$, where $a = \frac{3}{2}\delta, b = \log\frac{1}{4}, c = 3\left(\frac{1}{M} - 2L_1\delta\right)$ and $W(\cdot)$ denotes the Lambert W function. As $\delta \to 0$,

$$\frac{1}{4}\hat{n}^{-\frac{3}{2}\delta} = e^{\log\left(\frac{1}{4}\right)+\left(-\frac{3}{2}\delta\right)\log\hat{n}} = e^{-\frac{3}{\hat{n}}\left(\frac{1}{M}-2L_1\delta\right)} = \left(1 - \frac{3}{\hat{n}}\left(\frac{1}{M} - 2L_1\delta\right)\right)(1+o(1)). \tag{395}$$

It then follows that

$$\mathbb{E}\left[\mu_{\hat{n}}^{11}\right] = L_{\hat{n}}\mathbb{E}\left[x_{\hat{n}}^{11}\right] < (L_0 + \hat{n})\left(\frac{1}{3} + \frac{2L_1\delta}{\hat{n}} - \frac{1}{12}\hat{n}^{-\frac{3}{2}\delta}\right) = \frac{1}{M}(1+o(1)). \tag{396}$$

Thus by (392), for any constant $M > 1$, $\Pr(\tau_{11} \leq \hat{\tau}_M^{11}) \leq \frac{1+o(1)}{M}$ as $\delta \to 0$. Hence, $\hat{\tau}^{11}(M)$ is a lower bound for τ_{11} that holds with probability at least $1 - \frac{1}{M}$.

Similarly, $\hat{\tau}_M^{12} = \frac{c'}{aW\left(\frac{c'e^{-b/a}}{a}\right)}$ where $c' = 3\left(\frac{1}{M\delta} - 2L_1\delta\right)$ is a lower bound for τ_{12} that holds with probability at least $1 - \frac{1}{M}$.

Next, we study the waiting time in further detail by the second-order analysis discussed in Section III-E3. We derive a lower bound on the cdf of $\tau_{\boldsymbol{u}}(m)$ and show its application with an example.

Let $\sigma_n^{\boldsymbol{u}}$ be the standard deviation of $\mu_n^{\boldsymbol{u}}$. By Chebyshev's inequality,

$$\Pr(\mathbb{E}[\mu_n^{\boldsymbol{u}}] - \gamma\sigma_n^{\boldsymbol{u}} \leq \mu_n^{\boldsymbol{u}} \leq \mathbb{E}[\mu_n^{\boldsymbol{u}}] + \gamma\sigma_n^{\boldsymbol{u}}) > 1 - \frac{1}{\gamma^2}, \tag{397}$$

for any $\gamma > 0$. Therefore, by computing the expected value and variance of $\mu_n^{\boldsymbol{u}}$ using Theorems 55, 56 and 58, we can bound $\mu_n^{\boldsymbol{u}}$ in a range that contains most of the probability mass.

Example 17. Consider the same noisy duplication system as Example 16 and let $\delta = 0.2$. Figure 15 shows intervals computed using (397) for μ_n^{11} and μ_n^{12} that have probability at least 8/9, i.e., $\gamma = 3$. We can observe that the variance of μ_n^{11} is much larger than that of μ_n^{12}. This is in agreement with Figure 14, where it is observed that $\hat{n}_{\boldsymbol{u}}$, obtained based on average trajectories, better matches the waiting time for $\boldsymbol{u} = 12$ compared to $\boldsymbol{u} = 11$. The higher variance of μ_n^{11} is likely due to its high autocorrelation, which means that as soon as an instance of 1 is created, many instances of 11 can be produced by duplicating it. When variance is high, (397) will lead to loose or trivial bounds.

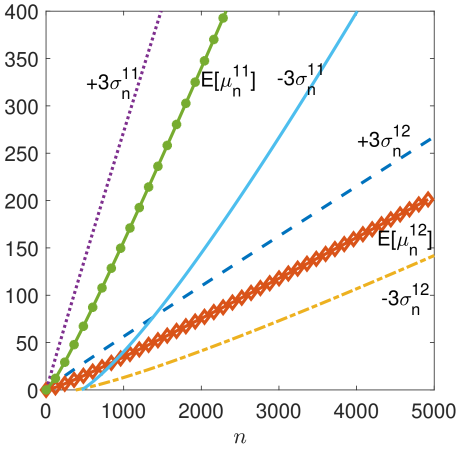


Figure 15: Expected values and $\pm 3\sigma$ range for μ_n^{12} and μ_n^{11}, which contain 8/9 of the probability. $\Sigma = \{0, 1, 2\}$, $\boldsymbol{s}_0 = 0000000000$, $q_1^1 = 0.8$, $q_1^0 = 0.2$.

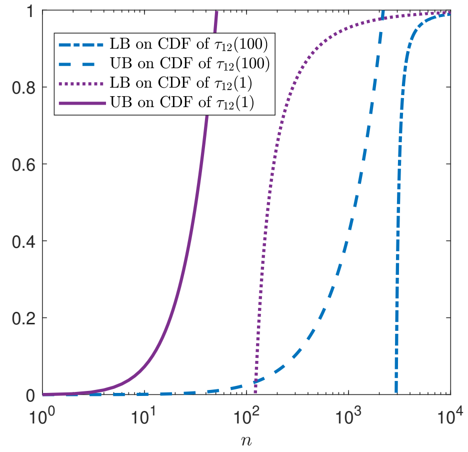


Figure 16: Bounds on $\Pr(\tau_{12}(m) \leq n)$ for $m = 1, 100$. LB and UB stand for lower and upper bounds. $\Sigma = \{0, 1, 2\}$, $\boldsymbol{s}_0 = 0000000000$, $q_1^1 = 0.8$, $q_1^0 = 0.2$.

Note that for any positive integer n, $\mu_n^{\boldsymbol{u}} \geq m$ is a sufficient (but not necessary) condition for $\tau_{\boldsymbol{u}}(m) \leq n$. Hence, from (397),

$$\Pr(\tau_{\boldsymbol{u}}(m) \leq n) \geq \Pr(\mu_n^{\boldsymbol{u}} \geq m) > 1 - \frac{1}{\gamma^2}, \tag{398}$$

where $\gamma = (\mathbb{E}[\mu_n^{\boldsymbol{u}}] - m)/\sigma_n^{\boldsymbol{u}}$. This tells us that we are likely to see m occurrences of $\boldsymbol{u}$ in the sequence not long after the expected number of occurrences of $\boldsymbol{u}$ hits m, thus providing a lower bound on the CDF $\Pr(\tau_{\boldsymbol{u}}(m) \leq n)$ and a probabilistic upper bound on $\tau_{\boldsymbol{u}}(m)$.

Example 17 (continued)**.** Figure 16 illustrates lower and upper bounds on the CDF of $\tau_{12}(m)$ for $m = 1$ and 100, where the upper bounds are based on Theorem 59. The sharpness of the curves in the figure implies that in fact, most of the probability of $\tau_{12}(m)$ is concentrated in a small interval. In particular, the bounds provide the order of magnitude of the waiting times.

IV. Conclusion

The performances of deduplication algorithms on data streams with approximate repeats, a situation that is common in practice. For simplicity, the process producing approximate repeats is modeled as independent bit-wise Bernoulli substitutions. Correctly choosing the chunk lengths is critical to the success of deduplication. With chunk lengths improperly chosen, it was shown that deduplication algorithms can be substantially suboptimal. With optimally chosen chunk lengths, deduplication in the fixed-length scheme is shown to achieve performance within a constant factor of optimal for a specific family of source models and with the knowledge of source parameters. Additionally, appropriately choosing the length of the marker leads to suitable chunk lengths for variable-length deduplication, resulting in arbitrarily large compression ratios as source entropy gets smaller. From the perspective of universal compression, Theorems 37, 38 show that the dictionary-based pattern compressor in deduplication algorithms has high pattern redundancy. Deduplication algorithms, although effective in practice, are far from optimal and the saving mainly results from removing duplicate chunks. Thus, finding constraints on pattern compressors that can achieve low redundancies while keeping time and memory costs affordable can benefit the performance of deduplication algorithms.

While shedding light on certain important aspects of the problem of deduplication, the information-theoretic analysis of data deduplication provides a wealth of open problems. For example, while VLD was shown to achieve high compression ratios, it is not known whether it is order optimal. Moreover, the source models proposed only included substitution edits. However, in practice, insertions, deletions and substitutions of single symbols, as well as longer strings, occur frequently. The probabilistic description of the source models can also be further refined based on experiments. Therefore, to gain a fuller understanding, it is important to study deduplication algorithms under more general source models and edit processes. Pattern compressors being of high redundancy also provides an intriguing direction for future work, which may benefit from Lemmas 33 and 34 (general ways for computing redundancies are derived). Another direction of interest is determining families of distributions for which common constraints, such as $\mathcal{C}_2$, lead to low redundancy. Such families would represent suitable applications for existing deduplication algorithms.

The limiting behavior of the stochastic duplication system, tandem duplication with substitution, is studied. Stochastic approximation framework is used to compute the limits of k-mer frequencies for tandem duplications and substitutions. A method is also provided for determining upper bounds on the entropy of these systems. The finite-time behavior of noisy duplication string systems is also studied by representing the average trajectories of the frequencies of k-mers in an eigenbasis of the characteristic matrix of the system. It is shown that the coordinate corresponding to eigenvalue $\lambda \neq 0$ converges to 0 with rate approximately $n^{\lambda/\ell}$. We also provided a method for computing the second moment of k-mer frequencies, as well as bounds on the CDFs of waiting times, which are the first such bounds for any type of mutation other than independent substitution.

References

[1] J. Aberg, Y. M. Shtarkov, and B. J. Smeets, “Multialphabet coding with separate alphabet description”, in *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, IEEE, 1997, pp. 56–65.

[2] J. Acharya, H. Das, A. Jafarpour, A. Orlitsky, and A. T. Suresh, “Tight bounds for universal compression of large alphabets”, in *2013 IEEE International Symposium on Information Theory*, IEEE, 2013, pp. 2875–2879.

[3] J. Acharya, H. Das, and A. Orlitsky, “Tight bounds on profile redundancy and distinguishability”, in *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, 2012, pp. 3257–3265.

[4] C. Adami, “Information theory in molecular biology”, *Physics of Life Reviews*, vol. 1, no. 1, pp. 3–22, 2004.

[5] T. M. Apostol, *Introduction to analytic number theory*. Springer Science & Business Media, 2013.

[6] G. Battail, “Biology needs information theory”, *Biosemiotics*, vol. 6, no. 1, pp. 77–103, 2013.

[7] D. Bhagwat, K. Eshghi, D. D. Long, and M. Lillibridge, “Extreme binning: Scalable, parallel deduplication for chunk-based file backup”, in *2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, IEEE, 2009, pp. 1–9.

[8] V. S. Borkar, *Stochastic Approximation*. Cambridge University Press, 2008.

[9] S. Boucheron, A. Garivier, and E. Gassiat, “Coding on countably infinite alphabets”, *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 358–373, 2008.

[10] A. Z. Broder, “Some applications of rabin’s fingerprinting method”, in *Sequences II*, Springer, 1993, pp. 143–152.

[11] R. Bronson, *Matrix methods: An introduction*. Gulf Professional Publishing, 1991.

[12] M. D. Cao, T. I. Dix, L. Allison, and C. Mears, “A simple statistical algorithm for biological sequence compression”, in *2007 Data Compression Conference (DCC’07)*, IEEE, 2007, pp. 43–52.

[13] S. Chandak, K. Tatwawadi, I. Ochoa, M. Hernaez, and T. Weissman, “Spring: A next-generation compressor for fastq data”, *Bioinformatics*, 2018.

[14] Y. M. Chee, J. Chrisnata, H. M. Kiah, and T. T. Nguyen, “Deciding the confusability of words under tandem repeats”, *arXiv preprint arXiv:1707.03956*, 2017.

[15] X. Chen, S. Kwong, and M. Li, “A compression algorithm for DNA sequences and its applications in genome comparison”, *Genome informatics*, vol. 10, pp. 51–61, 1999.

[16] B. Chern, I. Ochoa, A. Manolakos, A. No, K. Venkat, and T. Weissman, “Reference based genome compression”, in *2012 IEEE Information Theory Workshop*, IEEE, 2012, pp. 427–431.

[17] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[18] S. Dancoff and H. Quastler, “The information content and error rate of living things”, *Essays on the Use of Information Theory in Biology*, 1953.

[19] David Williams, *Probability with Martingales*. Cambridge: Cambridge University Press, 1991.

[20] L. Davisson, “Minimax noiseless universal coding for markov sources”, *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 211–215, 1983.

[21] L. Davisson, R. McEliece, M. Pursley, and M. Wallace, “Efficient universal noiseless source codes”, *IEEE Transactions on Information Theory*, vol. 27, no. 3, pp. 269–279, 1981.

[22] P. Deutsch, “Deflate compressed data format specification version 1.3”, *IETF RFC 1951*, 1996.

[23] L. Dolecek and V. Anantharam, “Repetition error correcting sets: Explicit constructions and prefixing methods”, *SIAM Journal on Discrete Mathematics*, vol. 23, no. 4, pp. 2120–2146, 2010.

[24] I. Drago, E. Bocchi, M. Mellia, H. Slatman, and A. Pras, “Benchmarking personal cloud storage”, in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 205–212.

[25] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras, “Inside dropbox: Understanding personal cloud storage services”, in *Proceedings of the 2012 Internet Measurement Conference*, 2012, pp. 481–494.

[26] R. Durrett and D. Schmidt, “Waiting for Two Mutations: With Applications to Regulatory Sequence Evolution and the Limits of Darwinian Evolution”, *Genetics*, vol. 180, no. 3, pp. 1501–1509, Nov. 2008.

[27] P. Elias, “Universal codeword sets and representations of the integers”, *IEEE transactions on information theory*, vol. 21, no. 2, pp. 194–203, 1975.

[28] O. Elishco, F. Farnoud, M. Schwartz, and J. Bruck, “The entropy rate of some Pólya string models”, *IEEE Trans. Information Theory*, 2019, to appear.

[29] O. Elishco, T. Meyerovitch, and M. Schwartz, “On encoding semiconstrained systems”, *IEEE Transactions on Information Theory*, 2017.

[30] O. Elishco, T. Meyerovitch, and M. Schwartz, “On independence and capacity of multidimensional semiconstrained systems”, *IEEE Transactions on Information Theory*, vol. 64, no. 10, pp. 6461–6483, 2018.

[31] O. Elishco, T. Meyerovitch, and M. Schwartz, “Semiconstrained systems”, *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1688–1702, 2016.

[32] K. Eshghi and H. K. Tang, “A framework for analyzing and improving content-based chunking algorithms”, *Hewlett-Packard Labs Technical Report TR*, vol. 30, no. 2005, 2005.

[33] M. Farach, M. Noordewier, S. Savari, L. Shepp, A. Wyner, and J. Ziv, “On the entropy of DNA: Algorithms and measurements based on memory and rapid convergence”, in *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’95, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1995, pp. 48–57.

[34] F. Farnoud, M. Schwartz, and J. Bruck, “Estimation of duplication history under a stochastic model for tandem repeats”, en, *BMC Bioinformatics*, vol. 20, no. 1, 2019.

[35] F. Farnoud, M. Schwartz, and J. Bruck, “Estimation of duplication history under a stochastic model for tandem repeats”, *BMC Bioinformatics*, vol. 20, no. 1, p. 64, 2019.

[36] F. Farnoud, M. Schwartz, and J. Bruck, “The capacity of string-duplication systems”, *IEEE Trans. Information Theory*, vol. 62, no. 2, pp. 811–824, Feb. 2016.

[37] W. Feller, “An introduction to probability theory and its applications”, *1957*,

[38] A. Garivier, “A lower-bound for the maximin redundancy in pattern coding”, *Entropy*, vol. 11, no. 4, pp. 634–642, 2009.

[39] X. Gou, Z. Wang, N. Li, *et al.*, “Whole-genome sequencing of six dog breeds from continuous altitudes reveals adaptation to high-altitude hypoxia”, *Genome research*, vol. 24, no. 8, pp. 1308–1315, 2014.

[40] S. Greenberg and M. Mohri, “Tight lower bound on the probability of a binomial exceeding its expectation”, *Statistics & Probability Letters*, vol. 86, pp. 91–98, 2014.

[41] B. Hajek, *Random processes for engineers*. Cambridge university press, 2015.

[42] P. Hanus, B. Goebel, J. Dingel, *et al.*, “Information and communication theory in molecular biology”, *Electrical Engineering*, vol. 90, no. 2, pp. 161–173, 2007.

[43] R. Heckel, I. Shomorony, K. Ramchandran, and N. David, “Fundamental limits of DNA storage systems”, in *2017 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2017, pp. 3130–3134.

[44] H. Herzel, W. Ebeling, and A. O. Schmitt, “Entropies of biosequences: The role of repeats”, *Physical Review E*, vol. 50, no. 6, p. 5061, 1994.

[45] N. Iri and O. Kosut, “Third-order coding rate for universal compression of markov sources”, in *2015 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2015, pp. 1996–2000.

[46] P. Jacquet and W. Szpankowski, “Markov types and minimax redundancy for markov sources”, *IEEE Transactions on Information Theory*, vol. 50, no. 7, pp. 1393–1402, 2004.

[47] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, “Noise and uncertainty in string-duplication systems”, in *IEEE Int. Symp. Information Theory (ISIT)*, Aachen, Germany, Jun. 2017.

[48] S. Jain, F. Farnoud, and J. Bruck, “Capacity and expressiveness of genomic tandem duplication”, *IEEE Trans. Information Theory*, vol. 63, no. 10, Oct. 2017.

[49] S. Jain, F. F. Hassanzadeh, M. Schwartz, and J. Bruck, “Duplication-correcting codes for data storage in the DNA of living organisms”, *IEEE Transactions on Information Theory*, vol. 63, no. 8, pp. 4996–5010, 2017.

[50] H. M. Kiah, G. J. Puleo, and O. Milenkovic, “Codes for DNA storage channels”, in *2015 IEEE Information Theory Workshop (ITW)*, IEEE, 2015, pp. 1–5.

[51] G. Korodi and I. Tabus, “Normalized maximum likelihood model of order-1 for the compression of DNA sequences”, in *2007 Data Compression Conference (DCC’07)*, IEEE, 2007, pp. 33–42.

[52] E. Kruus, C. Ungureanu, and C. Dubnicki, “Bimodal content defined chunking for backup streams.”, in *Fast*, 2010, pp. 239–252.

[53] E. S. Lander, L. M. Linton, B. Birren, *et al.*, “Initial sequencing and analysis of the human genome”, *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.

[54] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, “Coding over sets for DNA storage”, in *2018 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2018, pp. 2411–2415.

[55] A. Lenz, A. Wachter-Zeh, and E. Yaakobi, “Bounds on codes correcting tandem and palindromic duplications”, *arXiv preprint arXiv:1707.00052*, 2017.

[56] M. Levy and E. Yaakobi, “Mutually uncorrelated codes for DNA storage”, *IEEE Transactions on Information Theory*, 2018.

[57] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, “An information-based sequence distance and its application to whole mitochondrial genome phylogeny”, *Bioinformatics*, vol. 17, no. 2, pp. 149–154, 2001.

[58] M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezis, and P. Camble, “Sparse indexing: Large scale, inline deduplication using sampling and locality.”, in *Fast*, vol. 9, 2009, pp. 111–123.

[59] P. Liò, A. Politi, M. Buiatti, and S. Ruffo, “High statistics block entropy measures of DNA sequences”, *Journal of Theoretical Biology*, vol. 180, no. 2, pp. 151–160, May 1996.

[60] D. Loewenstern and P. N. Yianilos, “Significantly Lower Entropy Estimates for Natural DNA Sequences”, *Journal of Computational Biology*, vol. 6, no. 1, pp. 125–142, 1999.

[61] H. Lou, M. Schwartz, J. Bruck, and F. Farnoud Hassanzadeh, “Evolution of k-mer frequencies and entropy in duplication and substitution mutation systems”, *IEEE Transactions on Information Theory*, vol. 66, no. 5, pp. 3171–3186, 2020.

[62] H. Lou and F. Farnoud, “Data deduplication with random substitutions”, *IEEE Transactions on Information Theory*, 2022.

[63] H. Lou and F. Farnoud, “Finite-time behavior of k-mer frequencies and waiting times in noisy-duplication systems”, in *Proc. Asilomar Conference on Signals, Systems and Computers*, 2019.

[64] H. Lou and F. F. Hassanzadeh, “Asymptotic analysis of data deduplication with a constant number of substitutions”, in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 3296–3301.

[65] J. MacDonald, “File system support for delta compression”, Ph.D. dissertation, Citeseer, 2000.

[66] H. Mahmoud, *Pólya urn models*. Chapman and Hall/CRC, 2008.

[67] B. H. Marcus, R. M. Roth, and P. H. Siegel, “An introduction to coding for constrained systems”, *Lecture notes*, 2001.

[68] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, “Recalibrating global data center energy-use estimates”, *Science*, vol. 367, no. 6481, pp. 984–986, 2020. eprint: https://science.sciencemag.org/content/367/6481/984.full.pdf.

[69] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 71.

[70] D. T. Meyer and W. J. Bolosky, “A study of practical deduplication”, *ACM Transactions on Storage (ToS)*, vol. 7, no. 4, pp. 1–20, 2012.

[71] O. Milenkovic, G. Alterovitz, G. Battail, *et al.*, “Introduction to the special issue on information theory in molecular biology and neuroscience”, Institute of Electrical and Electronics Engineers, 2010.

[72] M. Mitzenmacher and E. Upfal, *Probability and computing: randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[73] A. S. Motahari, G. Bresler, and N. David, “Information theory of DNA shotgun sequencing”, *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6273–6289, 2013.

[74] N. Mundy and A. J. Helbig, “Origin and evolution of tandem repeats in the mitochondrial DNA control region of shrikes (lanius spp.)”, *Journal of Molecular Evolution*, vol. 59, no. 2, pp. 250–257, 2004.

[75] A. Muthitacharoen, B. Chen, and D. Mazieres, “A low-bandwidth network file system”, in *ACM SIGOPS Operating Systems Review*, ACM, vol. 35, 2001, pp. 174–187.

[76] National Human Genome Research Institute (NHGRI), *The Cost of Sequencing a Human Genome*.

[77] U. Niesen, “An information-theoretic analysis of deduplication”, *IEEE Transactions on Information Theory*, vol. 65, no. 9, pp. 5688–5704, Sep. 2019.

[78] M. Oberhumer, “Lzo-a real-time data compression library”, *http://www. oberhumer. com/opensource/lzo/*, 2008.

[79] S. Ohno, *Evolution by Gene Duplication*. Springer-Verlag, 1970.

[80] A. Orlitsky, N. Santhanam, K. Viswanathan, and J. Zhang, “Limit results on pattern entropy”, *IEEE Trans. Information Theory*, vol. 52, no. 7, pp. 2954–2964, 2006.

[81] A. Orlitsky and N. P. Santhanam, “Speaking of infinity”, *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2215–2230, 2004.

[82] A. Orlitsky, N. P. Santhanam, and J. Zhang, “Universal compression of memoryless sources over unknown alphabets”, *IEEE Transactions on Information Theory*, vol. 50, no. 7, pp. 1469–1481, 2004.

[83] Y. L. Orlov and V. N. Potapov, “Complexity: An internet resource for analysis of DNA sequence complexity”, *Nucleic Acids Research*, vol. 32, no. Web Server issue, W628–W633, Jul. 2004.

[84] D. S. Pavlichin, T. Weissman, and G. Yona, “The human genome contracts again”, *Bioinformatics*, vol. 29, no. 17, pp. 2199–2202, 2013.

[85] S. Quinlan and S. Dorward, “Venti: A new approach to archival storage”, in *FAST*, vol. 2, 2002, pp. 89–101.

[86] M. O. Rabin, “Fingerprinting by random polynomials”, *Technical report*, 1981.

[87] N. Raviv, M. Schwartz, and E. Yaakobi, “Rank-modulation codes for DNA storage with shotgun sequencing”, *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 50–64, 2018.

[88] D. Reinsel, J. Rydning, and J. Gantz, “Worldwide global datasphere forecast, 2020–2024: The covid-19 data bump and the future of data growth”, *Int. Data Corp.(IDC), Framingham, MA, USA, Tech. Rep. US44797920*, 2020.

[89] B. C. Rennie and A. J. Dobson, “On stirling numbers of the second kind”, *Journal of Combinatorial Theory*, vol. 7, no. 2, pp. 116–121, 1969.

[90] J. Rissanen, “Universal coding, information, prediction, and estimation”, *IEEE Transactions on Information theory*, vol. 30, no. 4, pp. 629–636, 1984.

[91] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1964, vol. 3.

[92] A. O. Schmitt and H. Herzel, “Estimating the entropy of DNA sequences”, *Journal of Theoretical Biology*, vol. 188, no. 3, pp. 369–377, Oct. 1997.

[93] B. Schroeder and G. A. Gibson, “Understanding disk failure rates: What does an mttf of 1,000,000 hours mean to you?”, *ACM Transactions on Storage (TOS)*, vol. 3, no. 3, 8–es, 2007.

[94] R. Sedgewick and P. Flajolet, *An introduction to the analysis of algorithms*. Pearson Education India, 2013.

[95] G. I. Shamir, “On the mdl principle for iid sources with large alphabets”, *IEEE transactions on information theory*, vol. 52, no. 5, pp. 1939–1955, 2006.

[96] G. I. Shamir, “Universal lossless compression with unknown alphabets—the average case”, *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 4915–4944, 2006.

[97] C. E. Shannon, “A mathematical theory of communication”, *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[98] P. Shilane, M. Huang, G. Wallace, and W. Hsu, “Wan-optimized replication of backup datasets using stream-informed delta compression”, *ACM Transactions on Storage (ToS)*, vol. 8, no. 4, pp. 1–26, 2012.

[99] A. El-Shimi, R. Kalach, A. Kumar, A. Ottean, J. Li, and S. Sengupta, “Primary data deduplication—large scale study and system design”, in *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*, 2012, pp. 285–296.

[100] Y. M. Shtar'kov, "Universal sequential coding of single messages", *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 3–17, 1987.

[101] J. Shtarkov, "Coding of discrete sources with unknown statistics", *Topics in information theory*, pp. 559–574, 1977.

[102] A. Sievers, K. Bosiek, M. Bisch, *et al.*, "K-mer content, correlation, and position analysis of genome DNA sequences for the identification of function and evolutionary features", *Genes*, vol. 8, no. 4, Apr. 2017.

[103] W. Szpankowski and M. J. Weinberger, "Minimax pointwise redundancy for memoryless models over large alphabets", *IEEE transactions on information theory*, vol. 58, no. 7, pp. 4094–4104, 2012.

[104] Y. Tang, Y. Yehezkeally, M. Schwartz, and F. Farnoud, "Single-error detection and correction for duplication and substitution channels", in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2019.

[105] D. Tautz and T. Domazet-Lošo, "The evolutionary origin of orphan genes", en, *Nature Reviews Genetics*, vol. 12, no. 10, Oct. 2011.

[106] K. Usdin, "The biological effects of simple tandem repeats: Lessons from the repeat expansion diseases", *Genome Research*, vol. 18, no. 7, pp. 1011–1019, Jul. 2008.

[107] R. S. Varga, *Geršgorin and his circles*. Springer Science & Business Media, 2010, vol. 36.

[108] S. Vinga, "Information theory applications for biological sequence analysis", *Briefings in bioinformatics*, vol. 15, no. 3, pp. 376–389, 2013.

[109] B. Vogelstein and K. W. Kinzler, "Cancer genes and the pathways they control", *Nature medicine*, vol. 10, no. 8, 2004.

[110] M. Vrable, S. Savage, and G. M. Voelker, "Cumulus: Filesystem backup to the cloud", *ACM Transactions on Storage (TOS)*, vol. 5, no. 4, pp. 1–28, 2009.

[111] G. Wallace, F. Douglis, H. Qian, *et al.*, "Characteristics of backup workloads in production systems.", in *FAST*, vol. 12, 2012, pp. 4–4.

[112] H. Wan, L. Li, S. Federhen, and J. C. Wootton, "Discovering simple regions in biological sequences associated with scoring schemes", eng, *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, vol. 10, no. 2, pp. 171–185, 2003.

[113] T. A. Welch, "A technique for high-performance data compression", *Computer*, no. 6, pp. 8–19, 1984.

[114] W. Xia, H. Jiang, D. Feng, *et al.*, "A comprehensive study of the past, present, and future of data deduplication", *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1681–1710, 2016.

[115] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system.", in *Fast*, vol. 8, 2008, pp. 269–282.

[116] A. Zielezinski, S. Vinga, J. Almeida, and W. M. Karlowski, "Alignment-free sequence comparison: Benefits, applications, and tools", *Genome Biology*, vol. 18, no. 1, p. 186, Oct. 2017.

[117] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression", *IEEE Transactions on information theory*, vol. 23, no. 3, pp. 337–343, 1977.

[118] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding", *IEEE transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

Appendix

A. Deduplication over $\mathcal{I}_b(\delta)$

1) Proof of Lemma 8

Lemma 8. *Let $\boldsymbol{r}$ be a string drawn uniformly at random from Σ^ℓ. Let $\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_m$ be m iid descendants of $\boldsymbol{r}$ by δ-edit and let $\boldsymbol{r}_{[m]} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_m\}$. For any $\boldsymbol{w} \in \Sigma^\ell$, let $\boldsymbol{w} \in \boldsymbol{r}_{[m]}$ denote the event that $\boldsymbol{w} = \boldsymbol{r}_i$ for some i. Then*

$$\frac{1}{2}\frac{\mathcal{S}_\delta(\ell, m)}{2^\ell} \le \Pr\big(\boldsymbol{w} \in \boldsymbol{r}_{[m]}\big) \le \frac{\mathcal{S}_\delta(\ell, m)}{2^\ell}, \tag{36}$$

and thus the expected number of unique strings in $\boldsymbol{r}_{[m]}$ is bounded between $\frac{1}{2}\mathcal{S}_\delta(\ell, m)$ and $\mathcal{S}_\delta(\ell, m)$.

Furthermore, $\mathcal{S}_\delta(\ell, m)$ takes the following values for different values of ℓ and m:

- *If* $\ell \geq \frac{\log m}{H(\delta)}$*, then*

$$\mathcal{S}_\delta(\ell, m) \geq \frac{1}{4}m. \tag{37}$$

In particular if $\ell \geq \frac{\log m}{\log(\frac{1}{1-\delta})}$*, then*

$$\mathcal{S}_\delta(\ell, m) = m. \tag{38}$$

- *If* $\ell \leq \frac{\log m}{H(\frac{1}{2}, \delta)}$*, then*

$$\mathcal{S}_\delta(\ell, m) \geq 2^{\ell-1}. \tag{39}$$

In particular if $\ell \leq \frac{\log m}{\log(\frac{1}{\delta})}$*, then*

$$\mathcal{S}_\delta(\ell, m) = 2^{\ell}. \tag{40}$$

- *For any* $\delta < \delta' < \frac{1}{2}$*,*

$$\mathcal{S}_\delta(\ell, m) \leq 2^{\ell H(\delta')} + m 2^{-\ell D(\delta' || \delta)}. \tag{41}$$

In particular if $\ell = \frac{\log m}{H(\delta', \delta)}$*, then*

$$\mathcal{S}_\delta(\ell, m) \leq 2^{\ell H(\delta')} + m 2^{-\ell D(\delta' || \delta)} = 2^{\ell H(\delta') + 1}. \tag{42}$$

- *For any values of* ℓ *and* m*,*

$$\mathcal{S}_\delta(\ell, m) \leq \min\bigl(2^{\ell}, m\bigr). \tag{43}$$

Proof: We first prove inequality (36). Given $\boldsymbol{r}$, the probability of a δ-edit descendant being equal to $\boldsymbol{w}$ is $\delta^{d_{\boldsymbol{w},\boldsymbol{r}}}(1-\delta)^{\ell - d_{\boldsymbol{w},\boldsymbol{r}}}$, where $d_{\boldsymbol{w},\boldsymbol{r}}$ denotes the Hamming distance between $\boldsymbol{w}$ and $\boldsymbol{r}$. Therefore,

$$\begin{aligned}
\Pr\bigl(\boldsymbol{w} \in \boldsymbol{r}_{[m]}\bigr) &= 1 - \Pr\bigl(\boldsymbol{w} \notin \boldsymbol{r}_{[m]}\bigr) & (399)\\
&= 1 - \sum_{\boldsymbol{r} \in \Sigma^{\ell}} \Pr(\boldsymbol{r}) \Pr(\boldsymbol{w} \neq \boldsymbol{r}_1 | \boldsymbol{r})^m & (400)\\
&= 1 - \sum_{\boldsymbol{r} \in \Sigma^{\ell}} \Pr(\boldsymbol{r}) \bigl(1 - \delta^{d_{\boldsymbol{w},\boldsymbol{r}}}(1-\delta)^{\ell - d_{\boldsymbol{w},\boldsymbol{r}}}\bigr)^m & (401)\\
&= 1 - \sum_{t=0}^{\ell} \left(\frac{\binom{\ell}{t}}{2^{\ell}} \bigl(1 - \delta^t (1-\delta)^{\ell - t}\bigr)^m \right), & (402)
\end{aligned}$$

where the second equality follows from the fact that $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m$ are iid given $\boldsymbol{r}$ and the last equality follows from the fact that there are $\binom{\ell}{t}$ strings of length ℓ that are at Hamming distance t from $\boldsymbol{w}$. The desired inequalities then follow directly from applying inequalities (1) on $1 - \Bigl(1 - \delta^t(1-\delta)^{\ell-t}\Bigr)^m$.

The expected number of unique strings in $\boldsymbol{r}_{[m]}$ equals

$$\mathbb{E}\left[\sum_{\boldsymbol{w} \in \Sigma^{\ell}} I_{\boldsymbol{w} \in \boldsymbol{r}_{[m]}} \right] = \sum_{\boldsymbol{w} \in \Sigma^{\ell}} \Pr\bigl(\boldsymbol{w} \in \boldsymbol{r}_{[m]}\bigr). \tag{403}$$

So the upper bound $\mathcal{S}_\delta(\ell, m)$ and the lower bound $\frac{1}{2}\mathcal{S}_\delta(\ell, m)$ follow from replacing $\Pr\bigl(\boldsymbol{w} \in \boldsymbol{r}_{[m]}\bigr)$ with its upper and lower bounds, respectively.

We show that $\mathcal{S}_\delta(\ell, m)$ takes the given values for different m and ℓ:

- When $\ell \geq \frac{\log m}{H(\delta)}$, $m\delta^{\delta\ell}(1-\delta)^{(1-\delta)\ell} \leq 1$. It follows that

$$\mathcal{S}_\delta(\ell, m) \geq \sum_{t=\lceil \delta\ell \rceil}^{\ell} \binom{\ell}{t} \min\bigl(1, m\delta^t(1-\delta)^{\ell-t}\bigr) \tag{404}$$

$$= \sum_{t=\lceil \delta\ell \rceil}^{\ell} \binom{\ell}{t} m\delta^t(1-\delta)^{\ell-t} \geq \frac{1}{4}m, \tag{405}$$

where the equality follows from the fact that $m\delta^t(1-\delta)^{\ell-t}$ is decreasing in t so $m\delta^t(1-\delta)^{\ell-t} \leq 1$ for all $t \geq \delta\ell$ and the second inequality follows from the result shown in [40] that for a binomial random variable X with parameters n and p, $\Pr(X \geq np) > \frac{1}{4}$ if $p \geq 1/n$.

Moreover, when $\ell \geq \frac{\log m}{\log(\frac{1}{1-\delta})}$, $m\delta^t(1-\delta)^{\ell-t} \leq 1$ for all t. Hence,

$$\mathcal{S}_\delta(\ell, m) = \sum_{t=0}^{\ell} \binom{\ell}{t} m\delta^t(1-\delta)^{\ell-t} = m. \tag{406}$$

- When $\ell \leq \frac{\log m}{H(\frac{1}{2}, \delta)}$, $m\delta^{\frac{\ell}{2}}(1-\delta)^{\frac{\ell}{2}} \geq 1$. It follows that

$$\mathcal{S}_\delta(\ell, m) \geq \sum_{t=0}^{\lfloor \frac{\ell}{2} \rfloor} \binom{\ell}{t} \min\bigl(1, m\delta^t(1-\delta)^{\ell-t}\bigr) = \sum_{t=0}^{\lfloor \frac{\ell}{2} \rfloor} \binom{\ell}{t} \tag{407}$$

$$\geq 2^{\ell-1}, \tag{408}$$

where the first inequality follows from the fact that $m\delta^t(1-\delta)^{\ell-t} \geq 1$ for all $t \leq \frac{\ell}{2}$.

Moreover, when $\ell \leq \frac{\log m}{\log(\frac{1}{\delta})}$, $m\delta^t \geq 1$ for all t. Hence,

$$\mathcal{S}_\delta(\ell, m) = \sum_{t=0}^{\ell} \binom{\ell}{t} \cdot 1 = 2^\ell. \tag{409}$$

- For any $\delta < \delta' < 1/2$,

$$\mathcal{S}_\delta(\ell, m) \leq \sum_{t=0}^{\lfloor \delta'\ell \rfloor} \binom{\ell}{t} + \sum_{t=\lceil \delta'\ell \rceil}^{\ell} \binom{\ell}{t} m\delta^t(1-\delta)^{\ell-t} \tag{410}$$

$$\leq 2^{\ell H(\delta')} + m2^{-\ell D(\delta' || \delta)}, \tag{411}$$

where the second inequality follows from applying the Chernoff bound on a binomial distribution with parameters ℓ and δ.

When $\ell = \frac{\log m}{H(\delta', \delta)}$, $2^{\ell H(\delta')} = m2^{-\ell D\left(\delta' || \delta\right)}$. So $2^{\ell H(\delta')} + m2^{-\ell D(\delta' || \delta)} = 2^{\ell H(\delta')+1}$ and

$$\mathcal{S}_\delta(\ell, m) \leq 2^{\ell H\left(\delta'\right)+1}. \tag{412}$$

- The upper bounds 2^ℓ and m follow from:

$$\mathcal{S}_\delta(\ell, m) \leq \sum_{t=0}^{\ell} \frac{\binom{\ell}{t}}{2^\ell} = 1, \tag{413}$$

$$\mathcal{S}_\delta(\ell, m) \leq \sum_{t=0}^{\ell} \frac{\binom{\ell}{t}}{2^\ell} m\delta^t(1-\delta)^{\ell-t} = \frac{m}{2^\ell}. \tag{414}$$

■

2) Proofs of Lemma 9 and Lemma 10

Lemma 9. *Suppose K strings of length n are chosen independently and uniformly from Σ^n. Assume each string produces at least m_1 and at most m_2 descendants by δ-edits. For any string $\boldsymbol{w}$ with $|\boldsymbol{w}| = n$, let $G_{\boldsymbol{w}}$ denote the event that $\boldsymbol{w}$ equals one* *or more descendants. Then*

$$\frac{1}{2}\min\left(1,\frac{1}{2}K\frac{\mathcal{S}_\delta(n,m_1)}{2^n}\right)\leq \Pr(G_{\boldsymbol{w}})\leq \min\left(1,K\frac{\mathcal{S}_\delta(n,m_2)}{2^n}\right). \tag{46}$$

Proof: Let the K strings be denoted $\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_K$. Let $G_{\boldsymbol{w}}(i)$ denote the event that $\boldsymbol{w}$ equals one of the descendants of $\boldsymbol{y}_i$. Clearly, $G_{\boldsymbol{w}}(1),G_{\boldsymbol{w}}(2),\ldots,G_{\boldsymbol{w}}(K)$ are independent and

$$G_{\boldsymbol{w}}=\cup_{i=1}^K G_{\boldsymbol{w}}(i). \tag{415}$$

Note that by Lemma 8 and the fact that $\mathcal{S}_\delta(n,m)$ is non-decreasing in m,

$$\frac{1}{2}\frac{\mathcal{S}_\delta(n,m_1)}{2^n}\leq \Pr(G_{\boldsymbol{w}}(i))\leq \frac{\mathcal{S}_\delta(n,m_2)}{2^n}. \tag{416}$$

Applying the union bound on (415) gives

$$\Pr(G_{\boldsymbol{w}})\leq \sum_{i=1}^K \Pr(G_{\boldsymbol{w}}(i))\leq K\frac{\mathcal{S}_\delta(n,m_2)}{2^n}. \tag{417}$$

The desired upper bound follows by noting that 1 is a trivial upper bound.

We then prove the lower bound. By independence,

$$\begin{aligned}\Pr(G_{\boldsymbol{w}})&=\Pr\big(\cup_{i=1}^K G_{\boldsymbol{w}}(i)\big) &(418)\\ &=1-\prod_{i=1}^K(1-\Pr(G_{\boldsymbol{w}}(i))) &(419)\\ &\geq 1-\left(1-\frac{1}{2}\frac{\mathcal{S}_\delta(n,m_1)}{2^n}\right)^K &(420)\\ &\geq \frac{1}{2}\min\left(1,\frac{1}{2}K\frac{\mathcal{S}_\delta(n,m_1)}{2^n}\right), &(421)\end{aligned}$$

where the last inequality follows from inequality (1) that $1-(1-x)^n\geq\frac{1}{2}\min(1,nx)$ for $x\in(0,1)$ and integer n. ■

Lemma 10. *Consider the two-stage fixed-length chunking process with first-stage parsing length $D=L$ and chunk length ℓ. The dictionary sizes $T_F^1(\boldsymbol{s})$ and $T_F^{1/2}(\boldsymbol{s})$ satisfy*

$$\mathbb{E}\big[\big|T_F^1(\boldsymbol{s})\big||\mathcal{E}_u\big]\leq \min\left(2^\ell,AC\mathcal{S}_\delta\left(\ell,\frac{3B}{2A}\right)\right)+B, \tag{47}$$

$$\mathbb{E}\left[\left|T_F^{1/2}(\boldsymbol{s})\right||\mathcal{E}_l\right]\geq \frac{1}{2}\min\left(2^\ell,\frac{1}{2}AC\mathcal{S}_\delta\left(\ell,\frac{B}{4A}\right)\right). \tag{48}$$

Proof: The size of $T_F^1(\boldsymbol{s})$ equals the number of distinct strings among chunks $Z_c^b, 1\leq c\leq C+1, 1\leq b\leq B$. Clearly, chunks of length ℓ are δ-edit descendants of the AC source symbol substrings $U_c^a, 1\leq c\leq C, 1\leq a\leq A$, which are independent and uniformly distributed in Σ^ℓ. Given $\mathcal{E}_u$, each U_c^a has at most $\frac{3B}{2A}$ descendants. Moreover, since we assume that the source symbols $\mathsf{X}_1,\ldots,\mathsf{X}_A$ are chosen uniformly and independently, it follows directly from Lemma 9 that for any ℓ-string $\boldsymbol{w}$,

$$\Pr\big(\boldsymbol{w}\in T_F^1(\boldsymbol{s})|\mathcal{E}_u\big)\leq \min\left(1,AC\frac{\mathcal{S}_\delta\big(\ell,\frac{3B}{2A}\big)}{2^\ell}\right). \tag{422}$$

Hence

$$\begin{aligned}\mathbb{E}\big[\big|T_F^1(\boldsymbol{s})\big||\mathcal{E}_u\big]&\leq \sum_{\boldsymbol{w}\in\Sigma^\ell}\Pr\big(\boldsymbol{w}\in T_F^1(\boldsymbol{s})|\mathcal{E}_u\big)+B &(423)\\ &\leq \min\left(2^\ell,AC\mathcal{S}_\delta\left(\ell,\frac{3B}{2A}\right)\right)+B, &(424)\end{aligned}$$

where the addend B accounts for the chunks of lengths less than ℓ at the end of each source block, if any.

The lower bound on $\left|T_F^{1/2}(\boldsymbol{s})\right|$ given $\mathcal{E}_l$ follows similarly from Lemma 9. ■

3) Proofs of Lemma 19 and Lemma 20

Lemma 19. *Suppose K strings of length n are chosen independently and uniformly from Σ^n. Assume each string produces at least m_1 and at most m_2 descendants by δ-edits. For any string $\boldsymbol{w}$ with $|\boldsymbol{w}| \le n$, let $H_{\boldsymbol{w}}$ denote the event that $\boldsymbol{w}$ appears as a substring of one or more descendants. Then,*

$$\frac{1}{2}\min\left(1, \frac{1}{2}\left\lfloor\frac{n}{|\boldsymbol{w}|}\right\rfloor K\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_1)}{2^{|\boldsymbol{w}|}}\right) \le \Pr(H_{\boldsymbol{w}}) \le \min\left(1, (n-|\boldsymbol{w}|+1)K\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_2)}{2^{|\boldsymbol{w}|}}\right). \tag{92}$$

Proof: Let the K strings be denoted $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K$. We use $\mathcal{D}_i$ to denote the set of δ-edit descendants of $\boldsymbol{y}_i$. Let $H_{\boldsymbol{w}}(i,j)$ denote the event that $\boldsymbol{w} = \boldsymbol{x}_{j,|\boldsymbol{w}|}$ for some $\boldsymbol{x} \in \mathcal{D}_i$. Clearly,

$$H_{\boldsymbol{w}} = \cup_{i=1}^{K} \cup_{j=1}^{n-|\boldsymbol{w}|+1} H_{\boldsymbol{w}}(i,j). \tag{425}$$

Note that the strings $\{\boldsymbol{x}_{j,|\boldsymbol{w}|}\}_{\boldsymbol{x}\in\mathcal{D}_i}$ are iid δ-edit descendants of $(\boldsymbol{y}_i)_{j,|\boldsymbol{w}|}$. Hence by Lemma 8

$$\frac{1}{2}\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_1)}{2^{|\boldsymbol{w}|}} \le \frac{1}{2}\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, |\mathcal{D}_i|)}{2^{|\boldsymbol{w}|}} \le \Pr(H_{\boldsymbol{w}}(i,j)) \le \frac{\mathcal{S}_\delta(|\boldsymbol{w}|, |\mathcal{D}_i|)}{2^{|\boldsymbol{w}|}} \le \frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_2)}{2^{|\boldsymbol{w}|}}. \tag{426}$$

where the first and the last inequalities follow from $m_1 \le |\mathcal{D}_i| \le m_2$.

Applying the union bound on (425) gives

$$\Pr(H_{\boldsymbol{w}}) \le \cup_{i=1}^{K} \cup_{j=1}^{n-|\boldsymbol{w}|+1} \Pr(H_{\boldsymbol{w}}(i,j)) \le (n-|\boldsymbol{w}|+1)K\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_2)}{2^{|\boldsymbol{w}|}}. \tag{427}$$

The desired upper bound follows by noting that 1 is a trivial upper bound.

We next prove the lower bound. For each i, non-overlapping substrings of $\boldsymbol{r}_i$ are independent and so are their descendants. Hence, events $H_{\boldsymbol{w}}(i,j)$, $j = 1, 1+|\boldsymbol{w}|, \ldots, 1+(p-1)|\boldsymbol{w}|$, where $p = \left\lfloor\frac{n}{|\boldsymbol{w}|}\right\rfloor$, are mutually independent. It follows that

$$\begin{aligned}
&\Pr\big(\cup_{i=1}^{K} \cup_{a=1}^{p} H_{\boldsymbol{w}}(i, 1+(a-1)|\boldsymbol{w}|)\big) && (428)\\
&= 1 - \prod_{i=1}^{K}\prod_{a=1}^{p}(1 - \Pr(H_{\boldsymbol{w}}(i, 1+(a-1)|\boldsymbol{w}|))) && (429)\\
&\ge 1 - \left(1 - \frac{1}{2}\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_1)}{2^{|\boldsymbol{w}|}}\right)^{Kp} && (430)\\
&\ge \frac{1}{2}\min\left(1, \frac{1}{2}Kp\frac{\mathcal{S}_\delta(|\boldsymbol{w}|, m_1)}{2^{|\boldsymbol{w}|}}\right), && (431)
\end{aligned}$$

where the last inequality follows from inequality (1) that $1-(1-x)^n \ge \frac{1}{2}\min(1, nx)$ for $x \in (0,1)$ and integer n. The desired lower bound thus follows by noting that

$$\cup_{i=1}^{K} \cup_{a=1}^{p} H_{\boldsymbol{w}}(i, 1+(a-1)|\boldsymbol{w}|) \subseteq H_{\boldsymbol{w}}. \tag{432}$$

■

Lemma 20. *Consider the source model with edit probability δ. For any string $\boldsymbol{w} \in \Sigma^*$ with $|\boldsymbol{w}| \le 2L$,*

$$\Pr(\boldsymbol{w} \in Y_1^B | \mathcal{E}_u) \le \min\left(1, 2AL\frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{3B}{2A}\big)}{2^{|\boldsymbol{w}|}}\right). \tag{93}$$

For any string $\boldsymbol{w} \in \Sigma^*$ with $|\boldsymbol{w}| \le \lceil\frac{1}{2}L\rceil$,

$$\Pr\Big(\boldsymbol{w} \in Y_1^{B/2} | \mathcal{E}_l\Big) \ge \frac{1}{2}\min\left(1, \frac{AL}{8|\boldsymbol{w}|}\frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}}\right). \tag{94}$$

Proof: Recall that we assume every source symbol (and thus every source block) is of length at least $\frac{1}{2}L$ and at most $2L$. So we can get a lower bound on $\Pr\Big(\boldsymbol{w} \in Y_1^{B/2} | \mathcal{E}_l\Big)$ by assuming every source block is of length $\frac{L}{2}$. Similarly, we get an upper bound on $\Pr\big(\boldsymbol{w} \in Y_1^B | \mathcal{E}_u\big)$ by assuming every source block is of length $2L$.

Now that the B source blocks are independent and each is a δ-edit descendant of one of the A source symbols. Moreover, each random string (source symbol) has at most $\frac{3B}{2A}$ descendants given $\mathcal{E}_u$. Therefore, by directly applying Lemma 19,

$$\Pr\big(\boldsymbol{w} \in Y_1^B | \mathcal{E}_u\big) \le \min\left(1, (2L - |\boldsymbol{w}| + 1)A \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{3B}{2A}\big)}{2^{|\boldsymbol{w}|}}\right) \tag{433}$$

$$\le \min\left(1, 2LA \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{3B}{2A}\big)}{2^{|\boldsymbol{w}|}}\right). \tag{434}$$

The lower bound can be obtained similarly:

$$\Pr\Big(\boldsymbol{w} \in Y_1^{B/2} | \mathcal{E}_l\Big) \ge \frac{1}{2} \min\left(1, \frac{1}{2}\left\lfloor \frac{L/2}{|\boldsymbol{w}|} \right\rfloor A \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}}\right) \tag{435}$$

$$\ge \frac{1}{2} \min\left(1, \frac{1}{8} \frac{L}{|\boldsymbol{w}|} A \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}}\right). \tag{436}$$

■

4) Proof of Lemma 21

Lemma 21. *Consider the source model with edit probability $\delta < \frac{1}{2}$. For any n such that $\frac{\log(B/A)-2}{H(\delta)} \le n + 2M + 2 \le \frac{L}{4}$,*

$$\sum_{\boldsymbol{u} \in R_M^n} \Pr\Big(10^M \boldsymbol{u} 10^M \in Y_1^{B/2} | \mathcal{E}_l\Big) \ge \frac{BL}{2^7 \cdot 2^{2M+2}} \cdot \left(1 - \frac{1}{2^{M-1}}\right)^n - \frac{3B^2L^2}{2^{n+2M+2}}. \tag{95}$$

Proof: Let $\boldsymbol{w} = 10^M \boldsymbol{u} 10^M$. By assumption, $|\boldsymbol{w}| = |\boldsymbol{u}| + 2M + 2 \ge \frac{\log(B/A)-2}{H(\delta)}$.

For definiteness, we assume $\big|Y_{1/2}(a)\big| = \frac{B}{4A}$ for all a and all source symbols are of length $\frac{L}{2}$. With these assumptions, we have a similar setting to that in Lemma 19. So we adopt the same notation. Let $H_{\boldsymbol{w}}$ denote $\boldsymbol{w} \in Y_1^{B/2}$ and $H_{\boldsymbol{w}}(a, j)$ denote the event that $\boldsymbol{w} = \boldsymbol{x}_{j,|\boldsymbol{w}|}$ for some $\boldsymbol{x} \in Y_{1/2}(a)$. Similar to (425):

$$H_{\boldsymbol{w}} = \cup_{a=1}^{A} \cup_{j=1}^{\lceil L/2 \rceil - |\boldsymbol{w}| + 1} H_{\boldsymbol{w}}(a, j). \tag{437}$$

Moreover,

$$\frac{1}{2} \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}} \le \Pr(H_{\boldsymbol{w}}(a, j)) \le \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}}. \tag{438}$$

In Lemma 20, an upper bound on $\Pr\Big(\boldsymbol{w} \in Y_1^{B/2} | \mathcal{E}_l\Big)$ is obtained by applying the union bound on (437). Here, we get a lower bound by the inclusion-exclusion principle:

$$\Pr(H_{\boldsymbol{w}}) \ge \sum_{a=1}^{A} \sum_{i=1}^{\lceil L/2 \rceil - |\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a, i)) \tag{439}$$

$$- \sum_{1 \le a_1 \ne a_2 \le A} \sum_{j=1}^{\lceil L/2 \rceil - |\boldsymbol{w}|} \sum_{k=1}^{\lceil L/2 \rceil - |\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a_1, j) \cap H_{\boldsymbol{w}}(a_2, k)) \tag{440}$$

$$- \sum_{a=1}^{A} \sum_{\substack{1 \le j,k \le \lceil L/2 \rceil - |\boldsymbol{w}| \\ j \ne k}} \Pr(H_{\boldsymbol{w}}(a, j) \cap H_{\boldsymbol{w}}(a, k)). \tag{441}$$

We compute the three terms on the right-hand side of the inequality above as follows.

For the term in (439), since $|\boldsymbol{w}| \ge \frac{\log(B/A)-2}{H(\delta)}$,

$$\Pr(H_{\boldsymbol{w}}(a, i)) \ge \frac{1}{2} \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \lceil \frac{B}{4A} \rceil\big)}{2^{|\boldsymbol{w}|}} \ge \frac{1}{2} \frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|, \frac{B}{4A}\big)}{2^{|\boldsymbol{w}|}} \ge \frac{B}{32A \cdot 2^{|\boldsymbol{w}|}}, \tag{442}$$

where the last inequality follows from (37). It follows that

$$\sum_{a=1}^{A} \sum_{i=1}^{\lceil L/2 \rceil - |\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a, i)) \ge A(\lceil L/2 \rceil - |\boldsymbol{w}|) \frac{B}{32A \cdot 2^{|\boldsymbol{w}|}} \ge \frac{BL}{2^7 \cdot 2^{|\boldsymbol{w}|}}. \tag{443}$$

For the term in (440), since for all $a_1 \neq a_2$, $\boldsymbol{y}_{a_1}$ and $\boldsymbol{y}_{a_2}$ are independent and so are their descendants, we get

$$\begin{aligned}
&\sum_{1\le a_1\neq a_2\le A} \sum_{j=1}^{\lceil L/2\rceil-|\boldsymbol{w}|} \sum_{k=1}^{\lceil L/2\rceil-|\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a_1,j)\cap H_{\boldsymbol{w}}(a_2,k)) && (444)\\
&= \sum_{1\le a_1\neq a_2\le A} \sum_{j=1}^{\lceil L/2\rceil-|\boldsymbol{w}|} \sum_{k=1}^{\lceil L/2\rceil-|\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a_1,j))\Pr(H_{\boldsymbol{w}}(a_2,k)) && (445)\\
&\le \sum_{1\le a_1\neq a_2\le A} \sum_{j=1}^{\lceil L/2\rceil-|\boldsymbol{w}|} \sum_{k=1}^{\lceil L/2\rceil-|\boldsymbol{w}|} \left(\frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|,\lceil\frac{B}{4A}\rceil\big)}{2^{|\boldsymbol{w}|}}\right)^2 && (446)\\
&\le \sum_{1\le a_1\neq a_1\le A} \frac{B^2L^2}{A^2 2^{2|\boldsymbol{w}|}} && (447)\\
&\le \frac{B^2L^2}{2^{2|\boldsymbol{w}|}}, && (448)
\end{aligned}$$

where the second inequality follows from (43) that $\mathcal{S}_\delta(\ell,m)\le m$ and the inequalities $\lceil L/2\rceil - |\boldsymbol{w}| \le L$, $\lceil\frac{B}{4A}\rceil \le \frac{B}{A}$.

We then consider the term in (441), where the two occurrences of $\boldsymbol{w}$ are among the descendants of a single source symbol, and thus might not be independent. Unlike the previous two terms, we consider lower bounding the sum of probabilities $\Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k))$ over all $\boldsymbol{w}$ of the form $10^M\boldsymbol{u}10^M, \boldsymbol{u}\in R_M^n$. For clarity of presentation, we first claim (to be proved later) that for any a,

$$\sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \sum_{\substack{1\le j,k\le\lceil L/2\rceil-|\boldsymbol{w}|\\ j\neq k}} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \le \frac{B^2L^2}{A^2 2^{|\boldsymbol{w}|}}\left(1+\frac{n+M+1}{L}\right). \tag{449}$$

It follows that

$$\sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \sum_{a=1}^{A} \sum_{\substack{1\le j,k\le\lceil L/2\rceil-|\boldsymbol{w}|\\ j\neq k}} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \le \frac{B^2L^2}{A 2^{|\boldsymbol{w}|}}\left(1+\frac{n+M+1}{L}\right). \tag{450}$$

Thus, combining (441), (443), (448) and (450) gives

$$\begin{aligned}
\sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \Pr(H_{\boldsymbol{w}}) &\ge \sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \left(\frac{BL}{2^7\cdot 2^{|\boldsymbol{w}|}} - \frac{B^2L^2}{2^{2|\boldsymbol{w}|}}\right) - \frac{B^2L^2}{A2^{|\boldsymbol{w}|}}\left(1+\frac{n+M+1}{L}\right) && (451)\\
&\ge \frac{BL}{2^7\cdot 2^{|\boldsymbol{w}|}}\cdot|R_M^n| - \frac{B^2L^2}{2^{2|\boldsymbol{w}|}}\cdot|R_M^n| - \frac{B^2L^2}{A2^{|\boldsymbol{w}|}}\left(1+\frac{n+M+1}{L}\right) && (452)\\
&\ge \frac{BL}{2^7\cdot 2^{|\boldsymbol{w}|}}\cdot|R_M^n| - \frac{3B^2L^2}{2^{|\boldsymbol{w}|}}, && (453)
\end{aligned}$$

where the last inequality follows from $|R_M^n|\le 2^{|\boldsymbol{w}|}$, $\frac{n+M+1}{L}\le 1$ and $A\ge 1$. The desired lower bound thus follows from bounding $|R_M^n|$ by Lemma 1.

Finally, we prove inequality (449). Fix a. That $H_{\boldsymbol{w}}(a,j)$ and $H_{\boldsymbol{w}}(a,k)$ both hold means there exist descendants $\boldsymbol{x}_1,\boldsymbol{x}_2$ (possibly the same one) of $\boldsymbol{y}_a$ such that $(\boldsymbol{x}_1)_{j,|\boldsymbol{w}|} = (\boldsymbol{x}_2)_{k,|\boldsymbol{w}|} = \boldsymbol{w}$. Assume $j<k$ without loss of generality. We compute $\Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k))$ for different values of (j,k):

- $|j-k|\ge|\boldsymbol{w}|$. The two occurrences of $\boldsymbol{w}$ in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are plotted in Figure 17. In this case, they are produced by two non-overlapping substrings of $\boldsymbol{y}_a$ and thus are independent. It follows that

$$\begin{aligned}
\sum_{|j-k|\ge|\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) &= \sum_{|j-k|\ge|\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a,j))\Pr(H_{\boldsymbol{w}}(a,k)) && (454)\\
&\le L^2\left(\frac{\mathcal{S}_\delta\big(|\boldsymbol{w}|,\lceil\frac{B}{4A}\rceil\big)}{2^{|\boldsymbol{w}|}}\right)^2 && (455)\\
&\le \frac{L^2B^2}{A^2 2^{2|\boldsymbol{w}|}}. && (456)
\end{aligned}$$

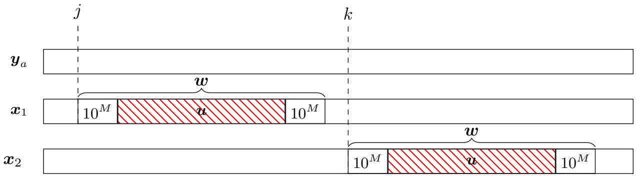


Figure 17: Relative position of the two occurrences of $\boldsymbol{w}$ at position j and k when $|j-k| \geq |\boldsymbol{w}|$.

- $1 \leq |j-k| < |\boldsymbol{u}|$. The two occurrences of $\boldsymbol{w}$ in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are plotted in Figure 18. In this case, the two occurrences of $\boldsymbol{w}$ are descendants of two overlapping substrings of $\boldsymbol{y}_a$. Recall that $\boldsymbol{w} = 10^M \boldsymbol{u} 10^M$. We write the string $\boldsymbol{u}$ in $\boldsymbol{x}_1$ as $\boldsymbol{u}_1\boldsymbol{u}_2$, and write the string $\boldsymbol{u}$ in $\boldsymbol{x}_2$ as $\boldsymbol{u}_2'\boldsymbol{u}_3$, so that $\boldsymbol{u}_2$ and $\boldsymbol{u}_2'$ have the same ancestors, denoted $\boldsymbol{r}_2$. Denote the ancestor of $\boldsymbol{u}_1$ and $\boldsymbol{u}_3$ by $\boldsymbol{r}_1$ and $\boldsymbol{r}_3$, respectively. Denote the ancestor of 10^M at the beginning of $\boldsymbol{w}$ in $\boldsymbol{x}_1$ by $\boldsymbol{r}_0$, and the ancestor of 10^M at the end of $\boldsymbol{w}$ in $\boldsymbol{x}_2$ by $\boldsymbol{r}_4$. We have $|\boldsymbol{r}_1| = |\boldsymbol{u}_1| = |\boldsymbol{r}_3| = |\boldsymbol{u}_3| = k-j$, $|\boldsymbol{r}_2| = |\boldsymbol{u}_2| = |\boldsymbol{u}_2'| = |\boldsymbol{u}| - (k-j)$. Write $\boldsymbol{r} = \boldsymbol{r}_0\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{r}_3\boldsymbol{r}_4$.

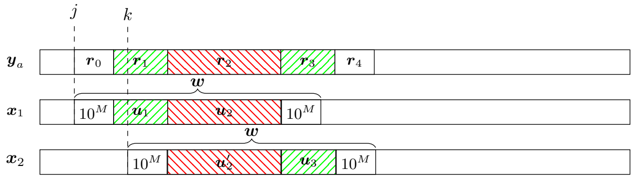


Figure 18: Relative position of the two occurrences of $\boldsymbol{w}$ at position j and k when $1 \leq |j-k| < |\boldsymbol{u}|$.

For a single descendant $\boldsymbol{x}$ of $\boldsymbol{y}_a$, $\boldsymbol{x}$ can not have $\boldsymbol{w}$ as substrings at positions j and k simultaneously since $\boldsymbol{u}$ is M-RLL. In other words, either exactly one of $\boldsymbol{x}_{j,|\boldsymbol{w}|}$ and $\boldsymbol{x}_{k,|\boldsymbol{w}|}$ equals $\boldsymbol{w}$ or none of them does. So given $\boldsymbol{r}$, we can get an upper bound on the probability of $H_{\boldsymbol{w}}(a,j) \cap H_{\boldsymbol{w}}(a,k)$ by assuming they are independent, i.e.,

$$\Pr(H_{\boldsymbol{w}}(a,j) \cap H_{\boldsymbol{w}}(a,k)|\boldsymbol{r}) \leq \Pr(H_{\boldsymbol{w}}(a,j)|\boldsymbol{r}) \Pr(H_{\boldsymbol{w}}(a,k)|\boldsymbol{r}). \tag{457}$$

We prove (457) rigorously by Lemma 60 at the end of this section.

Denote the Hamming distance between $\boldsymbol{r}_0$ and 10^M by d_0, $\boldsymbol{r}_1$ and $\boldsymbol{u}_1$ by d_1, $\boldsymbol{r}_2$ and $\boldsymbol{u}_2$ by d_2, $\boldsymbol{r}_2$ and $\boldsymbol{u}_2'$ by d_2', $\boldsymbol{r}_3$ and $\boldsymbol{u}_3$ by d_3, and $\boldsymbol{r}_4$ and 10^M by d_4. Let $\boldsymbol{w}_l = 10^M\boldsymbol{u}$ and $\boldsymbol{w}_r = \boldsymbol{u}10^M$. The probability of occurrences increases if we only consider substrings $\boldsymbol{w}_l$ or $\boldsymbol{w}_r$. We have

$$\begin{aligned}\Pr(H_{\boldsymbol{w}}(a,j)|\boldsymbol{r}) &\leq \Pr(H_{\boldsymbol{w}_l}(a,j)|\boldsymbol{r}) && (458)\\ &= 1 - \left(1 - \delta^{d_0+d_1+d_2}(1-\delta)^{|\boldsymbol{w}|-M-1-(d_0+d_1+d_2)}\right)^{\left\lceil \frac{B}{4A} \right\rceil} && (459)\\ &\leq \frac{B}{A}\delta^{d_0+d_1+d_2}(1-\delta)^{|\boldsymbol{w}|-M-1-(d_0+d_1+d_2)}, && (460)\end{aligned}$$

and

$$\begin{aligned}\Pr(H_{\boldsymbol{w}}(a,k)|\boldsymbol{r}) &\leq \Pr(H_{\boldsymbol{w}_r}(a,k+M+1)|\boldsymbol{r}) && (461)\\ &= 1 - \left(1 - \delta^{d_2'+d_3+d_4}(1-\delta)^{|\boldsymbol{w}|-M-1-\left(d_2'+d_3+d_4\right)}\right)^{\left\lceil \frac{B}{4A} \right\rceil} && (462)\\ &\leq \frac{B}{A}\delta^{d_2'+d_3+d_4}(1-\delta)^{|\boldsymbol{w}|-M-1-\left(d_2'+d_3+d_4\right)}. && (463)\end{aligned}$$

It follows from (457) that $\Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k))$ is less than or equal to

$$\sum_{\boldsymbol{r}\in\Sigma^{|\boldsymbol{w}|}} \Pr(\boldsymbol{r})\Pr(H_{\boldsymbol{w}}(a,j)|\boldsymbol{r})\Pr(H_{\boldsymbol{w}}(a,k)|\boldsymbol{r}) \tag{464}$$

$$=\left(\frac{B}{A}\right)^2\cdot\left(\sum_{\boldsymbol{r}_0\in\Sigma^{M+1}}\frac{1}{2^{|\boldsymbol{r}_0|}}\delta^{d_0}(1-\delta)^{|\boldsymbol{r}_0|-d_0}\right) \tag{465}$$

$$\cdot\left(\sum_{\boldsymbol{r}_1\in\Sigma^{k-j}}\frac{1}{2^{|\boldsymbol{r}_1|}}\delta^{d_1}(1-\delta)^{|\boldsymbol{r}_1|-d_1}\right) \tag{466}$$

$$\cdot\left(\sum_{\boldsymbol{r}_3\in\Sigma^{k-j}}\frac{1}{2^{|\boldsymbol{r}_3|}}\delta^{d_3}(1-\delta)^{|\boldsymbol{r}_3|-d_3}\right) \tag{467}$$

$$\cdot\left(\sum_{\boldsymbol{r}_4\in\Sigma^{M+1}}\frac{1}{2^{|\boldsymbol{r}_4|}}\delta^{d_4}(1-\delta)^{|\boldsymbol{r}_4|-d_4}\right) \tag{468}$$

$$\cdot\left(\sum_{\boldsymbol{r}_2\in\Sigma^{|\boldsymbol{u}|-(k-j)}}\frac{1}{2^{|\boldsymbol{r}_2|}}\delta^{d_2+d_2'}(1-\delta)^{2|\boldsymbol{r}_2|-(d_2+d_2')}\right) \tag{469}$$

$$=\left(\frac{B}{A}\right)^2\cdot\frac{1}{2^{2M+2+2(k-j)}} \tag{470}$$

$$\cdot\left(\sum_{\boldsymbol{r}_2\in\Sigma^{|\boldsymbol{u}|-(k-j)}}\frac{1}{2^{|\boldsymbol{r}_2|}}\delta^{d_2+d_2'}(1-\delta)^{2|\boldsymbol{r}_2|-(d_2+d_2')}\right). \tag{471}$$

Let d° denote the Hamming distance between $\boldsymbol{u}_2$ and $\boldsymbol{u}_2'$. Among the $|\boldsymbol{u}_2|-d^\circ$ positions where $\boldsymbol{u}_2$ and $\boldsymbol{u}_2'$ are the same, suppose $\boldsymbol{u}_2$ differs from $\boldsymbol{r}_2$ in v of them. Among the d° positions where $\boldsymbol{u}_2$ differs from $\boldsymbol{u}_2'$, suppose $\boldsymbol{u}_2$ differs from $\boldsymbol{r}_2$ in t of them. It follows that $d_2=v+t$ and $d_2'=d^\circ+v-t$. Thus, we further have

$$\sum_{\boldsymbol{r}_2\in\Sigma^{|\boldsymbol{u}|-(k-j)}}\frac{1}{2^{|\boldsymbol{r}_2|}}\delta^{d_2+d_2'}(1-\delta)^{2|\boldsymbol{r}_2|-(d_2+d_2')}=\sum_{\boldsymbol{r}_2\in\Sigma^{|\boldsymbol{u}|-(k-j)}}\frac{1}{2^{|\boldsymbol{r}_2|}}\delta^{2v+d^\circ}(1-\delta)^{2|\boldsymbol{r}_2|-2v-d^\circ} \tag{472}$$

$$=\sum_{v=0}^{|\boldsymbol{r}_2|-d^\circ}\binom{|\boldsymbol{r}_2|-d^\circ}{v}\frac{2^{d^\circ}}{2^{|\boldsymbol{r}_2|}}\delta^{2v+d^\circ}(1-\delta)^{2|\boldsymbol{r}_2|-2v-d^\circ} \tag{473}$$

$$=\frac{(2\delta(1-\delta))^{d^\circ}}{2^{|\boldsymbol{r}_2|}}\sum_{v=0}^{|\boldsymbol{r}_2|-d^\circ}\binom{|\boldsymbol{r}_2|-d^\circ}{v}\left(\delta^2\right)^v\left((1-\delta)^2\right)^{(|\boldsymbol{r}_2|-d^\circ)-v} \tag{474}$$

$$=\frac{1}{2^{|\boldsymbol{r}_2|}}(2\delta(1-\delta))^{d^\circ}\left(\delta^2+(1-\delta)^2\right)^{|\boldsymbol{r}_2|-d^\circ}. \tag{475}$$

Since $|\boldsymbol{r}_2|=|\boldsymbol{w}|-(k-j)$,

$$\Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k))\le\left(\frac{B}{A}\right)^2\frac{(2\delta(1-\delta))^{d^\circ}\left(\delta^2+(1-\delta)^2\right)^{|\boldsymbol{r}_2|-d^\circ}}{2^{|\boldsymbol{w}|+(k-j)}}. \tag{476}$$

Note that $\boldsymbol{u}_2$ is the $|\boldsymbol{r}_2|$-suffix of $\boldsymbol{u}$ and $\boldsymbol{u}_2'$ is the $|\boldsymbol{r}_2|$ prefix of $\boldsymbol{u}$. With $|\boldsymbol{u}|=n$, the number of n-strings whose $|\boldsymbol{r}_2|$-suffix and $|\boldsymbol{r}_2|$-prefix are at Hamming distance d° is $2^{n-|\boldsymbol{r}_2|}\binom{|\boldsymbol{r}_2|}{d^\circ}$ since an n-string can be uniquely determined by its $|\boldsymbol{r}_2|$-prefix and the mismatches. Therefore,

$$\sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \sum_{1\le|j-k|<|\boldsymbol{u}|} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \tag{477}$$

$$\le \sum_{1\le|j-k|<|\boldsymbol{u}|} \sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in\Sigma^n}} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \tag{478}$$

$$\le L|\boldsymbol{u}|\cdot\sum_{d^\circ=0}^{|\boldsymbol{r}_2|} 2^{|\boldsymbol{r}_1|}\binom{|\boldsymbol{r}_2|}{d^\circ}\cdot\left(\frac{B}{A}\right)^2 \tag{479}$$

$$\cdot\frac{(2\delta(1-\delta))^{d^\circ}\left(\delta^2+(1-\delta)^2\right)^{|\boldsymbol{r}_2|-d^\circ}}{2^{|\boldsymbol{w}|+k-j}} \tag{480}$$

$$=\left(\frac{B}{A}\right)^2\frac{Ln}{2^{|\boldsymbol{w}|}}. \tag{481}$$

- $|\boldsymbol{u}|\le|j-k|<|\boldsymbol{w}|$. The two occurrences of $\boldsymbol{w}$ in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are plotted in Figure 19.

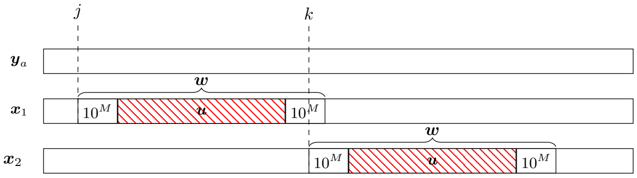


Figure 19: Relative position of the two occurrences of $\boldsymbol{w}$ at position j and k when $|\boldsymbol{u}|\le|j-k|<|\boldsymbol{w}|$.

It can be seen that the prefix $10^M\boldsymbol{u}$ of $\boldsymbol{w}$ in $\boldsymbol{x}_1$ and $\boldsymbol{w}$ in $\boldsymbol{x}_2$ are descendants of non-overlapping substrings of $\boldsymbol{y}_a$ and thus independent. We can write

$$\Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k))\le\Pr(H_{\boldsymbol{w}_l}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \tag{482}$$

$$=\Pr(H_{\boldsymbol{w}_l}(a,j))\Pr(H_{\boldsymbol{w}}(a,k)) \tag{483}$$

$$\le\frac{\mathcal{S}_\delta\left(|\boldsymbol{w}_l|,\left\lceil\frac{B}{4A}\right\rceil\right)\mathcal{S}_\delta\left(|\boldsymbol{w}|,\left\lceil\frac{B}{4A}\right\rceil\right)}{2^{|\boldsymbol{w}_l|+|\boldsymbol{w}|}} \tag{484}$$

$$\le\left(\frac{B}{A}\right)^2\frac{1}{2^{2|\boldsymbol{w}|-M-1}}. \tag{485}$$

It follows that

$$\sum_{|\boldsymbol{u}|\le|j-k|<|\boldsymbol{w}|}\Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k))\le\left(\frac{B}{A}\right)^2\frac{L\cdot2(M+1)}{2^{2|\boldsymbol{w}|-M-1}}. \tag{486}$$

Thus, combining (456), (481), (486) gives

$$\sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \sum_{\substack{1\le j,k\le\lceil L/2\rceil-|\boldsymbol{w}|\\ j\neq k}} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \tag{487}$$

$$\le \sum_{\substack{\boldsymbol{w}:\boldsymbol{w}=10^M\boldsymbol{u}10^M\\ \boldsymbol{u}\in R_M^n}} \Bigg(\sum_{|j-k|\ge|\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \tag{488}$$

$$+ \sum_{1\le|j-k|<|\boldsymbol{u}|} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \tag{489}$$

$$+ \sum_{|\boldsymbol{u}|\le|j-k|\le|\boldsymbol{w}|} \Pr(H_{\boldsymbol{w}}(a,j)\cap H_{\boldsymbol{w}}(a,k)) \Bigg) \tag{490}$$

$$\le \frac{L^2B^2}{A^2\cdot 2^{2|\boldsymbol{w}|}}\cdot|R_M^n| + \frac{B^2Ln}{A^2\cdot 2^{|\boldsymbol{w}|}} + \frac{2B^2L(M+1)}{A^2\cdot 2^{2|\boldsymbol{w}|-M-1}}\cdot|R_M^n| \tag{491}$$

$$\le \frac{B^2L^2}{A^2 2^{|\boldsymbol{w}|}}\bigg(1+\frac{n+M+1}{L}\bigg). \tag{492}$$

■

We present a lemma from which inequality (457) follows directly.

Lemma 60. *Let $\boldsymbol{r}$ be any string of length n with m iid δ-edit descendants. For a string $\boldsymbol{v}, |\boldsymbol{v}| < n$ and $1 \le j < k \le n-|\boldsymbol{v}|+1$, let $\mathcal{J}(\boldsymbol{v}), \mathcal{K}(\boldsymbol{v})$ denote the events that there exists a descendant of $\boldsymbol{r}$ whose j-th, k-th $|\boldsymbol{v}|$-substring equal $\boldsymbol{v}$, respectively. We have*

$$\Pr(\mathcal{J}(\boldsymbol{v})\cap\mathcal{K}(\boldsymbol{v})) \le \Pr(\mathcal{J}(\boldsymbol{v}))\Pr(\mathcal{K}(\boldsymbol{v})), \tag{493}$$

if the $(|\boldsymbol{v}|-(k-j))$-suffix and $(|\boldsymbol{v}|-(k-j))$-prefix of $\boldsymbol{v}$ are not the same.

Proof: If the $(|\boldsymbol{v}|-(k-j))$-suffix and $(|\boldsymbol{v}|-(k-j))$-prefix of $\boldsymbol{v}$ are not the same, then in any descendant $\boldsymbol{x}$, $\boldsymbol{v}$ can not be both the j-th and the k-th substring. Therefore, in $\boldsymbol{x}$, exactly one of the following three mutually exclusive events holds: i) $\boldsymbol{x}_{j,|\boldsymbol{v}|} = \boldsymbol{v}$, ii) $\boldsymbol{x}_{k,|\boldsymbol{v}|} = \boldsymbol{v}$, iii) $\boldsymbol{x}_{j,|\boldsymbol{v}|} \neq \boldsymbol{v}$ and $\boldsymbol{x}_{k,|\boldsymbol{v}|} \neq \boldsymbol{v}$. Let p_j denote the probability of $\boldsymbol{x}_{j,|\boldsymbol{v}|} = \boldsymbol{v}$ and p_k denote the probability of $\boldsymbol{x}_{k,|\boldsymbol{v}|} = \boldsymbol{v}$. We have

$$\Pr\big(\boldsymbol{x}_{j,|\boldsymbol{v}|}\neq\boldsymbol{v}\cap\boldsymbol{x}_{k,|\boldsymbol{v}|}\neq\boldsymbol{v}\big) = 1-p_j-p_k. \tag{494}$$

Therefore, among the m iid descendants of $\boldsymbol{r}$,

$$\Pr(\mathcal{J}(\boldsymbol{v})\cap\mathcal{K}(\boldsymbol{v})) = \Pr(\mathcal{J}(\boldsymbol{v})) + \Pr(\mathcal{K}(\boldsymbol{v})) + \Pr\big(\bar{\mathcal{J}}(\boldsymbol{v})\cap\bar{\mathcal{K}}(\boldsymbol{v})\big) - 1 \tag{495}$$

$$= (1-(1-p_j)^m) + (1-(1-p_k)^m) \tag{496}$$

$$+ (1-p_j-p_k)^m - 1 \tag{497}$$

$$= 1-(1-p_j)^m-(1-p_k)^m+(1-p_j-p_k)^m. \tag{498}$$

On the other hand,

$$\Pr(\mathcal{J}(\boldsymbol{v}))\Pr(\mathcal{K}(\boldsymbol{v})) = (1-(1-p_j)^m)(1-(1-p_k)^m) \tag{499}$$

$$= 1-(1-p_j)^m-(1-p_k)^m+(1-p_j)^m(1-p_k)^m. \tag{500}$$

The desired inequality thus follows by noting that $1-p_j-p_k \le (1-p_j)(1-p_k)$. ■

Inequality (457) can be obtained by replacing $\mathcal{J}(\boldsymbol{v})$ and $\mathcal{K}(\boldsymbol{v})$ with $H_{\boldsymbol{w}}(a,j)$ and $H_{\boldsymbol{w}}(a,k)$, respectively.

5) Proofs of Lemma 22 and Lemma 26

Lemma 22. *Consider the source string $\boldsymbol{s} = Y_1Y_2\ldots Y_B$. When $2^M = o(L)$, for B, L sufficiently large,*

$$\Pr\bigg(C_{VL}^M(\boldsymbol{s}) \ge \frac{1}{4}\cdot\left\lfloor\frac{B}{2}\right\rfloor\bigg(\frac{L}{2^{M+8}}-1\bigg)\bigg) \ge \frac{5}{6}. \tag{96}$$

Proof: Equally parse each of $Y_{\lceil B/2\rceil+1},\ldots,Y_B$ into segments of length 2^{M+7}. So that every Y_b contains $\left\lfloor \frac{|Y_b|}{2^{M+7}} \right\rfloor$ segments. We show that among these $\sum_{b=\left\lfloor \frac{B}{2}\right\rfloor+1}^{B} \left\lfloor \frac{|Y_b|}{2^{M+7}} \right\rfloor$ segments, a constant fraction of them contain a chunk of length over 2^{M-4}.

Pick an arbitrary segment, denoted $\boldsymbol{z}$. Consider the two halves of $\boldsymbol{z}$. The second half of $\boldsymbol{z}$, which is of length 2^{M+6}, is by itself a Bernoulli(1/2) process going forward. We study the first time a run of M 0's appears in this process. By the union bound, with probability at least $1-\frac{2^{M-5}}{2^M}$, there exist no runs of M 0s in the first 2^{M-5} bits. Moreover, the average position of the end of the first run of M 0s in a Bernoulli(1/2) process is $2^{M+1}-2$ [94]. Therefore, by Markov's inequality, with probability at least $1-\frac{2^{M+1}-2}{2^{M+6}}$, there is a 0^M within the first 2^{M+6} bits. So the first time we see 0^M is after 2^{M-5} bits and before 2^{M+6} bits (i.e., the first 0^M is within the last $2^{M+6}-2^{M-5}$ bits) with probability at least

$$1-\frac{2^{M-5}}{2^M}-\frac{2^{M+1}-2}{2^{M+6}} \geq 1-\frac{1}{2^4}.$$

Similarly, the first half of $\boldsymbol{z}$ can be regarded as a reversed Bernoulli(1/2) process. So we also have with probability at least $1-\frac{1}{2^4}$, the first 0^M (counting backwards) is within the first $2^{M+6}-2^{M-5}$ bits. Clearly, a chunk exists between these two occurrences of 0^M. So with probability at least $1-\frac{1}{2^3}$, $\boldsymbol{z}$ contains a chunk of length at least 2^{M-4}. Since this property holds for all such segments of length 2^{M+7}, by the Markov inequality, with probability at least $1-\frac{1}{6}$, at least $\frac{1}{4}$ of the segments in $Y_{\lceil B/2\rceil+1}\cdots Y_B$ contain a chunk of length at least 2^{M-4}. The desired result is derived by noting $|Y_b| \geq L/2$. ■

Lemma 26. *Consider the source string $\boldsymbol{s}=Y_1Y_2\cdots Y_B$, with each Y_b being a descendant of source symbol X_{J_b}. For any integer h and any pairs of integers $(b_1,b_2),(i_1,i_2)$, the probability of Y_{b_1} and Y_{b_2} having identical substrings of length h starting at positions i_1 and i_2, respectively, is*

$$\Pr\Big((Y_{b_1})_{i_1,h}=(Y_{b_2})_{i_2,h}\Big)=\frac{1}{2^h}, \tag{116}$$

if $J_{b_1}\neq J_{b_2}$ or $i_1\neq i_2$.

Proof: We compute $\Pr((Y_{b_1})_{i_1,h}=(Y_{b_2})_{i_2,h})$ as $(b_1,b_2),(i_1,i_2)$ take different values in the following three cases:

- $J_{b_1}\neq J_{b_2}$ or $|i_1-i_2|\geq h$. If $J_{b_1}\neq J_{b_2}$, then Y_{b_1} and Y_{b_2} have different ancestors and are thus independent. It follows that their substrings are also independent. If $|i_1-i_2|\geq h$, then $(Y_{b_1})_{i_1,h}$ and $(Y_{b_2})_{i_2,h}$ are descendants of non-overlapping substrings of the source alphabet and are thus also independent. The desired result follows from the fact that $(Y_{b_1})_{i_1,h}$ and $(Y_{b_2})_{i_2,h}$ are both Bernoulli(1/2) processes by themselves.
- $b_1=b_2, |i_1-i_2|<h$. In this case, $(Y_{b_1})_{i_1,h}$ and $(Y_{b_2})_{i_2,h}$ are overlapping substrings of a single source block. Again, Y_{b_1} is Bernoulli(1/2) by itself. So the probability of $(Y_{b_1})_{i_1,h}=(Y_{b_2})_{i_2,h}$ is the same as that when $(Y_{b_1})_{i_1,h}$ and $(Y_{b_2})_{i_2,h}$ are independent.
- $J_{b_1}=J_{b_2}, b_1\neq b_2, |i_1-i_2|<h$. Let $J_{b_1}=J_{b_2}=a$. Assume $i_1<i_2$ without loss of generality. In this case, $(Y_{b_1})_{i_1,h}$ and $(Y_{b_2})_{i_2,h}$ are two independent δ-edit descendants of $(\mathsf{X}_a)_{i_1,h}$ and $(\mathsf{X}_a)_{i_2,h}$, respectively. So $\Pr((Y_{b_1})_{i_1,h}=(Y_{b_2})_{i_2,h})$ is uniquely determined by the Hamming distance between $(\mathsf{X}_a)_{i_1,h}$ and $(\mathsf{X}_a)_{i_2,h}$. Moreover, the distribution of the Hamming distance between $(\mathsf{X}_a)_{i_1,h}$ and $(\mathsf{X}_a)_{i_2,h}$ is the same as the distribution of the Hamming distance between two independent Bernoulli(1/2) process of length h. Therefore, we can assume $(Y_{b_1})_{i_1,h}$ and $(Y_{b_2})_{i_2,h}$ are independent and thus $\Pr((Y_{b_1})_{i_1,h}=(Y_{b_2})_{i_2,h})=\frac{1}{2^h}$.

■

6) Summations

For integers $b\geq a$ and $\beta>1$, summations of the forms $\sum_{n=a}^{b}\Big(1-\frac{1}{\beta}\Big)^n$ and $\sum_{n=a}^{b} n\Big(1-\frac{1}{\beta}\Big)^n$ appear in the proofs of Theorem 25 and Theorem 29. Let $x=1-\frac{1}{\beta}$. The limits of these sums in a certain asymptotic regime is discussed below.

1) Asymptotic behavior of $\sum_{n=a}^{b} x^n$:

We have

$$\sum_{n=a}^{b} x^n = \frac{x^a\big(1-x^{b-a+1}\big)}{1-x} \tag{501}$$

$$=\beta\Big(1-\frac{1}{\beta}\Big)^a\left(1-\Big(1-\frac{1}{\beta}\Big)^{b-a+1}\right). \tag{502}$$

If $b - a = \omega(\beta)$, then as $\beta \to \infty$,

$$\left(1 - \frac{1}{\beta}\right)^{b-a+1} = \left(\left(1 - \frac{1}{\beta}\right)^{\beta}\right)^{\frac{b-a+1}{\beta}} = o(1). \tag{503}$$

It follows that

$$\sum_{n=a}^{b} \left(1 - \frac{1}{\beta}\right)^{n} = \beta\left(1 - \frac{1}{\beta}\right)^{a}(1 + o(1)) \tag{504}$$

$$= \beta e^{-a/\beta}(1 + o(1)). \tag{505}$$

2) Asymptotic behavior of $\sum_{n=a}^{b} nx^n$:
We have

$$\sum_{n=a}^{b} nx^n = x\sum_{n=a}^{b} nx^{n-1} = x\left(\sum_{n=a}^{b} x^n\right)' \tag{506}$$

$$= x\left(\frac{x^a\left(1 - x^{b-a+1}\right)}{1 - x}\right)' \tag{507}$$

$$= x\frac{\left(ax^{a-1} - (b+1)x^b\right)(1 - x) + \left(x^a - x^{b+1}\right)}{(1 - x)^2} \tag{508}$$

$$= \beta^2\left(\left(\frac{a-1}{\beta} + 1\right)\left(1 - \frac{1}{\beta}\right)^{a} + \left(\frac{b}{\beta} + 1\right)\left(1 - \frac{1}{\beta}\right)^{b+1}\right). \tag{509}$$

If $\frac{b}{\beta} = \omega(1)$, then as $\beta \to \infty$,

$$\left(\frac{b}{\beta} + 1\right)\left(1 - \frac{1}{\beta}\right)^{b+1} = \left(\frac{b}{\beta} + 1\right)\left(\left(1 - \frac{1}{\beta}\right)^{\beta}\right)^{\frac{b+1}{\beta}} \tag{510}$$

$$= o(1). \tag{511}$$

It follows that

$$\sum_{n=a}^{b} n\left(1 - \frac{1}{\beta}\right)^{n} = \beta^2\left(\left(\frac{a-1}{\beta} + 1\right)\left(1 - \frac{1}{\beta}\right)^{a} + o(1)\right) \tag{512}$$

$$= \beta^2\left(\frac{a-1}{\beta} + 1\right)e^{-a/\beta}(1 + o(1)). \tag{513}$$

B. Additional experiments on deduplication algorithms

In this section, we show additional experiments in more detail. Figures 20, 25, 24, 23, 22 and 21 shows the performances of the five different encoding methods on the synthetic data with different edit probabilities. We can observe that as the edit probability gets larger, the optimal number of check bits decreases. This is in accordance with our theoretic result proved for the AFLD algorithms, where the optimal chunk length is in inverse of $H(\delta)$.

C. Asymptotic Analysis of k-mer Frequencies and Entropy in TDS systems

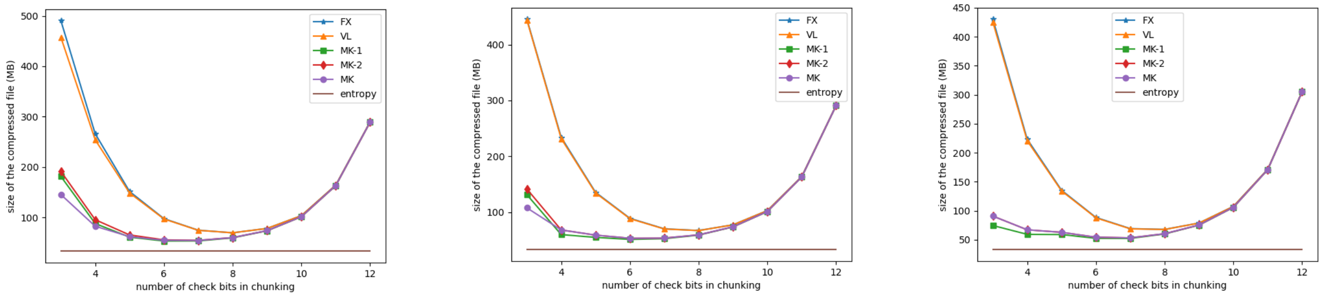
1) Proof of Theorem 45

Theorem 45. *For all $a \in \Sigma$ and $n \geq 1$ we have*

$$\Pr(|x_n^a - x_0^a| \geq \lambda) \leq 2e^{-\lambda^2 L_0/(2M^2)} .$$

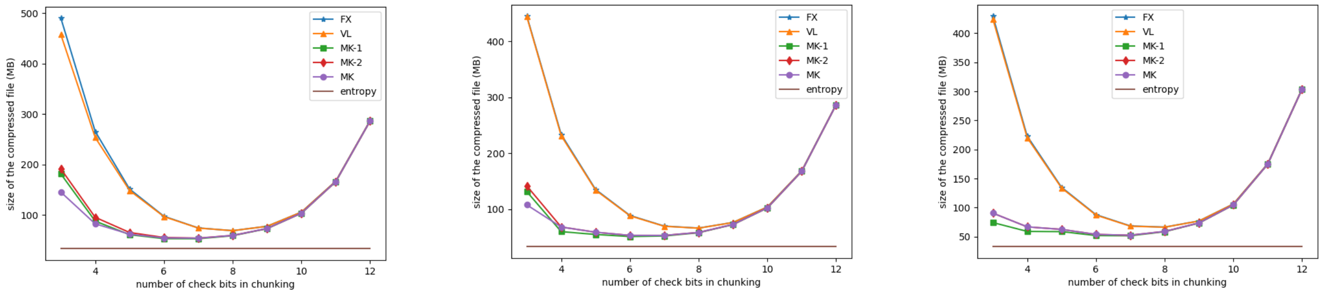
Proof: Since $q_i = 0$ for $i \geq M$ or $i \leq 0$, $\frac{\mu_{n-1}^a}{L_{n-1}+M} \leq \frac{\mu_n^a}{L_n} \leq \frac{\mu_{n-1}^a+M}{L_{n-1}+M}$. Thus

$$-\frac{M\mu_{n-1}^a}{L_{n-1}(L_{n-1}+M)} \leq \frac{\mu_n^a}{L_n} - \frac{\mu_{n-1}^a}{L_{n-1}} \leq \frac{M(L_{n-1} - \mu_{n-1}^a)}{L_{n-1}(L_{n-1}+M)} ,$$



no limit on chunk lengths $2^{a-2} \le$ chunk length$\le 2^{a+2}$ $2^{a-1} \le$ chunk lengths $\le 2^{a+1}$

Figure 20: Compressed file size of the synthetic dataset with $\delta = 0$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.



no limit on chunk lengths $2^{a-2} \le$ chunk length$\le 2^{a+2}$ $2^{a-1} \le$ chunk lengths $\le 2^{a+1}$

Figure 21: Compressed file size of the synthetic dataset with $\delta = 10^{-8}$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.

implying that

$$\begin{aligned}\left|x_n^a - x_{n-1}^a\right| &\le \frac{M \max\{L_{n-1} - \mu_{n-1}^a, \mu_{n-1}^a\}}{L_{n-1}(L_{n-1} + M)} \\ &\le \frac{M}{L_{n-1} + M} \le \frac{M}{L_0 + n - 1 + M} \le \frac{M}{L_0 + n} \ .\end{aligned}$$

Let $c_n = \frac{M}{L_0+n}$ so that $\left|x_n^a - x_{n-1}^a\right| \le c_n$ and note that

$$\begin{aligned}\sum_{i=1}^{n} c_i^2 = M^2 \sum_{i=1}^{n} \frac{1}{(L_0+i)^2} &\le M^2 \int_0^n \frac{dt}{(L_0+t)^2} \\ = \frac{M^2}{L_0} - \frac{M^2}{L_0+n} &= \frac{M^2 n}{L_0(L_0+n)} \le \frac{M^2}{L_0}.\end{aligned}$$

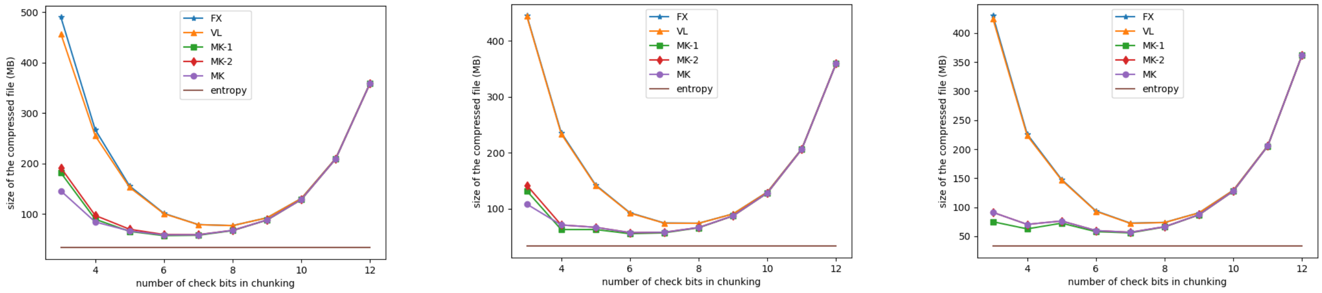
By the Hoeffding-Azuma inequality [41], since $\{x_n^a : n = 0, 1, 2, \ldots\}$ is a martingale and $\left|x_n^a - x_{n-1}^a\right| \le c_n$, we have

$$\begin{aligned}\Pr(|x_n^a - x_0^a| \ge \lambda) &\le 2\exp\left(\frac{-\lambda^2}{2\sum_{i=1}^{n} c_i^2}\right) \\ &\le 2\exp\left(\frac{-\lambda^2 L_0}{2M^2}\right).\end{aligned}$$

■

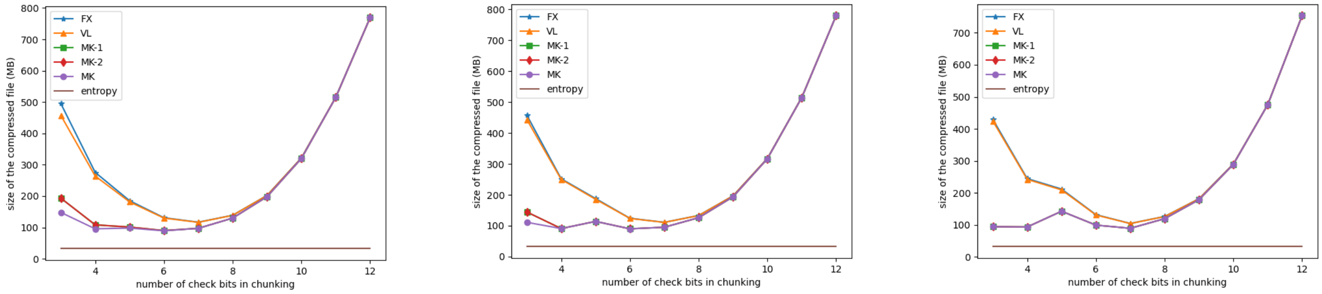
2) Proofs of Lemmas 47, 48 and 49

(a) In case 1, we have $1 \le b < \min(\ell, k-\ell+1)$ (regardless of whether $k \ge 2\ell$ or $k < 2\ell$), the new occurrences of $\boldsymbol{u}$ always contain some (but not all) of the template and all of the new copy. This scenario is labeled as Case 1 in Figure 8. Suppose $Y_b = 1$. Since the copy and the template are identical, elements of $\boldsymbol{u}$ that coincide with the same positions in



no limit on chunk lengths $\quad$ $2^{a-2} \leq$ chunk length$\leq 2^{a+2}$ $\quad$ $2^{a-1} \leq$ chunk lengths $\leq 2^{a+1}$

Figure 22: Compressed file size of the synthetic dataset with $\delta = 10^{-5}$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.



no limit on chunk lengths $\quad$ $2^{a-2} \leq$ chunk length$\leq 2^{a+2}$ $\quad$ $2^{a-1} \leq$ chunk lengths $\leq 2^{a+1}$

Figure 23: Compressed file size of the synthetic dataset with $\delta = 10^{-4}$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.

these two substrings must also be identical. So a necessary condition for $Y_b = 1$ is

$$\boldsymbol{u}_{1,b} = \boldsymbol{u}_{1+\ell,b}.$$

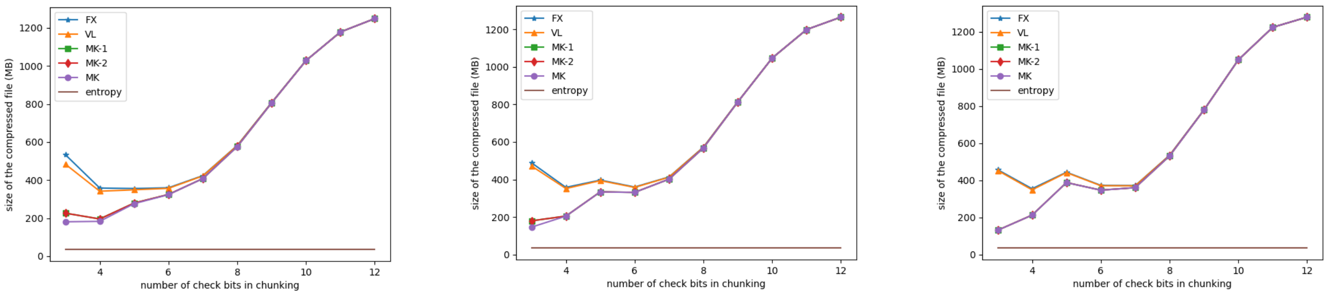
Assume this condition is satisfied. Then $Y_b = 1$ if and only if the sequence starting at the beginning of the template in $\boldsymbol{s}_n$ is equal to $\boldsymbol{u}_{b+1,k-b}$, which has probability $x^{\boldsymbol{u}_{b+1,k-b}}$.
As an example for $k \geq 2\ell$, consider

$$\begin{aligned}
\boldsymbol{s}_n &= \boldsymbol{v}1234567\boldsymbol{w},\\
\boldsymbol{s}_{n+1} &= \boldsymbol{v}12\overline{3\underline{123}456}7\boldsymbol{w}, \text{ where } Y_b = 1 \text{ for } b = 1,\\
u_1 &= u_4 = 3,
\end{aligned}$$

where $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{A}^*$, $\boldsymbol{u}$ is overlined, and the copy is underlined. Note that $\boldsymbol{s}_n$ contains $\boldsymbol{u}_{b+1,k-b} = 123456$. For $k < 2\ell$, consider

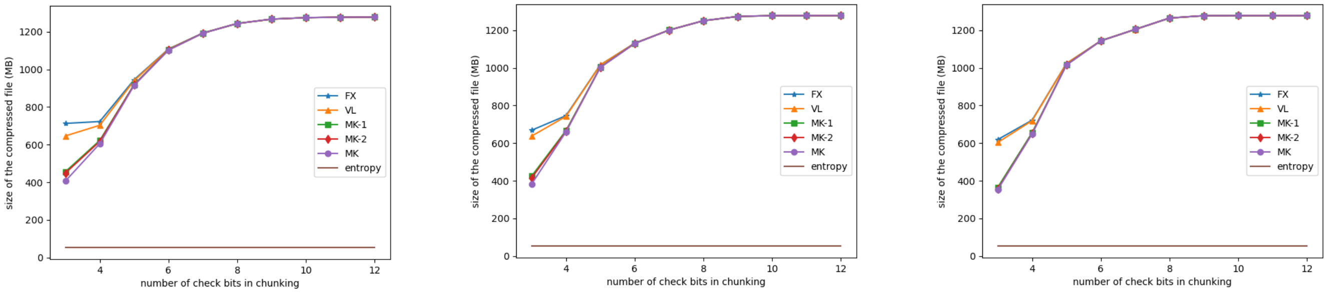
$$\begin{aligned}
\boldsymbol{s}_n &= \boldsymbol{v}1234\boldsymbol{w},\\
\boldsymbol{s}_{n+1} &= \boldsymbol{v}12\overline{3\underline{123}4}\boldsymbol{w}, \text{ where } Y_b = 1 \text{ for } b = 1,\\
u_1 &= u_4 = 3.
\end{aligned}$$

(b) In Case 2, $\boldsymbol{u}$ either i) contains both the template and the copy completely, or ii) intersects with both but contains neither. Note that this case cannot occur if $k = 2\ell - 1$.
First, assume $k \geq 2\ell$. The condition on b translates to $\ell \leq b < k - \ell + 1$ and the new occurrence of $\boldsymbol{u}$ contains both the template and the copy. This is labeled as Case 2 in Figure 8 (below $\boldsymbol{s}_{n+1}$). With the same logic as in Case 1, it is clear that we need



no limit on chunk lengths $2^{a-2} \leq$ chunk length$\leq 2^{a+2}$ $2^{a-1} \leq$ chunk lengths $\leq 2^{a+1}$

Figure 24: Compressed file size of the synthetic dataset with $\delta = 10^{-3}$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.



no limit on chunk lengths $2^{a-2} \leq$ chunk length$\leq 2^{a+2}$ $2^{a-1} \leq$ chunk lengths $\leq 2^{a+1}$

Figure 25: Compressed file size of the synthetic dataset with $\delta = 10^{-2}$ vs. number of check bits for different encoding schemes and different constraints on chunk lengths.

$$\boldsymbol{u}_{b-\ell+1,\ell} = \boldsymbol{u}_{b+1,\ell},$$

Assuming this condition is satisfied, we have $Y_b = 1$ if and only if the substring $\boldsymbol{u}_{1,b-\ell}\boldsymbol{u}_{b+1,k-b}$ occurs in $\boldsymbol{s}_n$ at a certain position, which occurs with probability $x^{\boldsymbol{u}_{1,b-\ell}\boldsymbol{u}_{b+1,k-b}}$.

For example, consider

$$\begin{aligned}\boldsymbol{s}_n &= \boldsymbol{v}412356\boldsymbol{w},\\ \boldsymbol{s}_{n+1} &= \boldsymbol{v}\overline{4123\underline{123}}56\boldsymbol{w}, \text{ where } Y_b = 1 \text{ for } b = 4,\\ \boldsymbol{u}_{2,3} &= \boldsymbol{u}_{5,3} = 123.\end{aligned}$$

Now suppose $\ell + 1 \leq k \leq 2\ell - 2$. The condition on b from the statement of the lemma is $k - \ell + 1 \leq b < \ell$. The new occurrence of $\boldsymbol{u}$ contains some (but not all) of the elements of the template and some (but not all) of the elements of the copy, as illustrated in Figure 8, Case 2, above $\boldsymbol{s}_{n+1}$. The following constraint on $\boldsymbol{u}$ must hold

$$\boldsymbol{u}_{1,k-\ell} = \boldsymbol{u}_{\ell+1,k-\ell},$$

implying that $\phi_\ell(\boldsymbol{u}) = \mathsf{X}^\ell 0^{k-\ell}$. For example, consider

$$\begin{aligned}\boldsymbol{s}_n &= \boldsymbol{v}123\boldsymbol{w},\\ \boldsymbol{s}_{n+1} &= \boldsymbol{v}1\overline{23\underline{12}}\underline{3}\boldsymbol{w}, \text{ where } Y_b = 1 \text{ for } b = 2,\\ u_1 &= u_4 = 2.\end{aligned}$$

We have $Y_b = 1$ iff the sequence starting at the beginning of the template in $\boldsymbol{s}_n$ is equal to $\boldsymbol{u}_{b+1,\ell-b}\boldsymbol{u}_{1,b}$, which has probability $x^{\boldsymbol{u}_{b+1,\ell-b}\boldsymbol{u}_{1,b}}$.

(c) In case 3, we have $\max(k - \ell + 1, \ell) \leq b \leq k - 1$ (regardless of whether $k \geq 2\ell$ or $k < 2\ell$), the new occurrence of $\boldsymbol{u}$ contains the template and some (but not all) of the elements of the copy. This is labeled as Case 3 in Figure 8. The constraint on $\boldsymbol{u}$ is

$$\boldsymbol{u}_{b-\ell+1,k-b} = \boldsymbol{u}_{b+1,k-b}.$$

As examples, consider

$$\begin{aligned}
\boldsymbol{s}_n &= \boldsymbol{v}456123\boldsymbol{w}, \\
\boldsymbol{s}_{n+1} &= \boldsymbol{v}\overline{456123\underline{1}}\underline{23}\boldsymbol{w}, \text{ where } Y_b = 1 \text{ for } b = 6, \\
u_4 &= u_7 = 1,
\end{aligned}$$

for $k \geq 2\ell$, and

$$\begin{aligned}
\boldsymbol{s}_n &= \boldsymbol{v}4123\boldsymbol{w}, \\
\boldsymbol{s}_{n+1} &= \boldsymbol{v}\overline{4123\underline{1}}\underline{23}\boldsymbol{w}, \text{ where } Y_b = 1 \text{ for } b = 4, \\
u_2 &= u_5 = 1,
\end{aligned}$$

for $\ell < k < 2\ell$.

We have $Y_b = 1$ if and only if $\boldsymbol{u}_{1,b}$ occurs in $\boldsymbol{s}_n$ at a certain position, which has probability $x^{\boldsymbol{u}_{1,b}}$.

3) Proof of Theorem 50

Theorem 50. *For an integer $\ell > 0$ and a string $\boldsymbol{u} = u_1 u_2 \cdots u_k$, if $\ell + 1 \leq k < 2\ell$, then*

$$\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) = F_{\ell,l}^{\boldsymbol{u}}(\boldsymbol{x}) + F_{\ell,r}^{\boldsymbol{u}}(\boldsymbol{x}) + M_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - (k-1-\ell)x^{\boldsymbol{u}},$$

and if $k \geq 2\ell$,

$$\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) = F_{\ell,l}^{\boldsymbol{u}}(\boldsymbol{x}) + F_{\ell,r}^{\boldsymbol{u}}(\boldsymbol{x}) + G_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - (k-1-\ell)x^{\boldsymbol{u}}. \tag{367}$$

Proof: From (366), we can write

$$\begin{aligned}
\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) &= \Big(\sum_{b=1}^{k-1} \mathbb{E}_\ell[Y_b | \mathcal{F}_n] \Big) - (k-\ell-1)x^{\boldsymbol{u}} \\
&= \sum_{b=1}^{\min(\ell-1,k-\ell)} \mathbb{E}_\ell[Y_b | \mathcal{F}_n] + \sum_{b=\min(\ell,k-\ell+1)}^{\max(k-\ell,\ell-1)} \mathbb{E}_\ell[Y_b | \mathcal{F}_n] + \sum_{b=\max(k-\ell+1,\ell)}^{k-1} \mathbb{E}_\ell[Y_b | \mathcal{F}_n] - (k-\ell-1)x^{\boldsymbol{u}}. \qquad (514)
\end{aligned}$$

By Lemma 47, we have

$$\begin{aligned}
\sum_{b=1}^{\min(\ell-1,k-\ell)} \mathbb{E}_\ell[Y_b | \mathcal{F}_n] &= \sum_{b=1}^{\min(\ell-1,k-\ell)} x^{\boldsymbol{u}_{b+1,k-b}} I(\boldsymbol{u}_{1,b}, \boldsymbol{u}_{1+\ell,b}) \\
&= \sum_{b=1}^{\min(\ell-1,k-\ell)} x^{\boldsymbol{u}_{b+1,k-b}} I(\boldsymbol{\varphi}_\ell(\boldsymbol{u})_{\ell+1,b}, 0^b) \\
&= \sum_{b=1}^{\min(\ell-1,k-\ell,l_\ell^{\boldsymbol{u}})} x^{\boldsymbol{u}_{b+1,k-b}} \\
&= \sum_{b=1}^{\min(\ell-1,l_\ell^{\boldsymbol{u}})} x^{\boldsymbol{u}_{b+1,k-b}} \\
&= F_{\ell,l}^{\boldsymbol{u}}(\boldsymbol{x}), \qquad (515)
\end{aligned}$$

where the fourth equality follows from the fact that $l_\ell^{\boldsymbol{u}} \leq k - \ell$.

Similarly, using Lemma 49, it can be shown that

$$\begin{aligned}
&\sum_{b=\max(k-\ell+1,\ell)}^{k-1} \mathbb{E}_\ell[Y_b|\mathcal{F}_n] \\
&= \sum_{b=\max(k-\ell+1,\ell)}^{k-1} x^{\boldsymbol{u}_{1,b}} I(\boldsymbol{u}_{b-\ell+1,k-b}, \boldsymbol{u}_{b+1,k-b}) \\
&= \sum_{b=\max(k-\ell+1,\ell)}^{k-1} x^{\boldsymbol{u}_{1,b}} I(\boldsymbol{\varphi}_\ell(\boldsymbol{u})_{b+1,k-b}, 0^{k-b}) \\
&= \sum_{b=\max(k-\ell+1,\ell,k-r_\ell^{\boldsymbol{u}})}^{k-1} x^{\boldsymbol{u}_{1,b}} \\
&= \sum_{b=\max(k-\ell+1,k-r_\ell^{\boldsymbol{u}})}^{k-1} x^{\boldsymbol{u}_{1,b}} \\
&= \sum_{i=1}^{\min(r_\ell^{\boldsymbol{u}},\ell-1)} x^{\boldsymbol{u}_{1,k-i}} \\
&= F_{\ell,r}^{\boldsymbol{u}}(\boldsymbol{x}), \qquad (516)
\end{aligned}$$

where the fourth equality follows from $r_\ell^{\boldsymbol{u}} \leq k-\ell$ and the fifth equality comes from setting $i = k-b$.

To complete the proof, we need to show that $\mathbb{E}_\ell[Y_b|\mathcal{F}_n]$ summed over the range $\min(\ell, k-\ell+1) \leq b \leq \max(k-\ell, \ell-1)$ reduces to $G_\ell^{\boldsymbol{u}}(\boldsymbol{x})$ or $M_\ell^{\boldsymbol{u}}(\boldsymbol{x})$ as appropriate.

From Lemma 48, if $\ell+1 \leq k \leq 2\ell-2$, then

$$\begin{aligned}
&\sum_{b=\min(\ell,k-\ell+1)}^{\max(k-\ell,\ell-1)} \mathbb{E}_\ell[Y_b|\mathcal{F}_n] = \sum_{b=k-\ell+1}^{\ell-1} \mathbb{E}_\ell[Y_b|\mathcal{F}_n] \\
&= \sum_{b=k-\ell+1}^{\ell-1} x^{\boldsymbol{u}_{b+1,\ell-b}\boldsymbol{u}_{1,b}} I(\boldsymbol{u}_{1,k-\ell}, \boldsymbol{u}_{\ell+1,k-\ell}) \\
&= \sum_{b=k-\ell+1}^{\ell-1} x^{\boldsymbol{u}_{b+1,\ell-b}\boldsymbol{u}_{1,b}} I(\boldsymbol{\varphi}_\ell(\boldsymbol{u})_{\ell+1,k-\ell}, 0^{k-\ell}) \\
&= M_\ell^{\boldsymbol{u}}(\boldsymbol{x}), \qquad (517)
\end{aligned}$$

and if $k = 2\ell-1$, also

$$\sum_{b=\min(\ell,k-\ell+1)}^{\max(k-\ell,\ell-1)} \mathbb{E}_\ell[Y_b|\mathcal{F}_n] = 0 = M_\ell^{\boldsymbol{u}}(\boldsymbol{x}). \qquad (518)$$

Finally, if $k \geq 2\ell$, from the same lemma, we find

$$\begin{aligned}
&\sum_{b=\min(\ell,k-\ell+1)}^{\max(k-\ell,\ell-1)} \mathbb{E}_\ell[Y_b|\mathcal{F}_n] = \sum_{b=\ell}^{k-\ell} \mathbb{E}_\ell[Y_b|\mathcal{F}] \\
&= \sum_{b=\ell}^{k-\ell} x^{\boldsymbol{u}_{1,b-\ell}\boldsymbol{u}_{b+1,k-b}} I(\boldsymbol{u}_{b-\ell+1,\ell}, \boldsymbol{u}_{b+1,\ell}) \\
&= \sum_{b=\ell}^{k-\ell} x^{\boldsymbol{u}_{1,b-\ell}\boldsymbol{u}_{b+1,k-b}} I(\boldsymbol{\varphi}_\ell(\boldsymbol{u})_{b+1,\ell}, 0^\ell) = G_k(\boldsymbol{u}), \qquad (519)
\end{aligned}$$

where the last step follows from the definition of G_k.

Summing over the expressions provided by (515)-(519) provides the desired result. ■

4) Proof of Theorem 52

Theorem 52. *Consider a tandem duplication and substitution system with distribution $q = (q_\ell)_{0\le\ell<M}$ over these mutations, with $q_0 < 1$, and let A be the matrix defined for this system by* (369). *The frequencies of substrings $\boldsymbol{u}$ of length $k \ge M$, $(x^{\boldsymbol{u}})_{\boldsymbol{u}\in\Sigma^k}$, converge almost surely to the null space of the matrix A.*

Proof: We first show that the resulting ODE is stable by showing that every eigenvalue of matrix A is either 0 or has a negative real part. This is done by applying the Gershgorin circle theorem [107] to the columns of A (see e.g., (370)). According to the Gershgorin circle theorem, every eigenvalue of A lies within at least one of the closed discs $D_1, \dots, D_{|\Sigma|^k}$ in the complex plain, where the i-th disc centers at the i-th diagonal entry of A with radius equal to the sum of the absolute values of the non-diagonal entries in the i-th column. Since in each column, the diagonal element is the only element that can be negative, it suffices to show that each column of A sums to 0, which then implies that the rightmost point of each circle is the origin. Thus, each eigenvalue of A is either 0 or has a negative real part.

We now show that each column of A_ℓ sums to zero for any ℓ. Fix $\boldsymbol{v} \in U$ and consider the column in A_ℓ that corresponds to $x^{\boldsymbol{v}}$. We denote this column by $A_\ell^{\boldsymbol{v}}$ for simplicity. To identify the element in $A_\ell^{\boldsymbol{v}}$ that corresponds to $\boldsymbol{u}$ (i.e., the element in A in the column corresponding to $\boldsymbol{v}$ and the row corresponding to $\boldsymbol{u}$), we must consider expressions for $h_\ell^{\boldsymbol{u}}(\boldsymbol{x}) = \delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - \ell x^{\boldsymbol{u}}$ and check if $x^{\boldsymbol{v}}$ appears on the right side. The coefficient of $x^{\boldsymbol{v}}$ in $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - \ell x^{\boldsymbol{u}}$ is exactly the entry of $A_\ell^{\boldsymbol{v}}$ in the row corresponding to $x^{\boldsymbol{u}}$. For $\ell > 0$, from (366) and Lemmas 47–49, we can see that the only term with a negative coefficient is $-(k-1)x^{\boldsymbol{u}}$, and the terms with nonnegative coefficients are $\sum_{b=1}^{k-1} \mathbb{E}_\ell[Y_b|\mathcal{F}_n]$. Therefore the case in which $x^{\boldsymbol{v}}$ appears in $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - \ell x^{\boldsymbol{u}}$ with negative coefficient happens only when $\boldsymbol{u} = \boldsymbol{v}$, which implies that $A_\ell^{\boldsymbol{v}}$ has exactly one negative entry, which equals $-(k-1)$. Then we study the case in which $x^{\boldsymbol{v}}$ appears in $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x}) - \ell x^{\boldsymbol{u}}$ with a nonnegative coefficient. By Lemmas 47–49, this happens if and only if $x^{\boldsymbol{v}} = \mathbb{E}_\ell[Y_b|\mathcal{F}_n]$ for some $1 \le b \le k-1$. Note that $\mathbb{E}_\ell[Y_b|\mathcal{F}_n]$ has different forms when b has different values. Inspecting the proofs of Lemmas 47–49 shows that for each value of $b \in [k-1]$, there is precisely one $\boldsymbol{u}$ such that $x^{\boldsymbol{v}} = \mathbb{E}_\ell[Y_b|\mathcal{F}_n]$. Hence, for each $b \in [k-1]$, $x^{\boldsymbol{v}}$ appears in $h_\ell^{\boldsymbol{u}}$ with a nonnegative coefficient, and the coefficient is 1. For example, for $b = 1$, from Lemma 47, this $\boldsymbol{u}$ is equal to $\boldsymbol{v}_\ell \boldsymbol{v}_{1,k-1}$. Since there are $k-1$ possible choices for b, the sum of all nonnegative coefficients is $k-1$, which is also the sum of all nonnegative entries in $A_\ell^{\boldsymbol{v}}$. Therefore the sum of all entries in $A_\ell^{\boldsymbol{v}}$, and thus every column in A_ℓ, is 0, as desired. For $\ell = 0$, we have $h_\ell^{\boldsymbol{u}}(\boldsymbol{x}) = \delta_\ell^{\boldsymbol{u}}(\boldsymbol{x})$, where $\delta_\ell^{\boldsymbol{u}}(\boldsymbol{x})$ is given in Theorem 51. The column corresponding to $x^{\boldsymbol{v}}$ has a negative term equal to $-k$ and $k(|\Sigma|-1)$ positive terms, where each of the positive terms is equal to $\frac{1}{|\Sigma|-1}$, so the sum is again 0.

We have shown that all eigenvalues are either 0 or have negative real parts. For any valid initial point $\boldsymbol{x}_0$, the sum of the elements must be 1. Furthermore, each element must be nonnegative. The fact that the columns of A sum to 0 shows that the sum of the elements of any solution $\boldsymbol{x}_t$ also equals 1 as $d\boldsymbol{x}_t/dt = A\boldsymbol{x}_t$. Furthermore, since only diagonal terms in A can be negative, each element of $\boldsymbol{x}_t$ is also nonnegative. Thus $\boldsymbol{x}_t$ is bounded.

By the Jordan canonical from theorem [69], any square matrix over $\mathbb{C}$ can be decomposed into the form QBQ^{-1} for some invertible matrix Q. Here

$$B = \begin{pmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_m \end{pmatrix}$$

is a block diagonal matrix consisting of Jordan blocks, and the Jordan blocks have the form

$$B_i = \begin{pmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_i & 1 \\ & & & & \lambda_i \end{pmatrix}, \text{ for all } i,$$

where λ_i is one of the eigenvalues of the original matrix. So we can write $A = PJP^{-1}$ for some invertible matrix P, where $J = \begin{pmatrix} J' & 0 \\ 0 & J'' \end{pmatrix}$ and J' and J'' are square matrices corresponding to the eigenvalue $\lambda = 0$ and other eigenvalues respectively.

Let $\boldsymbol{y}_t = P^{-1}\boldsymbol{x}_t$, so that $\dot{\boldsymbol{y}}_t = J\boldsymbol{y}_t$, which we can write in the form $\dot{\boldsymbol{u}}_t = J'\boldsymbol{u}_t$ and $\dot{\boldsymbol{w}}_t = J''\boldsymbol{w}_t$ with $\boldsymbol{y}_t = (\boldsymbol{u}_t, \boldsymbol{w}_t)^T$. Let C be any compact internally chain transitive set of the ODE $\dot{\boldsymbol{y}}_t = J\boldsymbol{y}_t$. We first show that if $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{w}) \in C$, then $\boldsymbol{w} = 0$. Consider the flow starting from $\boldsymbol{y}_0 = (\boldsymbol{u}_0, \boldsymbol{w}_0)^T \in C$ with $\boldsymbol{w}_0 \neq \boldsymbol{0}$. We have $\boldsymbol{w}_t = e^{J''}\boldsymbol{w}_0$. Since J'' has only eigenvalues with negative real parts, $\|\boldsymbol{w}_t\| \leq c_0 e^{-c_1 t}\|\boldsymbol{w}_0\|$ for $t \geq 0$ and some constants $c_0, c_1 > 0$. If $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{w}) \in C$, then $\boldsymbol{w}$ is also in an internally chain transitive set of lower dimension. For $T, \epsilon > 0$, let $\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(n)} = \boldsymbol{w}^{(1)}$ be a chain of points such that the flow of $\dot{\boldsymbol{w}}_t = J''\boldsymbol{w}_t$ starting at $\boldsymbol{w}^{(i)}$ meets the ϵ-neighborhood of $\boldsymbol{w}^{(i+1)}$ after a time $\geq T$. We thus have

$$\|\boldsymbol{w}^{(i+1)}\| \leq c_0 e^{-c_1 T}\|\boldsymbol{w}^{(i)}\| + \epsilon. \tag{520}$$

Since T, ϵ are arbitary, we choose them such that $c_0 e^{-c_1 T} < 1/2$ and $c_0 e^{-c_1 T}\|\boldsymbol{w}^{(1)}\| < \epsilon < \|\boldsymbol{w}^{(1)}\|/2$ if $\|\boldsymbol{w}^{(1)}\| > 0$. Hence, $\|\boldsymbol{w}^{(2)}\| \leq c_0 e^{-c_1 T}\|\boldsymbol{w}^{(i)}\| + \epsilon < 2\epsilon$ and by induction $\|\boldsymbol{w}^{(i+1)}\| \leq c_0 e^{-c_1 T}\|\boldsymbol{w}^{(i)}\| + \epsilon < 2\epsilon$ for $i > 1$. This leads to a contraction since it implies that $\|\boldsymbol{w}^{(n)}\| = \|\boldsymbol{w}^{(1)}\| < 2\epsilon$. Thus $\|\boldsymbol{w}^{(1)}\| = 0$ and for any $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{w})^T \in C$ we must have $\boldsymbol{w} = \boldsymbol{0}$.

Next, note that since $\boldsymbol{x}_t$ is bounded, so is $\boldsymbol{y}_t$. Hence for $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{0})^T \in C$, $e^{J't}\boldsymbol{u}$ must be a constant since it contains no exponential terms ($\lambda = 0$) and cannot contain a polynomial term in t with degree ≥ 1 (because of boundedness). So all flows initiated in C are constant. The same must hold for all flows in D, for any D that is an internally chain transitive invariant set of the ODE $\dot{\boldsymbol{x}}_t = A\boldsymbol{x}_t$. Hence, any point in $\boldsymbol{x} \in D$ must be in the null space of A, that is, $A\boldsymbol{x} = 0$. ■

5) Proof of Theorem 53

Theorem 53. *For the mutation process described above, for $k \in \mathbb{N}^+$, if the vector of the frequencies $\boldsymbol{x}$ of strings of length k converges almost surely to a set Γ_k, then $\mathcal{H}_\infty \leq \mathsf{cap}(\Gamma_k)$.*

Proof: Fix some positive real number $\epsilon > 0$. Denote by X the indicator random variable defined by

$$X = \begin{cases} 0 & ||\boldsymbol{s}_n| - \mathbb{E}[|\boldsymbol{s}_n|]| \geq \epsilon n, \\ 1 & \text{otherwise.} \end{cases}$$

By Hoeffding's inequality,

$$\Pr(X = 0) \leq 2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right).$$

We also note that $|\boldsymbol{s}_n| \leq |\boldsymbol{s}_0| + (M-1)n$ for all n.

Now, let Y be the indicator random variable defined by

$$Y = \begin{cases} 0 & \boldsymbol{x}_n \notin \mathbb{B}_\epsilon(\Gamma_k), \\ 1 & \text{otherwise.} \end{cases}$$

We know that $\boldsymbol{x}_n$ converges almost surely to some point in Γ_k as $n \to \infty$, and thus, there exists $N(\epsilon)$ such that for all $n \geq N(\epsilon)$,

$$\Pr(Y = 0) \leq \epsilon.$$

We combine X and Y by defining the indicator random variable,

$$Z = X \cdot Y.$$

By the union bound,

$$\Pr(Z = 0) \leq \epsilon + 2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right). \tag{521}$$

Using standard bounds on the joint entropy and conditional entropy,

$$H(\boldsymbol{s}_n) \leq H(\boldsymbol{s}_n, Z) = H(\boldsymbol{s}_n|Z) + H(Z).$$

By (521), for large enough n, we have

$$H(Z) \leq H_2\left(\epsilon + 2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right)\right)\log_{|\Sigma|} 2,$$

where $H_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function.

We also have

$$H(\boldsymbol{s}_n|Z)=H(\boldsymbol{s}_n|Z=0)+H(\boldsymbol{s}_n|Z=1).$$

For the first summand, by the definition of conditional entropy, and after replacing the unknown distribution with a uniform one to obtain an upper bound, we get

$$\begin{aligned}H(\boldsymbol{s}_n|Z=0)&\leq\left(\epsilon+2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right)\right)\log_{|\Sigma|}\left|\bigcup_{i=1}^{|\boldsymbol{s}_0|+(M-1)n}\Sigma^i\right|\\&\leq\left(\epsilon+2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right)\right)\log_{|\Sigma|}(|\boldsymbol{s}_0|+(M-1)n+1).\end{aligned}$$

Similarly, for the second summand,

$$\begin{aligned}H(\boldsymbol{s}_n|Z=1)&\leq\left(1-\epsilon-2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right)\right)\\&\qquad\cdot\log_{|\Sigma|}\left(\sum_{i=\mathbb{E}(|\boldsymbol{s}_n|)-\epsilon n}^{\mathbb{E}(|\boldsymbol{s}_n|)+\epsilon n}|\mathcal{B}_i(\mathbb{B}_\epsilon(\Gamma_k))|\right)\\&\leq\log_{|\Sigma|}\left(\sum_{i=\mathbb{E}(|\boldsymbol{s}_n|)-\epsilon n}^{\mathbb{E}(|\boldsymbol{s}_n|)+\epsilon n}|\mathcal{B}_i(\mathbb{B}_\epsilon(\Gamma_k))|\right).\end{aligned}$$

However, by the definition of the capacity of semiconstrained systems, for all large enough n,

$$|\mathcal{B}_i(\mathbb{B}_\epsilon(\Gamma_k))|\leq|\Sigma|^{i\cdot\mathsf{cap}(\mathbb{B}_\epsilon(\Gamma_k))+\epsilon}.$$

It follows that

$$H(\boldsymbol{s}_n|Z=1)\leq\log_{|\Sigma|}\left(2\epsilon n|\Sigma|^{(\mathbb{E}(|\boldsymbol{s}_n|)+\epsilon n)(\mathsf{cap}(\mathbb{B}_\epsilon(\Gamma_k))+\epsilon)}\right).$$

Combining all of these together,

$$\begin{aligned}\mathcal{H}_n&=\frac{1}{\mathbb{E}(|\boldsymbol{s}_n|)}\cdot H(\boldsymbol{s}_n)\\&\leq\frac{1}{\mathbb{E}(|\boldsymbol{s}_n|)}\log_{|\Sigma|}\left(2\epsilon n|\Sigma|^{(\mathbb{E}(|\boldsymbol{s}_n|)+\epsilon n)(\mathsf{cap}(\mathbb{B}_\epsilon(\Gamma_k))+\epsilon)}\right)\\&\quad+\frac{\epsilon+2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right)}{\mathbb{E}(|\boldsymbol{s}_n|)}(|\boldsymbol{s}_0|+(M-1)n+1)\\&\quad+\frac{1}{\mathbb{E}(|\boldsymbol{s}_n|)}H_2\left(\epsilon+2\exp\left(-\frac{2\epsilon^2}{(M-1)^2}n\right)\right)\log_{|\Sigma|}2.\end{aligned}$$

Taking $\limsup_{n\to\infty}$ of both sides we obtain

$$\begin{aligned}\mathcal{H}_\infty&\leq\left(1+\frac{\epsilon}{\sum_{i=1}^{M-1}iq_i}\right)\cdot(\mathsf{cap}(\mathbb{B}_\epsilon(\Gamma_k))+\epsilon)\\&\quad+\frac{\epsilon(M-1)}{\sum_{i=1}^{M-1}iq_i}+H_2(\epsilon)\log_{|\Sigma|}2.\end{aligned}$$

Finally, taking $\lim_{\epsilon\to0^+}$ of both sides, we obtain the claim. ■

D. Finite-time Analysis of k-mer Frequencies and Waiting Time in NTD Systems

1) Proof of Theorem 55 (sketched)

Theorem 55. *Consider the noisy duplication string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$. If the length L_0 of the initial string $\boldsymbol{s}_0$ is greater than ℓ, then for any $\ell < k \leq L_0$, the k-mer frequency vector $\boldsymbol{x}_n$ satisfies*[9]

$$\mathbb{E}[\boldsymbol{x}_{n+1}] - \mathbb{E}[\boldsymbol{x}_n] = \frac{A_k}{L_{n+1}}\mathbb{E}[\boldsymbol{x}_n] \tag{377}$$

for some constant matrix $A_k \in \mathbb{R}^{|\Sigma^k| \times |\Sigma^k|}$ determined by $\boldsymbol{q}$, k, ℓ and independent of any other quantities. Further, all eigenvalues of A_k have non-positive real parts.

sketched: With $\boldsymbol{x}_n$ and $\boldsymbol{\mu}_n$ being shorthand of $\boldsymbol{x}_n(k)$ and $\boldsymbol{\mu}_n(k)$,

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_{n+1} - \boldsymbol{x}_n] &= \mathbb{E}\left[\frac{\boldsymbol{\mu}_{n+1}}{L_{n+1}} - \frac{\boldsymbol{\mu}_n}{L_n}\right] &(522)\\ &= \mathbb{E}\left[\frac{1}{L_{n+1}}\big(\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n - (L_{n+1} - L_n)\boldsymbol{x}_n\big)\right] &(523)\\ &= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{L_{n+1}}\big(\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n - (L_{n+1} - L_n)\boldsymbol{x}_n\big)\mathrm{mid}\mathcal{F}_n\right]\right] &(524)\\ &= \mathbb{E}\left[\frac{1}{L_{n+1}}\big(\mathbb{E}\big[\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n|\mathcal{F}_n\big] - \ell\boldsymbol{x}_n\big)\right]. &(525)\end{aligned}$$

Thus, to prove the desired result (377), it suffices to prove that $\mathbb{E}\big[\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n|\mathcal{F}_n\big]$ equals

$$\mathbb{E}\big[\boldsymbol{\mu}_{n+1} - \boldsymbol{\mu}_n|\mathcal{F}_n\big] = H_k\boldsymbol{x}_n, \tag{526}$$

for some constant matrix H_k determined by $\boldsymbol{q}$, k and ℓ. The characteristic matrix A_k then equals $H + \ell I_{|\Sigma^k|}$, where I_m denotes the identity matrix of size $m \times m$.

For any k-mer $\boldsymbol{u}$, it can be shown that $\mathbb{E}\big[\mu_{n+1}^{\boldsymbol{u}} - \mu_n^{\boldsymbol{u}}|\mathcal{F}_n\big]$ is a linear function of $\boldsymbol{x}_n$. By (376),

$$\mathbb{E}\big[\mu_{n+1}^{\boldsymbol{u}} - \mu_n^{\boldsymbol{u}}|\mathcal{F}_n\big] = \sum_{j=1}^{\ell+k-1}\Pr\big(\boldsymbol{y}_j = \boldsymbol{u}|\mathcal{F}_n\big) - \sum_{l=1}^{k-1}\Pr(\boldsymbol{z}_l = \boldsymbol{u}|\mathcal{F}_n), \tag{527}$$

where substrings $\boldsymbol{y}_j$ are the newly created k-mers in $\boldsymbol{s}_{n+1}$ and substrings $\boldsymbol{z}_l$ are the eliminated k-mers in $\boldsymbol{s}_n$, as demonstrated in Section III-E1.

Summands $\Pr\big(\boldsymbol{y}_j = \boldsymbol{u}|\mathcal{F}_n\big), \Pr(\boldsymbol{z}_l = \boldsymbol{u}|\mathcal{F}_n)$ in (527) can be shown to be linear functions of $\boldsymbol{x}_n$. For instance, consider the term $\Pr(\boldsymbol{y}_1 = \boldsymbol{u}|\mathcal{F}_n)$. Write $\boldsymbol{u} = u_1u_2\cdots u_k$. Since $\boldsymbol{y}_1 = a_{i+\ell-k+2}a_{i+\ell-k+3}\cdots a_{i+\ell}b_{i+1}$, $\boldsymbol{y}_1$ equals $\boldsymbol{u}$ if and only if

$$a_{i+\ell-k+2}a_{i+\ell-k+3}\cdots a_{i+\ell} = u_1u_2\cdots u_{k-1}, \quad b_{i+1} = u_k. \tag{528}$$

Since the position i of the noisy duplication is uniformly distributed, $\Pr\big(\boldsymbol{a}_{i+\ell-k+2}^{i+\ell} = \boldsymbol{u}_1^{k-1}\mathrm{mid}\mathcal{F}_n\big) = x_n^{\boldsymbol{u}_1^{k-1}}$ by the definition of substring frequency. It follows that

$$\begin{aligned}\Pr(\boldsymbol{y}_1 = \boldsymbol{u}|\mathcal{F}_n) &= \Pr\big(\boldsymbol{a}_{i+\ell-k+2}^{i+\ell} = \boldsymbol{u}_1^{k-1}, a'_{i+1} = u_k|\mathcal{F}_n\big) &(529)\\ &= x_n^{\boldsymbol{u}_1^{k-1}}\Pr\big(a'_{i+1} = u_k|\mathcal{F}_n, \boldsymbol{a}_{i+\ell-k+2}^{i+\ell} = \boldsymbol{u}_1^{k-1}\big). &(530)\end{aligned}$$

Given $\boldsymbol{a}_{i+\ell-k+2}^{i+\ell} = \boldsymbol{u}_1^{k-1}$ and mutation $\mathcal{T}_\ell^d$, the probability of $a'_{i+1} = u_k$ equals $\frac{\binom{\ell-1}{d}}{\binom{\ell}{d}}$ (corresponding to that a_{i+1} is not flipped during $\mathcal{T}_\ell^d$) if $a_{i+1} = u_k$, and equals $\frac{1}{|\Sigma|-1}\frac{\binom{\ell-1}{d-1}}{\binom{\ell}{d}}$ (corresponding to that a_{i+1} is flipped to be u_k during $\mathcal{T}_\ell^d$). It follows

[9] Note that it suffices to consider $k > \ell$ since substring frequencies of smaller lengths are linear functions of substring frequencies of larger lengths. The assumption $k \leq L_0$ is to avoid complications of defining k-substrings in strings of lengths less than k.

that

$$\Pr\big(a'_{i+1} = u_k | \mathcal{F}_n, \boldsymbol{a}_{i+\ell-k+2}^{i+\ell} = \boldsymbol{u}_1^{k-1}\big) = \begin{cases} \sum_{d=0}^{\ell} \left(q_d \frac{\binom{\ell-1}{d}}{\binom{\ell}{d}} \right) & \text{if } u_1 = u_k, \\ \frac{1}{|\Sigma|-1} \sum_{d=0}^{\ell} \left(q_d \frac{\binom{\ell-1}{d-1}}{\binom{\ell}{d}} \right) & \text{if } u_1 \neq u_k. \end{cases} \tag{531}$$

Since we can represent $x_n^{\boldsymbol{u}_1^{k-1}}$ as a sum of frequencies of k-mers, $\Pr(\boldsymbol{y}_1 = \boldsymbol{u} | \mathcal{F}_n)$ is a linear function of $\boldsymbol{x}_n$ and the coefficients only depend on $\boldsymbol{q}, \ell$ and k, as shown in (531).

Every summand in (527) can be shown similarly to be a linear function of $\boldsymbol{x}_n$ and the existence of H_k follows immediately.

■

2) Proof of Theorem 56

Theorem 56. *Consider the noisy duplication system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$ with characteristic matrix A_k and k-mer frequency vectors $\boldsymbol{x}_n = \sum_{s=1}^{m} \alpha_n^s \boldsymbol{v}_s$. If A_k is diagonalizable, and all eigenvalues of A_k are real and no smaller than $-\frac{L_0}{2}$,*

1) *For $1 \leq s \leq m$ such that $\lambda_s = 0$ or $\alpha_0^s = 0$,*

$$\mathbb{E}[\alpha_n^s] = \alpha_0^s \quad \text{for all } n \in \mathbb{N}.$$

2) *For $1 \leq s \leq m$ such that $\lambda_s \neq 0$ and $\alpha_0^s \neq 0$,*

$$T_n^s < \frac{\mathbb{E}[\alpha_n^s]}{\alpha_0^s} < U_n^s, \tag{378}$$

where

$$U_n^s = \left(\frac{\lambda_s + L_n}{\lambda_s + L_1} \right)^{\frac{\lambda_s}{\ell}} e^{\lambda_s^2/(L_1 \ell)} \left(1 + \frac{\lambda_s}{L_n} \right),$$
$$T_n^s = \left(\frac{\lambda_s + L_n}{\lambda_s + L_1} \right)^{\frac{\lambda_s}{\ell}} e^{-\lambda_s^2/(L_n \ell)} \left(1 + \frac{\lambda_s}{L_1} \right),$$

and $L_n = L_0 + n\ell$.

Proof: From (377), we have for all $n \geq 1$,

$$\mathbb{E}[\boldsymbol{x}_n] = \mathbb{E}[\boldsymbol{x}_{n-1}] + \frac{A_k}{L_n} \mathbb{E}[\boldsymbol{x}_{n-1}]. \tag{532}$$

Replacing $\mathbb{E}[\boldsymbol{x}_n]$ with $\sum_{s=1}^{m} \alpha_n^s \boldsymbol{v}_s$ gives

$$\sum_{s=1}^{m} \mathbb{E}[\alpha_n^s] \boldsymbol{v}_s = \sum_{s=1}^{m} \mathbb{E}[\alpha_{n-1}^s] \boldsymbol{v}_s + \sum_{s=1}^{m} \mathbb{E}[\alpha_{n-1}^s] \frac{A}{L_n} \boldsymbol{v}_s \tag{533}$$
$$= \sum_{s=1}^{m} \mathbb{E}[\alpha_{n-1}^s] \boldsymbol{v}_s + \sum_{s=1}^{m} \mathbb{E}[\alpha_{n-1}^s] \frac{\lambda_s}{L_n} \boldsymbol{v}_s. \tag{534}$$

Since $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ are a basis, for a given s, we thus find

$$\mathbb{E}[\alpha_n^s] = \left(1 + \frac{\lambda_s}{L_n} \right) \mathbb{E}\big[\alpha_{n-1}^s\big] = \alpha_0^s e^{\sum_{i=1}^{n} f(i)}, \tag{535}$$

where

$$f(i) = \log\left(1 + \frac{\lambda_s}{L_0 + i\ell} \right).$$

It is clear that if $\lambda_s = 0$ or $\alpha_0^s = 0$, then $\mathbb{E}[\alpha_n^s] = \alpha_0^s$. It remains to prove the bounds on $\mathbb{E}[\alpha_n^s]$ when $\alpha_0^s \neq 0$ and $\lambda_s < 0$. We note that $f'(i) = \ell(\frac{1}{L_0 + i\ell + \lambda_s} - \frac{1}{L_0 + i\ell})$, which is positive and continuous for $i \in [1, \infty)$ since $\lambda_s < 0$. It can thus be shown by applying Euler's summation formula [5] that

$$f(1) < \sum_{i=1}^{n} f(i) - \int_{1}^{n} f(x) dx < f(n). \tag{536}$$

For the integral, we have

$$\int_1^n f(x)dx = \frac{1}{\ell}\left[\lambda_s \log\left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right) + L_n \log\left(1+\frac{\lambda_s}{L_n}\right) - L_1 \log\left(1+\frac{\lambda_s}{L_1}\right)\right]. \tag{537}$$

Since $x - x^2 \leq \log(1+x) \leq x$ for $-1/2 \leq x \leq 0$, we obtain

$$\log\left(1+\frac{\lambda_s}{L_n}\right) \leq \frac{\lambda_s}{L_n}, -\log\left(1+\frac{\lambda_s}{L_1}\right) \leq \left(\frac{\lambda_s}{L_1}\right)^2 - \frac{\lambda_s}{L_1},$$

and therefore

$$\int_1^n f(x)dx \leq \frac{\lambda_s}{\ell}\left[\log\left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right) + \frac{\lambda_s}{L_1}\right]. \tag{538}$$

The summation $\sum_{i=1}^n f(i)$ is upper bounded by

$$\sum_{i=1}^n f(i) < \int_1^n f(x)dx + f(n) \tag{539}$$

$$\leq \frac{\lambda_s}{\ell}\left[\log\left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right) + \frac{\lambda_s}{L_1}\right] + \log\left(1 + \frac{\lambda_s}{L_0 + n\ell}\right). \tag{540}$$

The desired upper bound for $\frac{\mathbb{E}[\alpha_n^s]}{\alpha_0^s}$ thus follows from (535).

Similarly, we have

$$\log\left(1+\frac{\lambda_s}{L_n}\right) \geq \frac{\lambda_s}{L_n} - \left(\frac{\lambda_s}{L_n}\right)^2, -\log\left(1+\frac{\lambda_s}{L_1}\right) \geq -\frac{\lambda_s}{L_1}, \tag{541}$$

and

$$\int_1^n f(x)dx \leq \frac{\lambda_s}{\ell}\left[\log\left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right) - \frac{\lambda_s}{L_n}\right]. \tag{542}$$

The desired lower bound thus follows from $f(1) = \log(1 + \lambda_s/L_1)$ and $\sum_{i=1}^n f(i) > f(1) + \int_1^n f(x)dx$. ■

3) Bounds on the eigenbasis representation coefficients when A_k is undiagonalizable

We give bounds on the coefficients of the eigenbasis representation when A_k is not necessarily diagonalizable. In this case, we consider the canonical eigenbasis V composed entirely of Jordan chains [11, Section 9.6]. Specifically, we can find $V = \cup_{i=1}^k V_i$ as a union of non-overlapping Jordan chains, where each $V_i = \{\boldsymbol{v}_i^j : 1 \leq j \leq d_i\}$ contains a chain of generalized eigenvectors corresponding to eigenvalue λ_i with $(A_k - \lambda_i I)^{d_i}\boldsymbol{v}_i^{d_i} = \mathbf{0}$ and $\boldsymbol{v}_i^j = (A_k - \lambda_i I)\boldsymbol{v}_i^{j+1}, j = 1, \ldots, d_i - 1$. Note that the number of Jordan chains corresponding to an eigenvalue λ equals the geometric multiplicity of λ. We use $\alpha_n^{i,j}$ to denote the coefficient of $\boldsymbol{v}_i^j$ in representing the k-mer frequency vector $\boldsymbol{x}_n$ in the eigenbasis V, i.e., $\boldsymbol{x}_n = \sum\limits_{i=1}^k \sum\limits_{j=1}^{d_i} \alpha_n^{i,j}\boldsymbol{v}_i^j$.

Theorem 61. *Consider the noisy duplication system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$ with characteristic matrix A_k and k-mer frequency vectors $\boldsymbol{x}_n = \sum\limits_{i=1}^k \sum\limits_{j=1}^{d_i} \alpha_n^{i,j}\boldsymbol{v}_i^j$. If all eigenvalues of A_k are real and no smaller than $-L_0/2$, then for $n \geq \max_i\{d_i\}$,*

$$\mathbb{E}[\alpha_n^{i,j}] = \left(\prod_{u=1}^n \left(1 + \frac{\lambda_i}{L_u}\right)\right) \cdot \sum_{c=0}^{d_i - j} \alpha_0^{i,j+c} B_i^{[n]}(c), \tag{543}$$

where

$$\left(\prod_{t\in[n]} \frac{1}{L_t + \lambda_i}\right)^{\frac{c}{n}} \binom{n}{c} < B_i^{[n]}(c) < \left(\frac{1}{n}\sum_{t\in[n]} \frac{1}{L_t + \lambda_i}\right)^c \binom{n}{c}. \tag{544}$$

Proof: Similar to the proof of Theorem 56, since $L_n = L_0 + n\ell$ for all $n \geq 0$, we have $\mathbb{E}[\boldsymbol{x}_n] = \left(\prod_{u=1}^n \left(I + \frac{A_k}{L_u}\right)\right)\boldsymbol{x}_0$.

It follows that

$$\mathbb{E}[\boldsymbol{x}_n] = \left(\prod_{u=1}^{n}\left(I+\frac{A_k}{L_u}\right)\right)\sum_{i=1}^{k}\sum_{j=1}^{d_i}\alpha_0^{i,j}\boldsymbol{v}_i^j \tag{545}$$

$$= \sum_{i=1}^{k}\sum_{j=1}^{d_i}\alpha_0^{i,j}\left(\prod_{u=1}^{n}\left(I+\frac{A_k}{L_u}\right)\right)\boldsymbol{v}_i^j. \tag{546}$$

Fix an eigenvalue λ_i and a Jordan chain $\boldsymbol{v}_i^1,\dots,\boldsymbol{v}_i^{d_i}$ of λ_i. We first claim (to be proved later) that for any $1\le j\le d_i$ and any set $H=\{a_1,\dots,a_m\}$ of m positive integers such that $m\ge j$,

$$\left(\prod_{u=1}^{m}\left(I+\frac{A_k}{L_{a_u}}\right)\right)\boldsymbol{v}_i^j = \left(\prod_{u=1}^{m}\left(1+\frac{\lambda_i}{L_{a_u}}\right)\right)\sum_{c=0}^{j-1}\boldsymbol{v}_i^{j-c}B_i^H(c), \tag{547}$$

where

$$B_i^H(c) = \begin{cases}\sum\limits_{\substack{S\subseteq H\\|S|=c}}\prod\limits_{t\in S}\frac{1}{L_t+\lambda_i}, & c\ge 1,\\ 1, \quad c=0.\end{cases}$$

It then follows from (546) that if $n\ge d_i$ for all i,

$$\mathbb{E}[\boldsymbol{x}_n] = \sum_{i=1}^{k}\left(\prod_{u=1}^{n}\left(1+\frac{\lambda_i}{L_u}\right)\right)\left(\sum_{j=1}^{d_i}\sum_{c=0}^{j-1}\boldsymbol{v}_i^{j-c}\alpha_0^{i,j}B_i^{[n]}(c)\right), \tag{548}$$

$$= \sum_{i=1}^{k}\sum_{j=1}^{d_i}\left(\left(\prod_{u=1}^{n}\left(1+\frac{\lambda_i}{L_u}\right)\right)\sum_{c=0}^{d_i-j}\alpha_0^{i,j+c}B_i^{[n]}(c)\right)\boldsymbol{v}_i^j, \tag{549}$$

which leads to

$$\mathbb{E}[\alpha_n^{i,j}] = \left(\prod_{u=1}^{n}\left(1+\frac{\lambda_i}{L_u}\right)\right)\sum_{c=0}^{d_i-j}\alpha_0^{i,j+c}B_i^{[n]}(c). \tag{550}$$

We further use Maclaurin's inequality to prove following bounds on $B_i^{[n]}(c)$: for positive real numbers $\frac{1}{L_t+\lambda_i}$,

$$\binom{n}{c}(S_n)^{\frac{c}{n}} \le B_i^{[n]}(c) = \sum_{\substack{S\subseteq[n]\\|S|=c}}\prod_{t\in S}\frac{1}{L_t+\lambda_i} \le \binom{n}{c}(S_1)^c, \tag{551}$$

where

$$S_1 = \frac{1}{n}\sum_{t\in[n]}\frac{1}{L_t+\lambda_i}, \quad S_n = \prod_{t\in[n]}\frac{1}{L_t+\lambda_i}. \tag{552}$$

It remains to prove the claim (547). For simplicity, we drop the subscript i in the following. Note that by definition of the Jordan chain

$$(A_k-\lambda I)\boldsymbol{v}^1 = \boldsymbol{0}, \tag{553}$$

and

$$A_k\boldsymbol{v}^{j+1} = \lambda\boldsymbol{v}^{j+1}+\boldsymbol{v}^j, \quad 1\le j\le d-1. \tag{554}$$

We run an induction on the pair (m,j).

When $j=1$, $\boldsymbol{v}^1$ is an eigenvector of A_k. By (553), for any $m\ge 1$,

$$\prod_{u=1}^{m}\left(I+\frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^1 = \prod_{u=1}^{m}\left(1+\frac{\lambda}{L_{a_u}}\right)\boldsymbol{v}^1. \tag{555}$$

It is clear that (547) holds.

Next, assume (547) holds for $(m,j)=(p,q)$, $p\ge q$. We show (547) also holds for $(m,j)=(p+1,q+1)$. For any set of positive integers $H = \{a_1, \dots, a_{p+1}\}$,

$$\left(\prod_{u=1}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\right)\boldsymbol{v}^{q+1} \tag{556}$$

$$= \left(\prod_{u=2}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\right)\left(\left(1 + \frac{\lambda}{L_{a_1}}\right)\boldsymbol{v}^{q+1} + \frac{1}{L_{a_1}}\boldsymbol{v}^{q}\right) \tag{557}$$

$$= \left(1 + \frac{\lambda}{L_{a_1}}\right)\prod_{u=2}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q+1} + \frac{1}{L_{a_1}}\prod_{u=2}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q} \tag{558}$$

$$= \left(1 + \frac{\lambda}{L_{a_1}}\right)\prod_{u=3}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\left(\left(1 + \frac{\lambda}{L_{a_2}}\right)\boldsymbol{v}^{q+1} + \frac{1}{L_{a_2}}\boldsymbol{v}^{q}\right) + \frac{1}{L_{a_1}}\prod_{u=2}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q} \tag{559}$$

$$= \left(1 + \frac{\lambda}{L_{a_1}}\right)\left(1 + \frac{\lambda}{L_{a_2}}\right)\prod_{u=3}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q+1} + \left(1 + \frac{\lambda}{L_{a_1}}\right)\frac{1}{L_{a_2}}\prod_{u=3}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q} + \frac{1}{L_{a_1}}\prod_{u=2}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q} \tag{560}$$

$$= \left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\boldsymbol{v}^{q+1} + \sum_{t=1}^{p+1}\left(\prod_{v=1}^{t-1}\left(1 + \frac{\lambda}{L_{a_v}}\right)\right)\frac{1}{L_{a_t}}\left(\prod_{y=t+1}^{p+1}\left(I + \frac{A_k}{L_{a_y}}\right)\right)\boldsymbol{v}^{q}, \tag{561}$$

where the first and the third equality follows from (554), and the last equality follows from continuing expanding the product $\left(I + \frac{A_k}{L_{a_u}}\right)\boldsymbol{v}^{q+1}$.

Since we assumed that (547) holds for $(m, j) = (p, q)$,

$$\sum_{t=1}^{p+1}\left(\prod_{v=1}^{t-1}\left(1 + \frac{\lambda}{L_{a_v}}\right)\right)\frac{1}{L_{a_t}}\left(\prod_{y=t+1}^{p+1}\left(I + \frac{A_k}{L_{a_y}}\right)\right)\boldsymbol{v}^{q} \tag{562}$$

$$=\sum_{t=1}^{p+1}\left(\prod_{v=1}^{t-1}\left(1 + \frac{\lambda}{L_{a_v}}\right)\right)\frac{1}{L_{a_t}}\left(\prod_{y=t+1}^{p+1}\left(1 + \frac{\lambda}{L_{a_y}}\right)\right)\left(\sum_{c=0}^{q-1}\boldsymbol{v}^{q-c}B^{H_t}(c)\right) \tag{563}$$

$$=\left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\left(\sum_{c=0}^{q-1}\sum_{t=1}^{p+1}\frac{1}{L_{a_t}+\lambda}B^{H_t}(c)\boldsymbol{v}^{q-c}\right) \tag{564}$$

$$=\left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\left(\sum_{c=0}^{q-1}B^{H}(c+1)\boldsymbol{v}^{q-c}\right) \tag{565}$$

$$=\left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\left(\sum_{c=1}^{q}B^{H}(c)\boldsymbol{v}^{q+1-c}\right), \tag{566}$$

where $H_t = \{a_{t+1}, \dots, a_{p+1}\}$.

Plugging (566) in (561) gives

$$\left(\prod_{u=1}^{p+1}\left(I + \frac{A_k}{L_{a_u}}\right)\right)\boldsymbol{v}^{q+1} = \left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\boldsymbol{v}^{q+1} + \left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\left(\sum_{c=1}^{q}B^{H}(c)\boldsymbol{v}^{q+1-c}\right) \tag{567}$$

$$= \left(\prod_{u=1}^{p+1}\left(1 + \frac{\lambda}{L_{a_u}}\right)\right)\left(\sum_{c=0}^{q}\boldsymbol{v}^{q+1-c}B^{H}(c)\right). \tag{568}$$

Hence, (547) also holds for $(m, j) = (p + 1, q + 1)$. ■

Note that $B_i^{[n]}(c) < \left(\frac{1}{n}\sum_{t\in[n]}\frac{1}{L_t+\lambda_i}\right)^c\binom{n}{c} = \left(\frac{\log n}{n}\right)^c\frac{n^c}{c!}(1+o(1)) = O((\log n)^c)$. Since $\alpha_0^{i,j}$ are constants determined by the initial state of the system, the behavior of $\mathbb{E}[\alpha_n^{i,j}]$ is again dominated by $\prod_{u\in[n]}(1 + \frac{\lambda_i}{L_u})$, which as can be seen from the proof of Theorem 56 and the discussion that follows, is $\Theta(n^{\frac{\lambda_i}{\ell}})$. So as $\boldsymbol{x}_n$ converges to the limit, the largest nonzero eigenvalue determines the rate of the convergence of the average trajectories.

4) Proof of Lemma 57

Lemma 57. *Consider the noisy string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$ with characteristic matrix A_k for k-mers. For any two k-mers $\boldsymbol{v}, \boldsymbol{w}$ (not necessarily distinct),*

$$\mathbb{E}[x_{n+1}^{\boldsymbol{v}} x_{n+1}^{\boldsymbol{w}}] - \left(\frac{L_n}{L_{n+1}}\right)^2 \mathbb{E}[x_n^{\boldsymbol{v}} x_n^{\boldsymbol{w}}] = \frac{\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T}{(L_{n+1})^2} \mathbb{E}[\boldsymbol{x}_n(2k-2)] \tag{381}$$

$$+ \frac{L_n}{(L_{n+1})^2} (\mathbb{E}[x_n^{\boldsymbol{w}} \cdot H_{i_{\boldsymbol{v}}} \cdot \boldsymbol{x}_n(k)] + \mathbb{E}[x_n^{\boldsymbol{v}} \cdot H_{i_{\boldsymbol{w}}} \cdot \boldsymbol{x}_n(k)]), \tag{382}$$

where H is the matrix $A_k + \ell I_{|\Sigma^k|}$ and H_m denotes the m-th row of H, $\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}$ is a constant vector of length $|\Sigma|^{2k-2}$ determined by $\boldsymbol{v}$, $\boldsymbol{w}$, $\boldsymbol{q}$, k, ℓ and independent of any other quantities. Note that $\boldsymbol{x}_n(0)$ is defined to be the zero vector for all n.

Proof: For any two k-mers $\boldsymbol{v}$ and $\boldsymbol{w}$,

$$\mathbb{E}\big[\mu_{n+1}^{\boldsymbol{v}} \mu_{n+1}^{\boldsymbol{w}} | \mathcal{F}_n\big] = \mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}} + \mu_n^{\boldsymbol{v}}\big)\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}} + \mu_n^{\boldsymbol{w}}\big) | \mathcal{F}_n\big] \tag{569}$$

$$= \mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big) | \mathcal{F}_n\big] + \mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\mu_n^{\boldsymbol{w}} | \mathcal{F}_n\big] \tag{570}$$

$$+ \mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big)\mu_n^{\boldsymbol{v}} | \mathcal{F}_n\big] + \mathbb{E}[\mu_n^{\boldsymbol{v}} \mu_n^{\boldsymbol{w}} | \mathcal{F}_n]. \tag{571}$$

We first show that $\mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big) | \mathcal{F}_n\big]$ is a linear function of $\boldsymbol{x}_n(2k-2)$. Consider again the evolution from $\boldsymbol{s}_n$ to $\boldsymbol{s}_{n+1}$ given by (374) and (375), in which noisy duplication inserts ℓ-substring $a'_{i+1} \cdots a'_{i+\ell}$. For each type of mutation $\mathcal{T}_\ell^d$, there are $\binom{\ell}{d}$ ways of choosing d symbols out of ℓ for substitution. Moreover, each symbol can be substituted by the other $|\Sigma| - 1$ alphabet symbols. So it adds up to totally $\binom{\ell}{d}(|\Sigma| - 1)^d$ mutation events, and they are all equally likely. We index the mutation events by $\mathcal{T}_\ell^d(t), t = 1, \ldots, m_{\ell,d}$, $m_{\ell,d} = \binom{\ell}{d}(|\Sigma| - 1)^d$. During noisy duplication, the created and the eliminated k-mers $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\ell+k-1}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k-1}$ are determined by the $(2k-2)$-substring $a_{i+\ell-k+2} \cdots a_{i+\ell+k-1}$ of $\boldsymbol{s}_n$ and the inserted noisy copy $a'_{i+1} \cdots a'_{i+\ell}$. Moreover, the inserted symbols $a'_{i+1} \cdots a'_{i+\ell}$ are uniquely determined by d, t and $a_{i+1}, \ldots, a_{i+\ell}$. Thus, given $\mathcal{T}_\ell^d(t)$ and $\boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1} = \boldsymbol{g}$ for some $\boldsymbol{g} \in \Sigma^{2k-2}$, $\mu_{n+1}^{\boldsymbol{u}} - \mu_n^{\boldsymbol{u}} = \sum_{b=1}^{\ell+k-1} I(\boldsymbol{y}_b, \boldsymbol{u}) - \sum_{c=1}^{k-1} I(\boldsymbol{z}_c, \boldsymbol{u})$ is a constant, denoted $\delta_{d,t,\boldsymbol{g}}^{\boldsymbol{u}}$, for any k-mer $\boldsymbol{u}$. Let $\boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1} = \boldsymbol{h}_i$. It follows that

$$\mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big) | \mathcal{F}_n\big] \tag{572}$$

$$= \sum_{d=0}^{\ell} \sum_{t=1}^{m_{\ell,d}} \sum_{\boldsymbol{g} \in \Sigma^{2k-2}} \mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big) | \mathcal{T}_\ell^d(t), \boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1}, \mathcal{F}_n\big] \Pr\big(\mathcal{T}_\ell^d(t), \boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1} | \mathcal{F}_n\big) \tag{573}$$

$$= \sum_{d=0}^{\ell} \sum_{t=1}^{m_{\ell,d}} \sum_{\boldsymbol{g} \in \Sigma^{2k-2}} \delta_{d,t,\boldsymbol{g}}^{\boldsymbol{v}} \delta_{d,t,\boldsymbol{g}}^{\boldsymbol{w}} \cdot \Pr\big(\boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1} = \boldsymbol{g} | \mathcal{F}_n\big) \cdot \Pr\big(\mathcal{T}_\ell^d(t) | \mathcal{F}_n\big) \tag{574}$$

$$= \sum_{\boldsymbol{g} \in \Sigma^{2k-2}} \left(\sum_{d=0}^{\ell} \sum_{t=1}^{m_{\ell,d}} \delta_{d,t,\boldsymbol{g}}^{\boldsymbol{v}} \delta_{d,t,\boldsymbol{g}}^{\boldsymbol{w}} \frac{q_\ell^d}{\binom{\ell}{d}(|\Sigma| - 1)^d} \right) \cdot x_n^{\boldsymbol{g}}, \tag{575}$$

where the second equality follows from that given $\mathcal{F}_n$, $\boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1} = \boldsymbol{g}$ is independent of $\mathcal{T}_\ell^d(t)$, and the last equality follows from that

$$\Pr\big(\boldsymbol{a}_{i+\ell-k+2}^{i+\ell+k-1} = \boldsymbol{g} | \mathcal{F}_n\big) = x_n^{\boldsymbol{g}}, \tag{576}$$

and $\mathcal{T}_\ell^d(1), \ldots, \mathcal{T}_\ell^d(m_{\ell,d})$ are equally likely with total probability q_ℓ^d. The equality

$$\mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big) | \mathcal{F}_n\big] = \boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T \cdot \boldsymbol{x}_n(2k-2) \tag{577}$$

thus follows immediately from (575) with the $i_{\boldsymbol{g}}$-th element of $\boldsymbol{d}_{\boldsymbol{v},\boldsymbol{w}}^T$ equal to $\sum_{d=0}^{\ell} \sum_{t=1}^{m_{\ell,d}} \delta_{d,t,\boldsymbol{g}}^{\boldsymbol{v}} \delta_{d,t,\boldsymbol{g}}^{\boldsymbol{w}} \frac{q_\ell^d}{\binom{\ell}{d}(|\Sigma|-1)^d}$.

Moreover, equation (526) gives

$$\mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{v}} - \mu_n^{\boldsymbol{v}}\big)\mu_n^{\boldsymbol{w}} | \mathcal{F}_n\big] = \mu_n^{\boldsymbol{w}} \cdot H_{i_{\boldsymbol{v}}} \boldsymbol{x}_n(k), \quad \mathbb{E}\big[\big(\mu_{n+1}^{\boldsymbol{w}} - \mu_n^{\boldsymbol{w}}\big)\mu_n^{\boldsymbol{v}} | \mathcal{F}_n\big] = \mu_n^{\boldsymbol{v}} \cdot H_{i_{\boldsymbol{w}}} \boldsymbol{x}_n(k). \tag{578}$$

The desired result thus follows by noting that $\mu_n^{\boldsymbol{u}} = L_n x_n^{\boldsymbol{u}}$. ■

5) Proof of Theorem 59

Theorem 59. *Consider the noisy string system $\mathcal{S}(\boldsymbol{s}_0, \ell, \boldsymbol{q})$. For $\boldsymbol{u} \in \Sigma^k$, if $\mathbb{E}[x_n^{\boldsymbol{u}}]$ is non-decreasing in n,*

$$\Pr(\tau_{\boldsymbol{u}}(m) \leq n) \leq \left(\frac{1}{m} + \frac{\max(|\boldsymbol{u}| - \ell - 1, 0)}{\ell} + \big(1 - q_\ell^0\big)\right)\mathbb{E}[\mu_n^{\boldsymbol{u}}].$$

Proof: Fix n, m. We first write

$$\Pr(\tau_{\boldsymbol{u}}(m) \leq n) = \Pr(\tau_{\boldsymbol{u}}(m) \leq n, \mu_n^{\boldsymbol{u}} \geq m) + \Pr(\tau_{\boldsymbol{u}}(m) \leq n, \mu_n^{\boldsymbol{u}} < m). \tag{579}$$

For the first term on the right-hand side of (579), since $\mu_n^{\boldsymbol{u}} \geq m$ directly implies that $\boldsymbol{u}$ appears m times before or at step n, i.e., $\tau_{\boldsymbol{u}}(m) \leq n$, we have

$$\Pr(\tau_{\boldsymbol{u}}(m) \leq n, \mu_n^{\boldsymbol{u}} \geq m) = \Pr(\mu_n^{\boldsymbol{u}} \geq m) \leq \frac{\mathbb{E}[\mu_n^{\boldsymbol{u}}]}{m}, \tag{580}$$

where the inequality follows from the Markov bound.

For the second term on the right-hand side of (579), we note that events $\tau_{\boldsymbol{u}}(m) \leq n$ and $\mu_n^{\boldsymbol{u}} < m$ imply that there must exist $n^\circ \leq n - 1$ such that $\mu_{n^\circ}^{\boldsymbol{u}} \geq m$ and $\mu_{n^\circ+1}^{\boldsymbol{u}} < m$. By the union bound that

$$\begin{aligned}\Pr(\tau_{\boldsymbol{u}}(m) \leq n, \mu_n^{\boldsymbol{u}} < m) &\leq \sum_{n^\circ=0}^{n-1} \Pr\big(\mu_{n^\circ}^{\boldsymbol{u}} \geq m, \mu_{n^\circ+1}^{\boldsymbol{u}} < m\big) & (581)\\ &= \sum_{n^\circ=0}^{n-1} \Pr\big(\mu_{n^\circ+1}^{\boldsymbol{u}} < m | \mu_{n^\circ}^{\boldsymbol{u}} \geq m\big) \Pr(\mu_{n^\circ}^{\boldsymbol{u}} \geq m). & (582)\end{aligned}$$

Consider the event that $\mu_{n^\circ}^{\boldsymbol{u}} \geq m$ but $\mu_{n^\circ+1}^{\boldsymbol{u}}$ becomes less than m. For a noisy tandem duplication to eliminate an occurrence of $\boldsymbol{u}$, the duplicated substring must be either fully contained by some occurrence of $\boldsymbol{u}$ (see Figure 26), or it overlaps with the beginning of some occurrence of $\boldsymbol{u}$ and the duplicate is not exact (see Figure 27). Focusing on any m occurrences of $\boldsymbol{u}$ at step n°, there are at most $m \cdot \max(|\boldsymbol{u}| - \ell - 1, 0)$ positions for the noisy duplication where the former case can happen and there are at most $m \cdot \ell$ positions where the latter case can happen. Since the position of noisy duplication is uniformly distributed,

$$\Pr\big(\mu_{n^\circ+1}^{\boldsymbol{u}} < m | \mu_{n^\circ}^{\boldsymbol{u}} \geq m\big) \leq \frac{m \cdot \max(|\boldsymbol{u}| - \ell - 1, 0)}{L_{n^\circ}} + \frac{m \cdot \ell}{L_{n^\circ}}\big(1 - q_\ell^0\big), \tag{583}$$

where q_ℓ^0 is the probability that the duplication is exact.

Thus, (582) is upper bounded by

$$\sum_{n^\circ=0}^{n-1}\left(\frac{m \cdot \max(|\boldsymbol{u}| - \ell - 1, 0)}{L_{n^\circ}} + \frac{m \cdot \ell}{L_{n^\circ}}\big(1 - q_\ell^0\big)\right)\frac{\mathbb{E}[\mu_{n^\circ}^{\boldsymbol{u}}]}{m} \tag{584}$$

$$\begin{aligned}&= \big(\max(|\boldsymbol{u}| - \ell - 1, 0) + \ell\big(1 - q_\ell^0\big)\big) \sum_{n^\circ=0}^{n-1} \mathbb{E}[x_{n^\circ}^{\boldsymbol{u}}] & (585)\\ &\leq \big(\max(|\boldsymbol{u}| - \ell - 1, 0) + \ell\big(1 - q_\ell^0\big)\big) \cdot n\mathbb{E}[x_n^{\boldsymbol{u}}] & (586)\\ &\leq \left(\frac{\max(|\boldsymbol{u}| - \ell - 1, 0)}{\ell} + \big(1 - q_\ell^0\big)\right)\mathbb{E}[\mu_n^{\boldsymbol{u}}], & (587)\end{aligned}$$

where the second inequality follows from $\mathbb{E}[x_n^{\boldsymbol{u}}]$ is increasing in n, the last inequality follows from $L_n \geq n\ell$.

The desired result thus follows from combining equations (580) and (587).

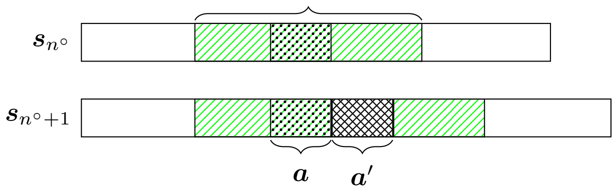


Figure 26: A noisy duplication with $\boldsymbol{a}'$ being an approximate copy of $\boldsymbol{a}$ and $\boldsymbol{a}$ is fully contained in $\boldsymbol{u}$.

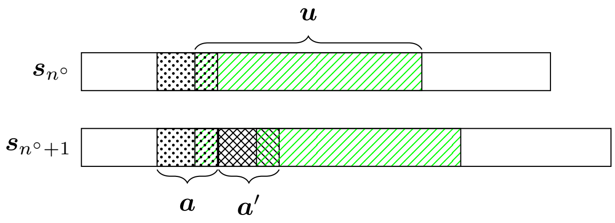


Figure 27: A noisy duplication with $\boldsymbol{a}'$ being an approximate copy of $\boldsymbol{a}$ and $\boldsymbol{a}$ overlaps with the beginning of $\boldsymbol{u}$.

■

6) Proofs of equations (393) **and** (394)

The characteristic matrix for 2-mers of the system in Example 16 can be found as

$$A_2 = \begin{bmatrix} -2\delta & 1-\delta & 1-\delta & \delta/2 & 0 & 0 & \delta/2 & 0 & 0 \\ \delta/2 & -\delta/2-1 & \delta/2 & 0 & \delta/2 & 0 & 0 & \delta/2 & 0 \\ \delta/2 & \delta/2 & -\delta/2-1 & 0 & 0 & \delta/2 & 0 & 0 & \delta/2 \\ \delta/2 & 0 & 0 & -\delta/2-1 & \delta/2 & \delta/2 & \delta/2 & 0 & 0 \\ 0 & \delta/2 & 0 & 1-\delta & -2\delta & 1-\delta & 0 & \delta/2 & 0 \\ 0 & 0 & \delta/2 & \delta/2 & \delta/2 & -\delta/2-1 & 0 & 0 & \delta/2 \\ \delta/2 & 0 & 0 & \delta/2 & 0 & 0 & -\delta/2-1 & \delta/2 & \delta/2 \\ 0 & \delta/2 & 0 & 0 & \delta/2 & 0 & \delta/2 & -\delta/2-1 & d/2 \\ 0 & 0 & \delta/2 & 0 & 0 & \delta/2 & 1-\delta & 1-\delta & -2\delta \end{bmatrix}. \tag{588}$$

The matrix A_2 is diagonalizable with an eigenbasis $[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_9] =$

$$\begin{bmatrix} -1 & -1 & 1 & 1 & 1 & 1 & 1 & (\delta-2)/\delta & 0 \\ 1 & 0 & -(2\delta)/(\delta-2) & -1 & 0 & -1 & 0 & 0 & \delta/(\delta-2) \\ 0 & 1 & -(2\delta)/(\delta-2) & 0 & -1 & 0 & -1 & -1 & -\delta/(\delta-2) \\ -1 & -1 & -(2\delta)/(\delta-2) & -1 & -1 & 0 & 0 & 0 & \delta/(\delta-2) \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & -(\delta-2)/\delta & -1 \\ 0 & 1 & -(2\delta)/(\delta-2) & 0 & 1 & 0 & 0 & 1 & 0 \\ -1 & -1 & -(2\delta)/(\delta-2) & 0 & 0 & -1 & -1 & -1 & -\delta/(\delta-2) \\ 1 & 0 & -(2\delta)/(\delta-2) & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \tag{589}$$

and corresponding eigenvalues $\lambda_1 = \lambda_2 = -1, \lambda_3 = 0, \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = -\frac{3}{2}\delta - 1, \lambda_8 = \lambda_9 = -\frac{3}{2}\delta$.

Consider the vector of 2-mer frequencies $\boldsymbol{x}_n = \left(x_n^{00}, x_n^{01}, x_n^{02}, x_n^{10}, x_n^{11}, x_n^{12}, x_n^{20}, x_n^{21}, x_n^{22}\right)$ and its eigenbasis representation

$$\boldsymbol{x}_n = \sum_{i=1}^{9} \alpha_n^i \boldsymbol{v}_i. \tag{590}$$

The frequencies of 11 and 12 thus equal

$$x_n^{11} = \alpha_n^1 \cdot 1 + \alpha_n^3 \cdot 1 + \alpha_n^4 \cdot 1 + \alpha_n^8 \cdot \left(-\frac{\delta-2}{\delta}\right) + \alpha_n^9 \cdot (-1), \tag{591}$$

$$x_n^{12} = \alpha_n^2 \cdot 1 + \alpha_n^3 \cdot \left(-\frac{2\delta}{\delta-2}\right) + \alpha_n^5 \cdot 1 + \alpha_n^8 \cdot 1. \tag{592}$$

With $\boldsymbol{s}_0$ being an all-zero sequence, we have $\boldsymbol{x}_0 = (1,0,0,0,0,0,0,0,0)$ and

$$\boldsymbol{\alpha}_0 = \left(\alpha_0^1, \alpha_0^2, \alpha_0^3, \alpha_0^4, \alpha_0^5, \alpha_0^6, \alpha_0^7, \alpha_0^8, \alpha_0^9\right)^T \tag{593}$$

$$= \left(0, 0, \frac{2-\delta}{3(3\delta+2)}, \frac{-\delta^2+2\delta}{2(3\delta+2)}, \frac{\delta^2}{3\delta+2}, \frac{\delta^2}{3\delta+2}, \frac{-\delta^2+2\delta}{2(3\delta+2)}, \frac{-\delta}{3}, \frac{\delta}{6} - \frac{1}{3}\right)^T. \tag{594}$$

We then use Theorem 56 to bound the expected value of $\boldsymbol{\alpha}_n$, which can further lead to the k-mer frequency $\boldsymbol{x}_n$.

For $s = 1, 2, 3$, Theorem 56 states that $\mathbb{E}[\alpha_n^s] = \alpha_0^s$. So $\mathbb{E}\big[\alpha_n^1\big] = \mathbb{E}\big[\alpha_n^2\big] = 0, \mathbb{E}\big[\alpha_n^3\big] = \frac{2-\delta}{3(3\delta+2)}$. Further, for $s = 4, 5$, since $\lambda_4 = \lambda_5 = -\frac{3}{2}\delta - 1$, Theorem 56 gives that if $L_1 \geq 10$ and $n \geq 2$,

$$
\begin{align}
\frac{\mathbb{E}[\alpha_n^s]}{\alpha_0^s} &< \left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right)^{\frac{\lambda_s}{\ell}} e^{\lambda_s^2/(L_1\ell)}\left(1 + \frac{\lambda_s}{L_n}\right) \tag{595}\\
&= \left(\frac{\big(-\frac{3}{2}\delta - 1\big) + L_n}{\big(-\frac{3}{2}\delta - 1\big) + L_1}\right)^{\left(-\frac{3}{2}\delta - 1\right)} e^{\left(-\frac{3}{2}\delta-1\right)^2/L_1}\left(1 + \frac{\big(-\frac{3}{2}\delta - 1\big)}{L_n}\right) \tag{596}\\
&< \left(1 + \frac{L_n - L_1}{-\frac{3}{2}\delta - 1 + L_1}\right)^{-1} e^{5/8} \tag{597}\\
&< \frac{2L_1}{n-1}. \tag{598}
\end{align}
$$

Similarly, for $s = 8, 9$, if $L_1 \geq 10$ and $n \geq 2$,

$$
\begin{align}
\frac{\mathbb{E}[\alpha_n^s]}{\alpha_0^s} &> \left(\frac{\lambda_s + L_n}{\lambda_s + L_1}\right)^{\frac{\lambda_s}{\ell}} e^{-\lambda_s^2/(L_n\ell)}\left(1 + \frac{\lambda_s}{L_1}\right) \tag{599}\\
&= \left(\frac{-\frac{3}{2}\delta + L_n}{-\frac{3}{2}\delta + L_1}\right)^{-\frac{3}{2}\delta} e^{-\left(-\frac{3}{2}\delta\right)^2/L_n}\left(1 + \frac{-\frac{3}{2}\delta}{L_1}\right) \tag{600}\\
&> \frac{1}{2}n^{-\frac{3}{2}\delta}. \tag{601}
\end{align}
$$

It thus follows from (591) and (592) that

$$
\begin{align}
\mathbb{E}\big[x_n^{11}\big] &= \mathbb{E}\big[\alpha_n^1\big] + \mathbb{E}\big[\alpha_n^3\big] + \mathbb{E}\big[\alpha_n^4\big] + \mathbb{E}\big[\alpha_n^8\big] \cdot \left(\frac{2-\delta}{\delta}\right) + \mathbb{E}\big[\alpha_n^9\big] \cdot (-1) \tag{602}\\
&= \frac{2-\delta}{3(3\delta+2)} + \frac{\mathbb{E}\big[\alpha_n^4\big]}{\alpha_0^4} \cdot \alpha_0^4 + \frac{\mathbb{E}\big[\alpha_n^8\big]}{\alpha_0^8} \cdot \left(\frac{(2-\delta)\alpha_0^8}{\delta}\right) + \frac{\mathbb{E}\big[\alpha_n^9\big]}{\alpha_0^9} \cdot \big(-\alpha_0^9\big) \tag{603}\\
&< \frac{2-\delta}{3(3\delta+2)} + \frac{L_1\delta(2-\delta)}{(n-1)(3\delta+2)} - \frac{1}{12}n^{-\frac{3}{2}\delta}(2-\delta) \tag{604}\\
&< \frac{1}{3} + \frac{L_1\delta}{n-1} - \frac{1}{12}n^{-\frac{3}{2}\delta}, \tag{605}\\
\mathbb{E}\big[x_n^{12}\big] &= \mathbb{E}\big[\alpha_n^2\big] + \mathbb{E}\big[\alpha_n^3\big] \cdot \left(-\frac{2\delta}{\delta-2}\right) + \mathbb{E}\big[\alpha_n^5\big] + \mathbb{E}\big[\alpha_n^8\big] \tag{606}\\
&= \left(\frac{2-\delta}{3(3\delta+2)}\right) \cdot \left(-\frac{2\delta}{\delta-2}\right) + \frac{\mathbb{E}\big[\alpha_n^5\big]}{\alpha_0^5} \cdot \alpha_0^5 + \frac{\mathbb{E}\big[\alpha_n^8\big]}{\alpha_0^8} \cdot \alpha_0^8 \tag{607}\\
&< \frac{2\delta}{3(3\delta+2)} + \frac{2L_1}{n-1} \cdot \frac{\delta^2}{3\delta+2} + \frac{1}{2}n^{-\frac{3}{2}\delta}\left(-\frac{\delta}{3}\right) \tag{608}\\
&< \frac{\delta}{3} + \frac{L_1}{n-1}\delta^2 - \frac{1}{12}\delta n^{-\frac{3}{2}\delta}. \tag{609}
\end{align}
$$