Tenschnische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Channel Codes for Reliable and Efficient Data Storage in Modern Memories

Andreas Philipp Lenz

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Reinhard Heckel
Prüfende der Dissertation:	Prof. Dr.-Ing. Antonia Wachter-Zeh
	Prof. Olgica Milenkovic, Ph.D.

Die Dissertation wurde am 17.11.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 30.03.2022 angenommen.

Acknowledgement

"*Nothing we achieve in this world is achieved alone. It is always achieved with others teaching us along the way.*" – Lee J. Colan

I could not have undertaken this journey without my advisor Antonia Wachter-Zeh, who offered me a position as a doctoral candidate, despite my little experience in the field coding theory. Through her, I found my way quickly into my dissertation project and later had the freedom to conduct research in many fruitful collaborations abroad. Major thanks to the trust and support she provided during the whole time of my candidacy.

At the beginning of my doctoral candidacy, I was lucky to gain Eitan Yaakobi as a mentor. I'm extremely grateful for all the things he taught me in the last years, ranging from scientific methodology to life lessons, such as not to double-check your research results on a Friday evening. My life as a researcher would not have been the same without the surfing, the research, and other fun activities that we did together.

I am further extremely grateful to Paul Siegel, with whom I had the honor to collaborate in several research projects. In addition to unforgettable memories from my visits in San Diego, through him, I learned the art of writing scientific paragraphs that are both precise and informative. Many thanks to his patience and eye for the details, which I could hopefully adopt to some extend.

It was an honor for me to have my dissertation examined from an internationally renowned examination board. I would like thank Olgica Milenkovic (University of Illinois, Urbana-Champaign) for reviewing and grading my thesis and Reinhard Heckel (Technical University of Munich) for leading the examination board.

During my collaborative research projects, I had the joy of working together with many inspiring scientists and I am grateful for all the fun and interesting things I learned through joint work. In particular, I would like to thank my co-authors Anisha Banerjee, Rawad Bitar, Alexandre Graell i Amat, Matthias Hiller, Vincent Immler, Niklas Jünger, Qinzhi Liu, Yi Liu, Issam Maarouf, Stephen Melczer, Nikita Polyanskii, Sven Puchinger, Cyrus Rashtchian, Eirik Rosnes, Tal Shinkar, and Lorenz Welter.

Special thanks to my long-term roommates Lukas Holzbaur and Julian Renner. It has been a wonderful journey and I deeply appreciate the fun and collaborative atmosphere throughout the years. The many joint conference and workshop visits have been a highlight of my research life. Sharing new findings or discussing pressing problems and design questions, whether boxes should have round or sharp edges has always been a pleasure! Further, I would like to acknowledge Maximilian Egger, Sebastian Bitzer, and Violetta Weger, my recent room mates, who arguably maintained their high quality standards.

The colleagues of the Coding and Cryptography (COD), Leitungsgebundene Übertragungstechnik LÜT, and Lehrstuhl für Nachrichtentechnik (LNT) groups form a formidable research environment and I am glad for having been part of this institute and its many celebrations and excursions. Many thanks for the collegiality and friendship.

A warm thanks to my whole family for their love and supporting my decisions, which has given me the confidence to approach new challenges.

Finally, I would like to express my deepest gratitude to my wife, Irina, who continuously empowered me with her positive attitude, affection, and emotional support and last, but not least, did not intervene when I used the door of the fridge to derive equations.

Andreas Lenz

München, May 2022

Abstract

Digital memories constitute an inevitable component of a vast share of novel technologies. The demand for ever increasing capacities and reliability fuels a continuous effort in increasing the efficiency of current storage systems as well as the development of innovative storage strategies.

This dissertation deals with information and coding theory for modern data storage systems. We propose and analyze models that abstract different aspects of digital memories with a particular focus on reliability and efficiency of the underlying systems. Many of our models find, amongst others, application in DNA-based data storage systems, which constitute a novel candidate for long-term archival data storage.

In the first part of this dissertation, we propose and investigate a channel model whose input and output are several unordered sequences. During transmission, these may be perturbed by errors or can even be completely lost. We design zero-error channel codes, viz. codes that guarantee error-free transmission under the assumption of limited severity of the channel. Our analysis is supported by existential and converse bounds which exhibit the theoretical limits for reliable transmission over this channel. We further explore the special case of indexing the sequences and propose a novel efficient code construction that achieves high information rates close to the maximum possible, in particular for a moderate number of errors.

In the second part of this dissertation, we derive the information capacity of the unordered parallel multinomial channel. This probabilistic channel comprises two stages. First, many parallel input sequences are permuted in an arbitrary manner and, second, these sequences traverse noisy channels, whose nature is controlled by a joint random process. We derive the capacity for a broad class of probability distributions on the channels in the second stage. The results are substantiated at the example of the noisy drawing channel, where input sequences are randomly drawn with replacement and received with errors. We show that all information rates below capacity are achievable using a decoder which clusters output sequences according to their distance. Conversely, we prove that reliable transmission at rates above the capacity is not possible.

The last part of this dissertation treats cost constrained channels, which are described by directed graphs with labeled and costly edges. Our main results include an easy-to-use algebraic framework to formulate the precise asymptotic growth rate of the number of fixed and variable-length paths with a limited weight for arbitrary strongly connected graphs. We develop new theorems regarding the spectral properties of the cost-enumerator matrices of costly constrained channels and connect these to the theory of analytical combinatorics in several variables. This novel connection creates the basis for a variety of new results ranging from the precise characterization of the subsequence spectrum of a periodic sequence to the exact calculation of the maximum information rate per cycle of array-based DNA synthesis.

Contents

Chapter 1

Introduction

The amount of digital data stored on devices around the globe has exploded in recent years. It is expected that the current growth continues in years to come. This trend is accompanied by the ever-growing importance of digital data in personal and professional life. Not only private devices, such as smartphones or personal computers, but also professional systems, such as communication systems or robotics are build on a digital architecture that heavily relies on digital memories. This digitization comes with the necessity to cope with the increasing demands for capacity and reliability of data storage systems. To meet these demands, researchers and industry are permanently improving existing storage technologies and designing new systems with desirable properties. To date, most of the existing long-term data storage solutions rely on magnetic media. The entertainment industry, for example, archives more than three quarters of their data on either hard disk drives or tapes [Cou19]. Such storage devices offer reliable storage at maximum over a couple of years for the case of hard disk drives [Bea13] or over a few decades for the case of tapes [VB95]. In case the storage time exceeds the lifetime of the storage media, it is therefore inevitable to maintain the archive and copy the data in regular intervals to guarantee the integrity of the stored data. Novel storage technologies that may overcome the need for such maintenance are therefore the core of uncountable recent experiments and research studies.

In this dissertation, we analyze modern storage technologies from an information-theoretic perspective. Our main focus are DNA-based data storage systems. We propose and study novel channel models that abstract the main properties of DNA-based data storage. Being relatively general in nature, several of these models apply to a larger range of applications. Our particular attention lies in evaluating fundamental limits with respect to reliability and cost efficiency. We further study novel information-theoretic problems that model the DNA synthesis process and derive results that enable a cost and material efficient synthesis, while still maintaining high information rates.

1.1 Outline

Chapter 2 introduces and discusses DNA-based data storage systems. We review current DNA-storage experiments and present the structure of typical DNA storage systems with a separate discussion on information theory in DNA-based data storage.

Part I of this dissertation presents and analyzes a novel channel model that comprises the process of synthesizing, sequencing and reconstructing DNA strands. The model incorporates

insertion, deletion, and substitution errors within the strands and further allows losses of whole sequences. In comparison to previous work, we incorporate the fact that the DNA strands are stored in a disordered manner by considering the input and output of the channel to be represented by *sets* of sequences. In Chapter 3, we study zero-error codes for this channel and discuss its general properties and characteristics. In particular, we derive existential Gilbert-Varshamov-type bounds on the size of error-correcting codes, which guarantee the capability of correcting certain error patterns. These results are completed conversely by deriving upper bounds on the size of such codes. We further propose code constructions that are particularly designed for the channel. Among these are novel set-based constructions that avoid the usage of an index to combat the disorder of the sequences. Instead, we use a vectorial representation of a set that allows the employment of standard error-correcting codes. We proceed by shifting our attention towards coding schemes that employ indices in Chapter 4. A refined channel model that distinguishes between errors within the indices and the remaining part of the sequences is introduced, allowing to highlight the effect of errors within the different parts of the sequences. We derive existential and converse results under the restriction of using a code that employs an indexing scheme. We complement the results by presenting an explicit code construction which employs anchors that allows to correct errors within the indices with little redundancy.

In Part II we turn our attention towards a probabilistic channel, namely the *unordered parallel multinomial* channel. In contrast to standard communication scenarios, this channel is fed with a large number of parallel sequences. In a first stage, the sequences are arbitrarily permuted with each other. Afterwards, each of the resulting sequences passes through a multinomial channel that repeats the input sequence a given number of times and perturbs the result according to a symmetric channel. The number of repetitions in all channels follows a joint random distribution, meaning that the number of repetitions of different channels can possibly be correlated. Through the permutation, the original order of the output sequences is unknown to the receiver, which is a key challenge within this channel. In Chapter 5 we compute the Shannon capacity of this channel, i.e., the supremum of achievable information rates under vanishing error probability. This is achieved by deriving a converse bound based on the mutual information between the input and output and further by proving achievability of all information rates below capacity using an argument that is based on a random choice of the codebook. Chapter 6 deals with a related channel that is derived from an abstraction of DNA-based data storage. In a random fashion, the output sequences are drawn from a set of input sequences and are received, possibly with errors. We show that this channel is a degraded unordered parallel multinomial channel, establishing a converse on the capacity of this channel. We further prove that any rate below the one defined by the converse is achievable by analyzing a decoder that clusters the output sequences and decides on a codeword based on a novel measure of typicality between estimated clusters and a codeword.

Costly constrained channels are the object of study in Part III. These are described by a directed graph, whose edges are weighted and labeled. Such a graph defines an associated language comprised of words that are generated by paths of limited weight through this graph. The main property of interest of these systems is the size of the language, whose exponential growth rate is termed *capacity*. In Chapter 7 we establish a comprehensive theory, building on classical results of Perron and Frobenius, that allows to associate graph properties with characteristics of the singularities of the generating functions of the language size. This builds the first bridge between the literature on analytic combinatorics in several variables and costly graphs bringing a new perspective and a set of powerful results to the literature of costly constrained channels. We use this connection to deduce results on the asymptotic behavior of the size of the limited-cost

language. As part of our analysis, we identify and analyze key properties of graphs that result in a well-behaved expression of the capacity. We use these results in Chapter 8 to solve a problem related to cost-efficient DNA synthesis. That is, we study a popular array-based synthesis process and show that the maximum amount of information that can be synthesized per synthesis cycle is characterized precisely by the capacity of a costly constrained channel. We show how to construct the graph representing the system for arbitrary synthesis sequences, which allows to compute the maximum number of bits that can be synthesized per cycle. We further extend our results to the case, where the synthesized sequences need to fulfill certain constraints. Representing these constraints in form of a directed and labeled graph, we define a novel graph product that allows to compute the information capacity per synthesis cycle for constrained sequences. Finally, we show that it is possible to compute the number of subsequences of an arbitrary supersequence using costly constrained channels, implying results on the asymptotic behavior of the subsequence spectrum of arbitrary periodic sequences.

Further publications by the author that resulted from his work as a doctoral candidate are summarized in Chapter 9. Chapter 10 concludes this manuscript.

1.2 Notation

This section provides a concise overview of the notation that is used within this dissertation. For chapter-specific notation, we refer the reader to the introduction of the respective chapter and the glossary in Appendix B.

1.2.1 General Notation

We start by introducing the basic notation that is used throughout the dissertation. Sets are highlighted by calligraphic letters, such as $\mathcal{A}, \mathcal{B}$. For two sets $\mathcal{A}, \mathcal{B}$ we write $|\mathcal{A}|$ as the cardinality of $\mathcal{A}$, $\mathcal{A} \setminus \mathcal{B} = \{x : x \in \mathcal{A} \wedge x \notin \mathcal{B}\}$ as the set difference and $\mathcal{A} \times \mathcal{B} = \{(a, b) : a \in \mathcal{A}, b \in \mathcal{B}\}$ as their *Cartesian product*. We denote by $\mathbb{N}$, $\mathbb{N}_0$, and $\mathbb{Z}$ the sets of integer numbers, where the former consists of the numbers $\{1, 2, 3, \dots\}$, $\mathbb{N}_0$ additionally contains 0 and the latter also contains the negative integers. The set $[n] = \{1, 2, \dots, n\}$ contains all positive integer numbers up to $n \in \mathbb{N}$. The rational numbers are depicted by $\mathbb{Q}$, the real numbers by $\mathbb{R}$ and the complex numbers by $\mathbb{C}$. Σ_q is a finite alphabet with q elements. In particular, we write $\Sigma_2 = \{0, 1\}$ for binary sequences and $\Sigma_4 = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ for DNA sequences. Multisets are sets which can contain an element multiple times and are highlighted by $\{\{\bullet\}\}$.

Vectors are written as lowercase bold font letters and matrices as uppercase bold font letters. Their entries are depicted in standard font, such that, for example, $\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \Sigma_q^n$ is a vector of length n with symbols $x_i \in \Sigma_q$. For two vectors $\boldsymbol{x} \in \Sigma_q^n, \boldsymbol{y} \in \Sigma_q^m$ we write $(\boldsymbol{x}, \boldsymbol{y})$ as the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}$ which has length $n + m$. The number of non-zero symbols in a vector $\boldsymbol{x} \in \Sigma_q^n$ are called the *Hamming weight* of $\boldsymbol{x}$ and denoted by $\mathsf{wt}_{\mathsf{H}}(\boldsymbol{x}) = |\{i \in [n] : x_i \neq 0\}|$. For another vector $\boldsymbol{y} \in \Sigma_q^n$ of the same length, we define $d_{\mathsf{H}}(\boldsymbol{x}, \boldsymbol{y}) = |\{i \in [n] : x_i \neq y_i\}|$ as their *Hamming distance*, i.e., the number of positions, in which the vectors disagree.

Throughout this dissertation, we denote the binary logarithm of a real number $a \in \mathbb{R}^+$ by $\log(a)$, the natural logarithm by $\ln(a)$ and the logarithm with respect to base $b \in \mathbb{R}^+$ by $\log_b(a)$. For an integer $n \in \mathbb{N}$, we write $n! = n \cdot (n-1) \dots 2 \cdot 1$ as the factorial. For $m \in \mathbb{N}$, $m \leq n$, the binomial coefficient is denoted by $\binom{n}{m} = \frac{n \cdot (n-1) \dots (n-m+1)}{m!}$.

We denote random variables by standard letters, such as x and their realization is usually highlighted with a typewriter font, such as $\mathtt{x}$. The probability of the event $x = \mathtt{x}$ is denoted by $\mathsf{Pr}\left(x = \mathtt{x}\right)$ and, where it is clear from the context, we abbreviate it with $\mathsf{Pr}\left(\mathtt{x}\right)$. We denote the entropy of a random variable by $H(x)$ and the conditional entropy and mutual information of two variables by $H(y|x)$ and $I(x;y)$, respectively. The q-ary entropy function is denoted by $H_q(p) = -(1-p)\log_q(1-p) - p\log_q(\frac{p}{q-1})$.

1.2.2 Asymptotic Statements

For the asymptotic behavior of functions, we use the Bachmann-Landau notation, i.e., for $f(n), g(n) : \mathbb{N} \mapsto \mathbb{R}$, we write

- $f(n) = o(g(n))$, if $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0$,
- $f(n) = \omega(g(n))$, if $\lim\limits_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| = \infty$,
- $f(n) = O(g(n))$, if $\limsup\limits_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| < \infty$,

Based on this, we may slightly abuse notation and may, for example, write $f(n) = g(n) + O(h(n))$ to denote $f(n) - g(n) = O(h(n))$. Further, we use inequalities involving this notation as follows. Writing, for instance, $f(n) \leq g(n) + O(h(n))$ means that $f(n) \leq g(n) + \widetilde{h}(n)$, for some function $\widetilde{h}(n)$ with $\widetilde{h}(n) = O(h(n))$.

Chapter 2

Archival Data Storage in DNA

Data storage in synthetic DNA molecules is a novel technology for archival storage of digital data. Paved by substantial progress in synthesis and sequencing technologies, it has attracted significant attention due to recent demonstrations of the viability of storing information in macromolecules. Offering unique properties and advantages over alternative storage systems, it has become a competitive archiving technology [Mil+18]. To date, DNA-based archival data storage is being explored by both industrial and scientific teams in various aspects. First, basic features required for digital storage in DNA are developed and tested. Such features include reading and writing data, but also more elaborate processes such as random access, rewriting data or erasing data. Second, efficiency aspects are core to make the storage technology commercially competitive. Here, the main bottlenecks are the synthesis and sequencing of DNA molecules, i.e., reading and writing of data. Finally, reliability is one of the main aspects that is considered for DNA-based data storage, because the reading and writing process, as well as the storage of the DNA molecules is prone to errors. Therefore, adequate mechanisms to protect the data from damage in its integrity or loss of some parts need to be developed and analyzed.

The main advantages of DNA-based storage over classical storage technologies are very high data densities due to the fact that data is stored on a molecular level. This allows theoretical storage densities of up to 215 petabytes per gram of DNA [EZ17]. DNA-based storage further provides long-term reliability without electrical supply due to the stable nature of the DNA molecules. From the study of fossils, it is known that the DNA of some organisms has a half-life of up to 500 years [All+12] and experiments [Gra+15] have shown that also in synthetic DNA it could be possible to store data thousands of years without loss of data. Finally, the information stored in DNA-based archives is easily replicable. This is by means of polymerase chain reactions that allow to create thousands of copies of DNA strands using biochemical processes.

On the other hand, DNA-based archival storage still features some difficulties that need to be overcome before it becomes commercially competitive. One aspect is the high costs associated with synthesizing and sequencing DNA. Currently, the price for sequencing 10^6 DNA bases is roughly US$0.01 [Wet], which relates to US$0.04 per Megabyte of digital data, assuming a reading rate of 2 bit per DNA base. Even more expensive is the synthesis of DNA, which can cost hundreds of dollars per Megabyte [Hec18], depending on the technology applied. However, given the current focus that is put towards more and more efficient DNA synthesis [Ant+20] and improved storage architectures [Tab+20], it is not unlikely that these costs will drop significantly over the next years. Another challenge for DNA-based data storage is the speed of the sequencing and synthesis process. While magnetic tapes can be read and written at several Megabytes per second, typical

sequencing and synthesis processes take several minutes to hours. Therefore, DNA is currently considered as a possible candidate for archival data storage, where data is written and read in non time-critical situations. For more background on the synthesis and sequencing process, we refer the reader to the essay [Car13] and the surveys [FL18; Hao+21; KC14].

2.1 History of DNA Storage Experiments

The rapid growth of research related to DNA-based storage has been guided by many experiments that have demonstrated the viability and the open problems of DNA-based data storage. In this section, we review some of these experiments and highlight their contributions with respect to archival storage systems. Many scientists agree that the first mentioning of data storage in macromolecules dates back to Richard Feynmans famous speech "*There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics*" in 1959 [Fey59]. To that date, manipulating molecules and arranging them such that they contain predetermined information was an idea that should be brought to life few decades later. In another article, Neiman [Nei65] discussed microminiaturization in electronics and also published considerations about the feasibility of information storage in DNA molecules. Three decades later, Eric Baum [Bau95] concretized some aspects of data storage in DNA. As an example, he already mentioned that the DNA strands will be replicated in many copies, which is the case for almost all current experiments.

In his artwork *Microvenus* [Dav96], Joe Davis incorporated 35 bits of digital data inside the DNA of living *Escherichia coli* bacteria. The data represented a 7×5 pixel bitmap of the Microvenus image, where each pixel was either white or black. In the context of secrecy, Clelland et al. [CRB99] stored a message of 23 letters in a synthetic DNA oligonucleotide of 69 nucleotides. Their research was motivated by microdots, which are in size reduced photographs that were used in the Second World War to convey secret messages. Due to the enormous length of human DNA, they believed that a short synthetic DNA message could be well hidden within the vast amount of biologic human DNA. Two years later, Bancroft et al. published their experiment and ideas regarding archival storage using DNA [Ban+01]. They highlighted important considerations, such as the inherent interest of humanity in writing and reading DNA and the ease of mitigating possible losses of information by replicating the molecules many times. In their experiment, they stored two lines of text with a total of 109 characters. In the following years, numerous studies showed how to synthesize digital information into DNA. Among those, Gustafsson published a successful experiment involving art by storing the poem *Tomtem* by Viktor Rydberg in DNA [Gus09]. Writing artificial watermarks, each having a length of roughly 1,000 nucleotides, into the DNA of Mycoplasma mycoides, Gibson et al. [Gib+10] further showed that synthesis, assembly and transplantation of synthetic genomes is possible, reporting the largest project to that date.

Among the first large-scale experiments was that conducted by Church et al. in 2012 [CGK12], where 0.66 MB of digital information were stored, split onto many short DNA strands. They encoded the binary data such that each bit is either mapped to A or C, if the bit is zero and to G or T, if the bit is one. This flexibility allowed the resulting DNA strands to be designed such that they have preferable properties for reading and writing. While this comes at the cost of redundancy, they were able to avoid errors in synthesis and sequencing, counting a total number of 10 bit errors after sequencing and decoding the whole stored archive. Shortly after, Goldman et al. [Gol+13] used a similar set up to store 0.65 MB and managed to read the data without errors. They addressed scalability and reliability of DNA-based data storage systems. To ensure

Table 2.1: Summary of parameters in large-scale DNA storage experiments. The data size is depicted as the information content of the data *after* compression. The strand length L counts the number of consecutive nucleotides used to store information. This includes the index and possible redundancy from error-correcting schemes. However, we exclude the length of eventual primers that are appended to the strands. We further present the number of DNA strands M on which the data is synthesized, as well as a quantity called *density*, which will be elaborated on in Section 3.1.

Work	Data Size	Strand Length L	Strands M	Density $\beta = \frac{\log_4 M}{L}$
[CGK12]	0.66 MB	115	54,898	0.0685
[Gol+13]	0.75 MB	117	153,335	0.0736
[Gra+15]	83 KB	117	4,991	0.0525
[Yaz+15b]	0.017 MB	960	32	0.0026
[Bor+16]	0.15 MB	120	45,652	0.0645
[Bla+16]	22 MB	190	900,000	0.0521
[EZ17]	2.14 MB	152	72,000	0.0531
[YGM17]	0.003 MB	1000	17	0.0020
[Org+18]	200.2 MB	$150 - 154$	13,448,372	$0.0769 - 0.0789$
[Cha+19]	0.22 MB	100	12,026	0.0939
[Lop+19]	1.67 MB	110	111,499	0.0762
[Ana+19]	6.4 MB	151	172,000	0.0572
[Ant+20]	1.3 MB	60	196,596	0.147
[Cha+20]	0.01 MB	108	1,466	0.0487
[Cho+20]	0.135 MB	120	5299	0.0515
[Pan+21]	0.36 MB	156	11,826	0.043

error-free decoding, the DNA strands have been encoded such that each two consecutive DNA strands overlap in 75 out of a total of 100 base pairs. Thus, each data segment was synthesized with 4-fold coverage. To identify the position of each DNA strand in the whole archive, a short index was appended to each strand.

Manifold experiments followed that addressed different aspects of archival data storage, such as reliability, scalability, random access and rewriting of data. A method for random access, i.e., accessing a specific file or part inside the archive without reading the full archive has been proposed in [Yaz+15b]. Yazdi et al. used specific primers that allow for amplification of selected DNA strands. They carefully designed the primers such that they have desirable properties, such as a large Hamming distance and a constant GC content. They also showed that rewriting of data is possible using DNA editing and mutating techniques. Using an exclusive-or operation at the strand level for error correction, [Bor+16] showed in their experiment that controllable redundancy can help to improve the archive's data density. Grass et al. [Gra+15] employed a concatenated coding scheme using inner and outer Reed-Solomon codes and simulated an aging process, hinting that DNA-based archives could provide reliable storage for several thousands of years. Another successful experiment storing 22 MB with concatenated codes has been reported by [Bla+16]. The binary data was modulated to oligos with a short inner code that consists of 5 nucleotides and a has dimension of 8 bits. For error detection, the strands were further encoded with a cyclic redundancy check code. An outer Reed-Solomon code then was used to protect against loss of sequences or burst errors. Erlich and Zielinski [EZ16; EZ17] opted to use outer fountain codes for their experiment. While the coding scheme did not allow for error correction inside the strands, an inner Reed-Solomon code was used for error detection. This way, erroneous strands could be detected and missing oligos were corrected using the outer fountain code. Most of the experiments to that date relied on accurate sequencing technologies and error detection together with a high reading coverage. Potential sequencing errors were detected using an inner code and erroneous strands were discarded. In contrast, Yazdi et al. [YGM17] used a nanopore sequencer, which has a higher error rate, however allows for a portable system. The resulting larger number of errors was approached by aligning multiple reads of the same strand and performing a majority decision of the aligned parts. The remaining errors were corrected by a new error-correcting scheme for deletions. Organick et al. [Org+18] stored 200 MB using an outer Reed-Solomon code, performing the largest published experiment to date. Their decoder acted in several stages, clustering the sequenced strands based on similarity, then reconstructing the clusters via sequence alignments and finally decoding the inner and outer code.

Most of the works to that date mainly focused on write information rates, i.e., the total number of data bits divided by the total number of synthesized nucleotides. In contrast, Chandak et al. [Cha+19] explored the trade-off between the write rate and read rate, i.e., the total number of information bits obtained after decoding divided by the total number of sequenced nucleotides. In their experiment, they employed low-density parity check codes and inner marker codes to maintain synchronization. Lopez et al. [Lop+19] presented a strategy optimized for nanopore sequencing. They concatenated several shorter DNA oligonucleotides to longer strands of about 5000 basepairs which allows for faster sequencing. An end-to-end automatic system from data storage to data retrieval has been presented in [Tak+19]. Another DNA-based storage architecture on physically separated spots of dehydrated DNA has been investigated in [New+19]. Antkowiak et al. explored low-cost synthesis for DNA-based data storage [Ant+20]. The resulting higher error rates were overcome using a sufficiently strong concatenation of Reed-Solomon codes together with clustering and multiple sequence alignment. In [Cha+20], the decoder of an inner convolutional

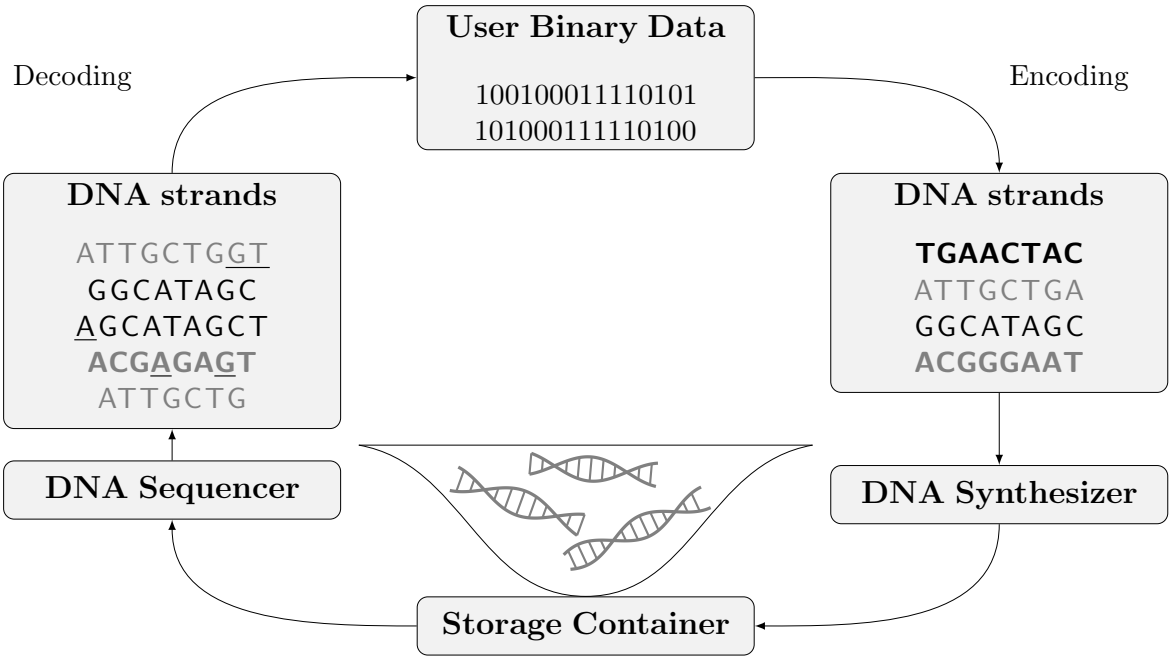


Figure 2.1: Illustration of a DNA-based data storage system. The text decoration of the sequenced strands highlight the original sequence. Errors in the sequences, which either occurred during synthesis, storage, or sequencing, are underlined.

code was adapted to match the channel imposed by nanopore sequencing. It has been shown that it is possible to use soft information from the nanopore readout to decode the convolutional code with an appropriate Viterbi decoder. Image processing on pictures stored in DNA has been investigated in [Pan+20; Pan+21]. Using little redundancy in the encoding procedure, it has been shown that machine learning algorithms can be used for error correction. A summary of the parameters used in recent large-scale experiments is presented in Table 2.1.

2.2 DNA-Based Storage Systems

Most recent DNA-based data storage systems exhibit a similar structure as the one we present and comment on in the following. An illustration of such systems is displayed in Figure 2.1.

A DNA-based data storage system usually consists of the following components: an encoder, a synthesizer, a storage medium, a sequencer, and a decoder. First, binary user data is encoded via an appropriate encoding algorithm to a set of multiple DNA strands. This encoder maps the binary data to vectors over the alphabet $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ such that each vector represents a DNA strand that will be synthesized later. Note that crucially, the encoder has to include appropriate mechanisms that allow to restore the original order of the DNA strands and correct possible errors that arise during synthesis or sequencing. The encoded strands are then synthesized by a synthesis machine that physically produces the DNA strands, as described by the encoder. Common synthesis processes synthesize each strand many times, resulting in many copies of the original strands. Additionally, the DNA strands can be duplicated using polymerase chain reactions. After synthesis, the resulting strands are transferred into a storage container that

preserves the strands and protects them from potential damage or loss. When accessing the data, the stored strands are sequenced, yielding reads of the original synthesized strands. It is possible that an original strand is read multiple times due to the fact that multiple copies of each strands are present in the storage medium. Using the read DNA sequences, it is the task of the decoder to estimate the original binary user data using an appropriate algorithm.

While the system described above is one of the most common ones to date, there have been proposals of a variety of other DNA systems, which we highlight shortly. In [Tab+20], information is encoded via nicking existing DNA strands at certain positions. This way, costly DNA synthesis is circumvented, allowing for more cost-efficient DNA-based data storage systems. Another approach is to modify the existing DNA of living organisms, such as bacteria. This way it is possible to encode digital data into the modified parts, which has been verified experimentally in, e.g., [KMC17; Shi+17; TL18]. Other works have pursued the idea of storing information also in the density of certain molecules. More precisely, composite letters that use mixtures of DNA sequences have been proposed in [Ana+19; PYA21], resulting in fewer synthesis cycles. Recently, it has been shown [Tab+21] that it is possible to extend the alphabet of nucleotides used in DNA molecules, improving the density and recording time of DNA-based storage systems.

2.3 Information Theory in DNA-Based Data Storage

From a coding- and information-theoretic perspective, there are several aspects of DNA-based data storage that differentiate such systems from conventional data storage systems. The most studied aspect is reliability, i.e., forward error correction to combat possible errors that arise during synthesis, sequencing, and aging. There are several error models that are relevant for DNA-based data storage. To start with, zero-error insertion- and deletion-correcting codes have been studied extensively for the case of unique decoding [BGH17; BGZ18; GS19; GW17; Hae19; HF02; Lev65; Lev66; Lev67; Maz17; SB19; Ten84; Yaz+18] and list decoding [GHS20; HSS18; HY20; LL17; Wac18]. Codes protecting against combinations of insertion and deletion errors with substitution errors are studied in [Cai+21; Sma+20; Son+21] While the previous works mainly employ non-linear codes, linear codes have been discussed in [Che+21; CST21]. Segmented channels depict the case with occurrence of at most a given number of errors per segment and have been discussed in [AVF18; HB21; LM10]. Especially, during *in vivo* DNA-based data storage, insertion and deletion errors can occur in bursts, i.e., consecutive positions, which has been investigated in [Che+14; GYM18; LP20; Sch+17]. On the other hand, studies on codes over probabilistic insertion and deletion channels date back to the early works of [Dob67; Gal61]. More recently, also due to their relevance in DNA-based data storage, new codes and decoders have been proposed for such channels, such as low-density parity check codes [BSW10; DM01; SHY19], polar codes [KK19; Tal+19], or convolutional codes [BF15; MB09]. For a comprehensive survey on channels with insertion and deletion channels, we refer the reader to [Mit09].

Due to the fact that DNA-based storage systems contain multiple copies of each strand, an important line of work is that on sequence reconstruction, which, for the combinatorial worst-case setting, has originally been introduced by Levenshtein [Lev01]. In his paradigm, a sequence is repeatedly transmitted over an erroneous adversarial channel and the classical goal is to analytically quantify, as a function of the sequence length, the number of sequences that are required to guarantee correct reconstruction of the transmitted sequence with zero-error probability. Recent work includes sequence reconstruction for coded sequences [Abr+19; GY18; Sal+17] and extension

to a broader class of channels, also allowing substitution errors [SY19] or duplications [YS18]. In the computer science community, the sequence reconstruction is often studied in a probabilistic setting with the goal of analyzing the number of sequences required for possible reconstruction with high probability [AKN14; BLS20; Che+20; KM05; MPV14]. Reconstruction from shotgun sequences, i.e., short reads of longer DNA strands, has been discussed in [Ach+15; KPM16; MBT13; MY20] and reconstruction from compositions of sequence prefixes and suffixes that were generated by mass spectrometry readouts is subject of [GPM20; GPM21; PGM19]. More recently, the trace reconstruction problem has also been formulated for a fixed number of sequences with a larger focus on algorithmic aspects [Sab+20; Sri+19; Sri+20; SYY20]. Finally, in [AVF18] Varshamov-Tenegolts codes have been proposed for error correction over multiple channels that introduce a fixed number of deletions.

Certain DNA molecules exhibit a more stable structure when certain patterns of nucleotides are avoided [Ros+13; Xu+21]. For example, balancing the number of times the bases G and C occur in a DNA strand is possible using constrained codes [SIC19; Son+18; Wan+19]. The runlengths of homopolymers can be limited with appropriate encoding mechanisms [Kov19a; SIC18], which have been extended with additional error-correction capabilities [LK21; Ngu+21]. For topological DNA-based data storage, [Aga+20] studied constrained codes with runlength limitations.

During DNA replication, it is not uncommon that certain substrings of the DNA are erroneously repeated and inserted into the DNA strand, causing duplication errors. Zero-error codes for duplication errors have been studied in [Che+18; DA10; Jai+17a; Jai+17b; Kov19b; KT18a; LJW18; LWY17; LWY19; Tan+19], and have also been addressed in a reconstruction scenario [YS18; YS21]. Classical coding theorems and the channel capacity for probabilistic duplication models have been analyzed in [Mit08; RA13].

One unique characteristic of DNA-based data storage is the disordered nature in which the strands are stored in the medium. Information-theoretic studies addressing this property include [Gab+20; Hec+17; KT18b; Len+20c; Len+20d; MSG15; SCSI19; SH19; SH21; SRB20; SRB21; WM21], addressing error correction over unordered vectors. Error-correcting codes and information theory for permutation channels have been discussed in [LSY17; Mak18; WG08], together with explicit code constructions, which have been presented in [Hof+13].

There are manifold other studies treating coding- and information-theoretic aspects of DNA-based archival storage and the list of associated publications is long. However, their discussion is beyond the scope of this dissertation and is treated in several recent survey papers, such as [Car+19; CNS19; HA21; TBK20; Xu+21; Yaz+15a].

Part I

Combinatorial DNA Storage Channel

Chapter 3

Zero-Error Codes for the Combinatorial DNA Storage Channel

DNA is a medium for digital data storage that is fundamentally different from conventional storage systems due to its following unique properties. First, DNA strands are stored and retrieved in an unordered fashion.[1] Next, the data is synthesized on many relatively short strands due to limitations from the synthesis technologies. Finally, the synthesis and sequencing is prone to errors, such as insertion, deletion, and substitution errors. These novel properties fueled theoretical investigations of channel models and reliability in DNA-based data storage systems. Among those, most related to our study are codes over unordered multisets of sequences that have been studied recently [KT18b] under errors that affect whole sequences. In that channel model, a sequence can either be inserted, deleted or completely corrupted to another sequence. Similarly, reconstruction of DNA sequences from their unordered sequence profiles has been analyzed in [KPM16]. It has been shown that, with appropriate error correction mechanisms, it is possible to combat errors within the profiles, such as errors within the sequences or a loss of some of the profile pieces.

We begin this chapter by introducing in Section 3.1 a novel combinatorial channel model for sets of DNA sequences that incorporates the main properties of DNA-based data storage systems. That is, we model the channel input and output by *sets* of sequences, incorporating the unordered nature in which the strands are synthesized and sequenced. Next, the sequences can be perturbed by errors, or may be completely lost. We proceed in Section 3.2 with highlighting the technical contributions and challenges arising when analyzing the proposed channel model. In Section 3.3, we prove existence of zero-error codes, i.e., codes that guarantee error correction under the combinatorial channel model, over that channel, using a proof technique that was originally introduced by Gilbert and Varshamov [Gil52; Var57]. Conversely, in Section 3.4, we prove lower bounds on the redundancy using sphere-packing arguments. We compare the resulting bounds and identify parameter regimes in which both bounds are close and others, in which there is a gap. Finally, we present code constructions suitable for the presented channel model in Section 3.5.

The results in this chapter have previously been published in [Len+18; Len+20d].

[1] There are studies [Yaz+15b; YGM17] that have developed methods for random access and for sequencing of specific strands. This was accomplished by designing primers that are appended to the DNA strand. Here we are studying the *raw* system without the usage of such additional primers.

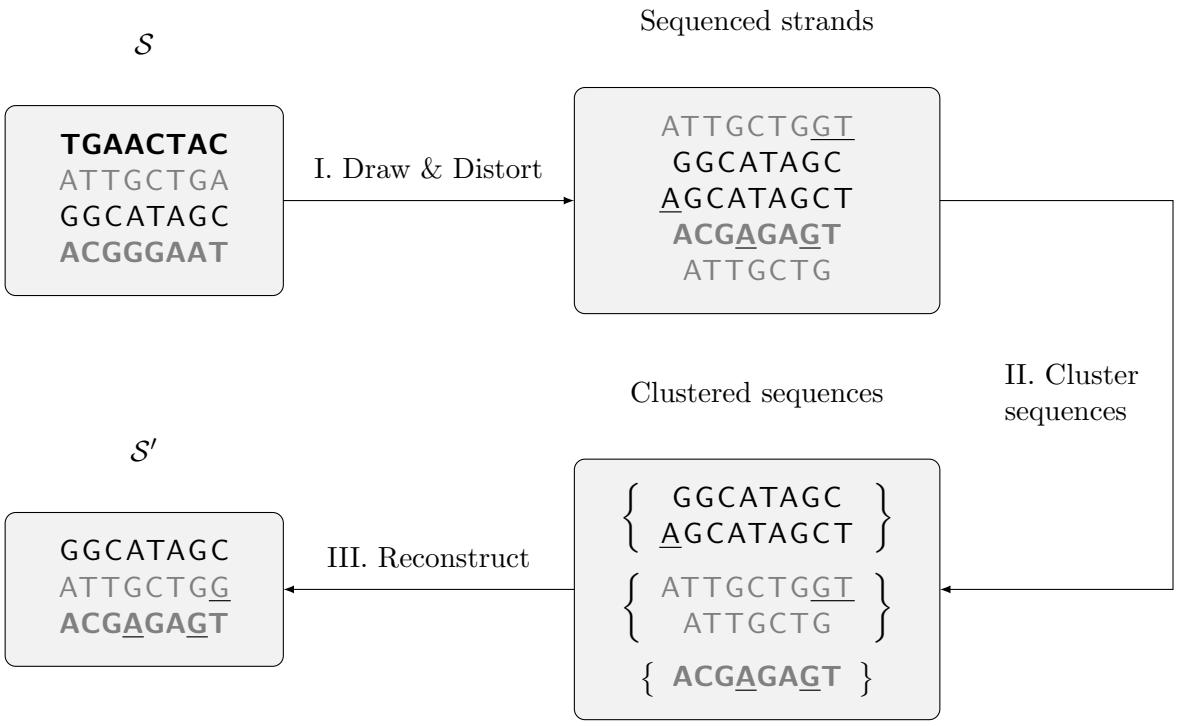


Figure 3.1: DNA storage channel model. Sequences with the same text decoration stem from the same original sequence. Errors are underlined.

3.1 Channel Model

Based on the main aspects of recent DNA storage experiments, we present a combinatorial channel that models the relationship between stored and received DNA sequences. To start with, DNA consists of four types of nucleotides: adenine (A), cytosine (C), guanine (G), and thymine (T). A single DNA *strand*, also called an *oligonucleotide*, or *sequence* is an ordered sequence of some combination of these nucleotides. Although DNA consists of two complementary strands, for the purposes of digital data storage usually we only consider a single strand, since the complementary strand does not contain additional information. We therefore view a DNA strand as a vector over the alphabet of nucleotides $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$. The channel comprises the process of synthesizing, storing and sequencing DNA strands and is visualized in Figure 3.1. In a DNA-based data storage system, data is synthesized and stored in an unordered *set*

$$\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \subseteq \Sigma_q^L,$$

of M distinct DNA *sequences* $\boldsymbol{x}_i \in \Sigma_q^L$, i.e. $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for $i \neq j$. Each sequence $\boldsymbol{x}_i$ has length L. Here and in the rest of Part I whenever we write the set $\mathcal{S}$ we assume it is a set of M sequences as defined above. We refer to the vectors $\boldsymbol{x}_i$ by *sequences* or *strands* and to $\mathcal{S}$ by data *sets* or *words*. Representing data words as unordered sets is inherently natural, due to the following two reasons. First, any information about ordering of the data sequences is lost during the storage and second, in the reading process it is not easily possible to distinguish exactly how many times each sequence was stored, since the sequences are multiplied in the storage medium and not necessarily

all of them are read. For more details on the channel model, see [HMG19; SH21; Yaz+15a].

Any such stored data set $\mathcal{S}$ of M sequences is a possible input of the DNA storage channel. Hence, the input space, which comprises all possible data sets is denoted by

$$\mathcal{X}_M^L = \{\mathcal{S} \subseteq \Sigma_q^L : |\mathcal{S}| = M\}.$$

The DNA storage channel can be split into the three following stages, as visualized in Figure 3.1.

I. Random sequences are drawn with replacement from the storage medium $\mathcal{S}$ and sequenced, possibly with substitution, insertion or deletion errors.

II. The sequenced strands are clustered according to their Levenshtein distance or other similarity measures.[2]

III. The clustered sequences are reconstructed by performing an estimate $\boldsymbol{x}'$ for each cluster, resulting in the received estimates $\mathcal{S}'$. If two or more reconstructions result in the same estimate $\boldsymbol{x}'$, we only output a single sequence $\boldsymbol{x}'$ to avoid possible duplicates of a single stored sequence. Therefore, $\mathcal{S}'$ is a set with distinct elements.

In this work we consider the combination of the above three stages, from the stored sequences $\mathcal{S}$ to the reconstructed sequences $\mathcal{S}'$, as the DNA storage channel. Note that in principle it is also possible to exclude the reconstruction step from the channel model. In this case however, a probabilistic channel model is more appropriate, which we will discuss in detail Chapter 6 of this dissertation. Each sequence $\boldsymbol{x} \in \mathcal{S}$ is therefore either reconstructed correctly, without errors ($\boldsymbol{x} \in \mathcal{S}_\mathsf{C}$), never drawn or its cluster is not identified and thus lost in the storage medium ($\boldsymbol{x} \in \mathcal{S}_\mathsf{L}$), or reconstructed with errors ($\boldsymbol{x} \in \mathcal{S}_\mathsf{E}$), where $(\mathcal{S}_\mathsf{C}, \mathcal{S}_\mathsf{L}, \mathcal{S}_\mathsf{E})$ is a partition of $\mathcal{S}$, i.e., $\mathcal{S}_\mathsf{C} \cap \mathcal{S}_\mathsf{L} = \emptyset$, $\mathcal{S}_\mathsf{C} \cap \mathcal{S}_\mathsf{E} = \emptyset$, $\mathcal{S}_\mathsf{L} \cap \mathcal{S}_\mathsf{E} = \emptyset$, $\mathcal{S}_\mathsf{C} \cup \mathcal{S}_\mathsf{L} \cup \mathcal{S}_\mathsf{E} = \mathcal{S}$.

According to the above three cases, we thus associate the following four parameters $(s, t, u)_\mathbb{T}$ that characterize the DNA storage channel. We denote by s the maximum number of sequences that are never drawn (or whose clusters are not identified), by t the maximum number of sequences that have been reconstructed with errors, and by u the maximum number of errors of type $\mathbb{T}$ in each of the latter. Notice that we naturally assume throughout the subsequent discussion that $s + t \leq M$ and $t > 0$ if and only if $u > 0$. Typical error types $\mathbb{T}$ after the reconstruction step are various combinations of insertions, deletions and substitutions, where the latter two are the most prominent ones in DNA storage systems [HMG19].

In order to define a precise channel model, we proceed with defining the error balls, i.e., the sets of words that can be obtained through the channel. We start with the characterization of point errors inside single sequences.

Definition 3.1. *The error ball $B^\mathbb{T}(\boldsymbol{x}, u)$ of radius u around a sequence $\boldsymbol{x} \in \Sigma_q^L$ is defined to be the set of all possible outcomes $\boldsymbol{x}' \in B^\mathbb{T}(\boldsymbol{x}, u)$, after u (or fewer) errors of type $\mathbb{T}$ in $\boldsymbol{x}$. Possible types of errors are insertions ($\mathbb{I}$), deletions ($\mathbb{D}$), substitutions ($\mathbb{S}$), or combinations of the above, denoted by, e.g., $\mathbb{ID}$ for the case of insertions and deletions. We use the abbreviation $\mathbb{L} \triangleq \mathbb{IDS}$ for insertions, deletions, and substitutions. Similarly, we define the error sphere $S^\mathbb{T}(\boldsymbol{x}, u)$ as the set*

[2] This technique has been used in [Org+18], exploiting the fact that sequences are drawn several times. Other groups have either clustered the sequences according to their indices (as in [Gra+15]), or used specifically designed primers to control, which strands shall be read [YGM17]. Some works simply discarded sequences of incorrect length to avoid the challenging task of dealing with insertion and deletion errors.

of possible results after exactly u errors of type $\mathbb{T}$. For uniform error balls and spheres, where the size does not depend on the center $\boldsymbol{x} \in \Sigma_q^L$ we use the abbreviations $B^{\mathbb{T}}(L,u) \triangleq |B^{\mathbb{T}}(\boldsymbol{x},u)|$ and $S^{\mathbb{T}}(L,u) \triangleq |S^{\mathbb{T}}(\boldsymbol{x},u)|$, respectively. In particular we have

- $S^{\mathbb{I}}(L,u) = \sum_{i=0}^{u} \binom{L+u}{i}(q-1)^i$ *(c.f. [Lev01; Lev74]),*
- $B^{\mathbb{I}}(L,u) = \sum_{i=0}^{u} S^{\mathbb{I}}(L,i)$,
- $S^{\mathbb{S}}(L,u) = \binom{L}{u}(q-1)^u$,
- $B^{\mathbb{S}}(L,u) = \sum_{i=0}^{u} \binom{L}{i}(q-1)^i$.

Note that for the case of deletions, such a closed-form expression does not exist, since the size of the deletion ball and sphere depends on the center $\boldsymbol{x}$. The following example illustrates the definitions of error balls for different error types.

Example 3.2. *Consider the sequence $\boldsymbol{x} = (\mathsf{AC}) \in \Sigma_4^2$ of length $L = 2$ and a single error, $u = 1$. The substitution error ball is given by $B^{\mathbb{S}}(\boldsymbol{x},1) = \{(\mathsf{AC}),(\mathsf{CC}),(\mathsf{GC}),(\mathsf{TC}),(\mathsf{AA}),(\mathsf{AG}),(\mathsf{AT})\}$. Similarly, the deletion ball around $\boldsymbol{x}$ is given by $B^{\mathbb{D}}(\boldsymbol{x},1) = \{(\mathsf{AC}),(\mathsf{C}),(\mathsf{A})\}$ and the deletion ball around $\boldsymbol{y} = (\mathsf{CC})$ is given by $B^{\mathbb{D}}(\boldsymbol{y},1) = \{(\mathsf{CC}),(\mathsf{C})\}$, where the former has size 3 and the latter has size 2. The insertion sphere around the center $\boldsymbol{x}$ is given by the set $S^{\mathbb{I}}(\boldsymbol{x},1) = \{(\mathsf{AAC}),(\mathsf{CAC}),(\mathsf{GAC}),(\mathsf{TAC}),(\mathsf{ACC}),(\mathsf{AGC}),(\mathsf{ATC}),(\mathsf{ACA}),(\mathsf{ACG}),(\mathsf{ACT})\}$.*

In a similar fashion it is possible to define the error ball of a data set, as the set of possible received sets after the DNA storage channel.

Definition 3.3. *For $\mathcal{S} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M\} \in \mathcal{X}_M^L$, the error ball $B^{\mathbb{T}}(\mathcal{S},s,t,u)$ is defined to be the set of all possible received sets $\mathcal{S}'$ after s (or fewer) sequences have been lost and t (or fewer) sequences of the remaining sequences have been distorted by u (or fewer) errors of type $\mathbb{T}$ each.*

More precisely, let $\mathsf{Part}_{s,t}(\mathcal{S})$ be the set of all partitions $(\mathcal{S}_\mathsf{C},\mathcal{S}_\mathsf{L},\mathcal{S}_\mathsf{E})$ of $\mathcal{S}$ with $|\mathcal{S}_\mathsf{L}| \le s$, $|\mathcal{S}_\mathsf{E}| \le t$. We then define $B^{\mathbb{T}}(\mathcal{S},s,t,u)$ to be

$$B^{\mathbb{T}}(\mathcal{S},s,t,u) = \left\{ \mathcal{S}' = \bigcup_{i=1}^{M} \begin{cases} \{\boldsymbol{x}_i\}, & \text{if } \boldsymbol{x}_i \in \mathcal{S}_\mathsf{C}, \\ \emptyset, & \text{if } \boldsymbol{x}_i \in \mathcal{S}_\mathsf{L}, \\ \{\boldsymbol{x}_i'\}, & \text{if } \boldsymbol{x}_i \in \mathcal{S}_\mathsf{E} \end{cases} \middle| \, \boldsymbol{x}_i' \in B^{\mathbb{T}}(\boldsymbol{x}_i,u), (\mathcal{S}_\mathsf{C},\mathcal{S}_\mathsf{L},\mathcal{S}_\mathsf{E}) \in \mathsf{Part}_{s,t}(\mathcal{S}) \right\}.$$

We denote by $\mathcal{S}_\mathsf{E}' = \{\boldsymbol{x}_i' : \boldsymbol{x}_i \in \mathcal{S}_\mathsf{E}\}$ the set of erroneous received sequences, which satisfies $|\mathcal{S}_\mathsf{E}'| \le |\mathcal{S}_\mathsf{E}|$. Similarly we define $S^{\mathbb{T}}(\mathcal{S},s,t,u)$ to be the set of all words $\mathcal{S}' \in B^{\mathbb{T}}(\mathcal{S},s,t,u)$ obtained from $\mathcal{S}$ by a loss of exactly *s sequences and* exactly *u errors of type $\mathbb{T}$ in each of t sequences.*

The erroneous sequences $\boldsymbol{x}_i'$ are not necessarily distinct from each other or from the correct sequences in $\mathcal{S}_\mathsf{C}$ and therefore it is possible that two erroneous sequences or one error-free and one erroneous sequence agree with one another, resulting in a loss of a sequence. The number of distinct received sequences $|\mathcal{S}'|$ therefore satisfies $M - t - s \le |\mathcal{S}'| \le M$.

Example 3.4. *Consider the example in Figure 3.1 for the DNA storage channel with $M = 4$ stored sequences, $\boldsymbol{x}_1 = (\mathsf{TGAACTACG})$, $\boldsymbol{x}_2 = (\mathsf{ATTGCTGAA})$, and $\boldsymbol{x}_3 = (\mathsf{GGCATAGCT})$, $\boldsymbol{x}_4 = (\mathsf{ACGGGAATC})$ each of length $L = 8$, i.e., $\mathcal{S} = \{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4\} \in \mathcal{X}_4^8$. The sequenced strands are clustered and reconstructed, resulting in the three estimates $\boldsymbol{y}_1 = (\mathsf{GGCATAGCT})$, $\boldsymbol{y}_2 = (\mathsf{ATTGCTGGT})$, and $\boldsymbol{y}_3 = (\mathsf{ACGAGAGTC})$. The received set is therefore $\mathcal{S}' = \{\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{y}_3\}$.*

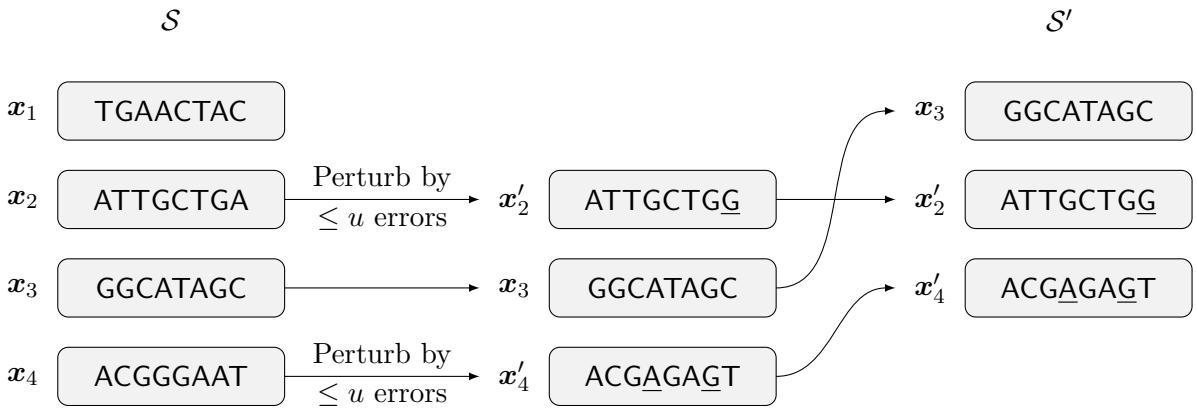


Figure 3.2: Illustration of the $(s,t,u)_{\mathbb{T}}$ channel model. Out of the M input sequences, t sequences are perturbed by u errors of type $\mathbb{T}$ each. Out of the remaining sequences, s are lost and not observed in the received set. Errors are underlined. In this example $s=1, t=2, u=2$, and $\mathbb{T}=\mathbb{S}$. The vectors in both the input $\mathcal{S}$ and output $\mathcal{S}'$ are not ordered, resulting in possible permutations of the sequences.

Hereby $\boldsymbol{x}_3$ was received correctly as $\boldsymbol{y}_1$, $\boldsymbol{x}_1$ was lost, $\boldsymbol{x}_2$ was received in error as $\boldsymbol{y}_2$ and $\boldsymbol{x}_4$ was received in error as $\boldsymbol{y}_3$. It follows that the set of correct, lost and erroneous sequences is given by

$$\begin{aligned}\mathcal{S}_{\mathsf{C}} &= \{\boldsymbol{x}_3\} = \{(\mathsf{GGCATAGCT})\},\\ \mathcal{S}_{\mathsf{L}} &= \{\boldsymbol{x}_1\} = \{(\mathsf{TGAACTACG})\},\\ \mathcal{S}_{\mathsf{E}} &= \{\boldsymbol{x}_2, \boldsymbol{x}_4\} = \{(\mathsf{ATTGCTGAA}), (\mathsf{ACGGGAATC})\}.\end{aligned}$$

It follows that $s = |\mathcal{S}_{\mathsf{L}}| = 1$ and $t = |\mathcal{S}_{\mathsf{E}}| = 2$, where there were $u = 2$ substitution errors in $\boldsymbol{x}_2$ and $\boldsymbol{x}_4$. Therefore, $\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, 1, 2, 2)$.

The combinatorial channel is thus the entity that, given the input $\mathcal{S} \in \mathcal{X}_M^L$ outputs a random set $\mathcal{S}' \in B^{\mathbb{T}}(\mathcal{S}, s, t, u)$. It is visualized in Figure 3.2. Throughout, we refer to the following definition of an error-correcting code in DNA storage systems.

Definition 3.5. *A code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is called an $(s,t,u)_{\mathbb{T}}$-**correcting code**, if it can correct a loss of s (or fewer) sequences and u (or fewer) errors of type $\mathbb{T}$ in each of t (or fewer) sequences, i.e., for any pair $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ with $\mathcal{S}_1 \neq \mathcal{S}_2$, it holds that*

$$B^{\mathbb{T}}(\mathcal{S}_1, s, t, u) \cap B^{\mathbb{T}}(\mathcal{S}_2, s, t, u) = \emptyset.$$

We say $\mathcal{C} \subseteq \mathcal{X}_M^L$ is an $(s,t,\bullet)_{\mathbb{T}}$-correcting code if the number of errors u per erroneous sequences can be arbitrarily large.

Note that by this definition, a *code* is a set of *codewords*, where each *codeword* is again a set of M *sequences*, each of length L. Further, by definition, this code is a *zero-error* code in the following sense. If any codeword $\mathcal{S} \in \mathcal{C}$ of an $(s,t,u)_{\mathbb{T}}$-correcting code is transmitted over the combinatorial channel, it is possible to uniquely recover $\mathcal{S}$, given any received word $\mathcal{S}' \in B^{\mathbb{T}}(\mathcal{S}, s, t, u)$. The redundancy of a code is defined as follows.

Definition 3.6. *The* redundancy *of a code* $\mathcal{C} \subseteq \mathcal{X}_M^L$ *is*

$$r(\mathcal{C}) = \log |\mathcal{X}_M^L| - \log |\mathcal{C}| = \log \binom{q^L}{M} - \log |\mathcal{C}|.$$

We present the results in this work for binary sequences ($q = 2$), however most or all of them can be extended to the non-binary case (and, in particular, the quaternary case).

3.1.1 Discussion of the Channel Model

Designing and analyzing codes over sets allows to efficiently combat several important aspects of DNA-based data storage. These include the loss of the ordering information of the sequences and the loss or erroneous reception of some of the stored sequences as described in our channel model. Especially when not all sequences are received with errors (i.e. some sequences are received correctly), it is not obvious at all, whether, e.g., prepending an index to each sequence is optimal and how the stored sequences should be protected from errors. This is due to the following considerations. Assume, for the sake of the argument, that the DNA strands are encoded in a concatenated manner, which is in fact the standard procedure used in most experiments to date. That is, the digital data to be stored in the archive is encoded using an outer, e.g., Reed-Solomon, code and afterwards sliced into short fragments, such that each fragment corresponds to a DNA strand. Every such fragment, respectively strand is then protected individually with an error-correcting code. We will discuss and modify such constructions in Sections 3.5.1 and 3.5.2. On the one hand this allows for a simpler code design, as the inner code must be designed to only protect a single sequence. However, in such a coding scheme, those sequences which have been received error-free would not have had to be protected against errors. On the other hand, if a weaker inner code were to be used, the outer code has to be very strong in order to cope with many wrong fragments. In fact, the main challenge in this paradigm is that the sequences that will suffer from errors are not known a priori, and the distribution of errors over the sequences after reconstruction is considerably non-uniform due to the fact the number of times a strand is sequenced can vary heavily. This makes coding over sets, an approach to code over all sequences inside the archive jointly, necessary for efficient error correction. Therefore, discussing the channel model from stored sets to received sets is of relevance when aiming for efficient and error-free data storage in DNA. Such a discussion is not possible when only the channel from a single stored sequence to a single received sequence is analyzed. The following remarks summarize two further observations about the channel model.

Remark 3.7. *While in practical DNA-based storage systems, the length of the sequences L is moderate, e.g., in the order of a few hundreds, M is significantly larger. In general, we say that $M = q^{\beta L}$ for some $0 < \beta < 1$. Typical values for the parameters M, L and β can be found in Table 2.1 on Page 17.*

Remark 3.8. *In view of the underlying DNA storage system, which is visualized in Figure 3.1, the parameters s, t, u of the channel model depend on the number of sequences that are drawn from the storage medium and also the reconstruction algorithm. Using an efficient reconstruction algorithm, it can be assumed that s, t, u decrease as the number of draws increases, since the reconstruction can be performed more accurately. In particular, when many more than M sequences are drawn from the storage medium, it can be assumed that there are enough draws per sequence that the*

sequencing errors are corrected by the reconstruction algorithm. Consequently there only remain errors which have been introduced when synthesizing the sequences. However, this dependence of the channel parameters on the clustering and reconstruction algorithm is quite cumbersome to analyze and we therefore directly define the channel based on these parameters.

3.1.2 Relationship of Insertion- and Deletion-Correcting Codes

In this section, we investigate the relationship between $(s,t,u)_{\mathbb{I}}$-insertion-correcting and $(s,t,u)_{\mathbb{D}}$-deletion-correcting codes. It is known [Lev66] that for the case of standard blockcodes, any code can correct u insertions if and only if it can correct any u insertions *and* deletions. Interestingly such an equivalence does *not* hold for our channel model. Here we show a counterexample that an $(s,t,u)_{\mathbb{D}}$-correcting code is not necessarily an $(s,t,u)_{\mathbb{I}}$-correcting code.

Example 3.9. *Consider the code* $\mathcal{C} = \{\mathcal{S}_1, \mathcal{S}_2\}$*, with*

$$\mathcal{S}_1 = \{(\mathsf{AACCA}), (\mathsf{AACAA}), (\mathsf{GGTTG})\},$$
$$\mathcal{S}_2 = \{(\mathsf{ACCAA}), (\mathsf{GGTGG}), (\mathsf{GTTGG})\}.$$

We can verify that $\mathcal{C}$ *is* $(0,3,1)_{\mathbb{D}}$*-correcting. It is however not* $(0,3,1)_{\mathbb{I}}$*-correcting, since the word* $\{(\mathsf{AACCAA}), (\mathsf{GGTTGG})\} \in B^{\mathbb{I}}(\mathcal{S}_1, 0, 3, 1)$ *by editing both the sequences* (AACCA) *and* (AACAA) *to become* (AACCAA) *and* (GGTTG) *to become* (GGTTGG)*. Similarly, the same word can be obtained from* $\mathcal{S}_2$*, i.e.,* $\{(\mathsf{AACCAA}), (\mathsf{GGTTGG})\} \in B^{\mathbb{I}}(\mathcal{S}_2, 0, 3, 1)$*, since we can edit* (ACCAA) *to become* (AACCAA) *and both* (GGTGG) *and* (GTTGG) *to become* (GGTTGG)*.*

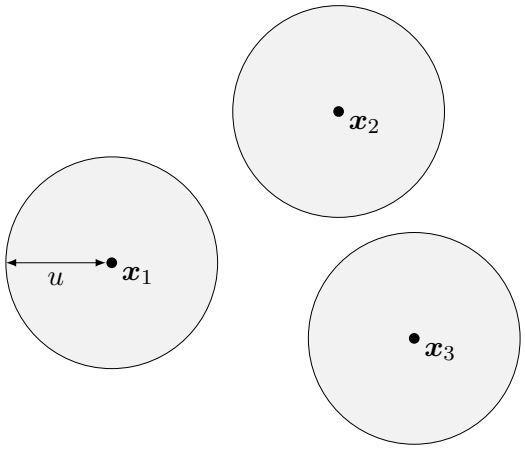
The main reason for this non-equivalence is due to the following. While for two sequences $\boldsymbol{x}_1, \boldsymbol{x}_2$ it holds that $B^{\mathbb{I}}(\boldsymbol{x}_1, u) \cap B^{\mathbb{I}}(\boldsymbol{x}_2, u) = \emptyset$ if and only if $B^{\mathbb{D}}(\boldsymbol{x}_1, u) \cap B^{\mathbb{D}}(\boldsymbol{x}_2, u) = \emptyset$ [Lev66], a generalization of such a statement to three or more sequences does not hold anymore. An example of this fact are the sequences chosen in the above example, $\boldsymbol{x}_1 = (\mathsf{GGTTG})$, $\boldsymbol{x}_2 = (\mathsf{GGTGG})$, $\boldsymbol{x}_3 = (\mathsf{GTTGG})$. They all share a common supersequence (GGTTGG) of length 6, i.e., $(\mathsf{GGTTGG}) \in B^{\mathbb{I}}(\boldsymbol{x}_i, 1)$ for all $i \in \{1,2,3\}$. However, it can be verified that they do not have a common subsequence of length 4. Analogously, based on the same fact, it is possible to find a counterexample for the other direction, i.e., an $(s,t,u)_{\mathbb{I}}$-correcting code is not necessarily an $(s,t,u)_{\mathbb{D}}$-correcting code.

3.2 Overview over Technical Contributions and Methods

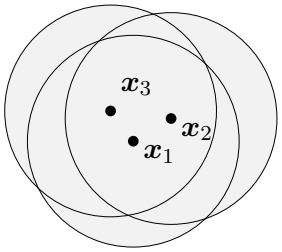
General techniques of bounding the size of optimal zero-error codes are well-established. On the one hand, the arguments used by Gilbert and Varshamov to prove existential results, i.e., upper bounds on the redundancy of optimal codes, have been generalized to a broad class of channels [GF93; Tol97]. In fact, the only ingredient required to directly establish lower bounds in that fashion is the following. Given some channel input, we require an estimate - or more precisely, an upper bound - on the average number of other channel inputs that are potentially confusable with the original input at the receiver. The main contribution in this chapter is thus mostly of combinatorial nature together with adequate limiting techniques for the case of large parameters. For the textbook example of u-substitution-correcting codes, this quantity is precisely the number of words that have Hamming distance at most $2u$ from the input, see, e.g., [Lin99, Ch. 5.1], [Bos14, Ch. 6.3]. However, for the channel under discussion, this derivation is more involved due to the fact that first, the number of confusable words depends on the channel input, and

second, by Definition 3.3, different channel realizations might lead to the same channel output. The general procedure to obtain a valid bound in most cases will be to first bound the number of possible channel outputs for a given channel input and then, for each such output, bound the number of possible inputs that might have produced this output. By a union bound argument, this also bounds the total number of confusable words. We will refine this procedure in some cases to obtain tighter bounds.

On the other hand, a well-known technique, originating from the seminal work of Hamming [Ham50], to conversely derive lower bounds on the redundancy of optimal codes is based on sphere-packing arguments. The generic way to obtain such bounds is to analyze the set of possible channel outputs obtained from a given channel input. Based on the zero-error property of the code, those sets, also called spheres, need to be distinct. Then, dividing the size of the output space by the size of these sets one obtains a valid upper bound on the size of a zero-error code. While this procedure is generally applicable, there are cases that require more elaborate arguments. If the sphere sizes depend on their centers, as is the case for our channel model, it is not obvious how to obtain a valid bound, as the sphere sizes depend on the choice of codewords. In principle, there are three common ways in such scenarios. First, divide by the minimum possible sphere size. Second, generalized sphere packing bounds [FVY15; KK13] can be derived based on a transversal on a hypgergraph associated with the channel. Third, in case that with growing channel parameters, most spheres sizes approach a common size, it is possible to derive asymptotic bounds. We follow the first approach for reasons of simplicity and tractability in this chapter and switch to the third approach in cases where the first approach yields unsatisfactory results. The asymptotic results using the third approach are, intuitively speaking, derived as follows. Split the channel inputs $\mathcal{S} \in \mathcal{X}_M^L$ into two sets $\mathcal{X}_1$ and $\mathcal{X}_2$, such that $\mathcal{X}_1$ contains words with a large number of possible channel outputs and $\mathcal{X}_2$ has small size. Splitting the codebook $\mathcal{C}$ accordingly into words from $\mathcal{X}_1$ and $\mathcal{X}_2$, it is immediate that any zero-error code has size at most $|\mathcal{C}| = |\mathcal{C} \cap \mathcal{X}_1| + |\mathcal{C} \cap \mathcal{X}_2| \leq \frac{|\mathcal{X}_1|}{B_{\min}} + |\mathcal{X}_2|$, where $B_{\min}$ is the minimum number of possible channel outputs over all words in $\mathcal{X}_1$. To obtain a good bound it is hereby desirable to choose $\mathcal{X}_1$ and $\mathcal{X}_2$ such that $B_{\min}$ is large and $|\mathcal{X}_2|$ is small compared to $\frac{|\mathcal{X}_1|}{B_{\min}}$. Recall to this end from Definitions 3.3 and 3.5 that in our channel model, the channel input is a set of sequences, where a subset of sequences can be lost, and another can be distorted by point errors. We will make use of the fact that sets of outspread sequences, i.e., sequences whose individual error spheres intersect in few or no points, have a large number of channel outputs. Conversely, sets with sequences that are clumped together have small number of possible channel outputs, as many different error events yield the same result. Figure 3.3 visualizes such sets in both cases. We will combine these facts with the observation that a very large fraction of sets do have the property that the sequences are well spread and thus their number of possible channel outputs is close to the maximum. The technical challenges hereby are as follows. First, it is necessary to extract an easy-to-analyze property of channel inputs that allows to bound the number of possible channel outputs from below. We will identify this property as the size of the largest subset of input sequences that are well spread. For a formal definition, we refer the reader to Lemmas 3.21 and 3.26. Second, we need a quantitative criterion for the partition of sets into $\mathcal{X}_1$ and $\mathcal{X}_2$. We split the sets based on their number of channel outputs and carefully choose a threshold such that both $B_{\min}$ is large and the second set is very small. Our results about upper and lower bounds on the optimal redundancy are summarized in Table 3.1.



(a) Example of a set $\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ with a large number of possible channel outputs.



(b) Example of a set $\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ with a small number of possible channel outputs.

Figure 3.3: Illustration of sets with many and few channel outputs. Sequences that are close to each other have a small distance in the appropriate metric (e.g. Hamming, Levenshtein) and, similarly, sequences that are far apart have a large distance.

3.3 Existential Gilbert-Varshamov-type Upper Bounds

We start by deriving Gilbert-Varshamov lower bounds on the size of optimal $(s,t,\bullet)_{\mathbb{L}}$, $(s,t,u)_{\mathbb{S}}$, $(s,t,u)_{\mathbb{I}}$ and $(s,t,u)_{\mathbb{D}}$-correcting codes. These bounds imply upper bounds on the optimal redundancy of such codes. The central quantity for the derivation of the Gilbert-Varshamov bounds is, for a given $\mathcal{S} \in \mathcal{X}_M^L$, the set of words $\widetilde{\mathcal{S}} \in \mathcal{X}_M^L$, which have intersecting error balls with $\mathcal{S}$. It is formally defined as follows.

Definition 3.10. *For a set $\mathcal{S} \in \mathcal{X}_M^L$, we denote by $V^{\mathbb{T}}(\mathcal{S}, s, t, u)$ the set of all sets $\widetilde{\mathcal{S}} \in \mathcal{X}_M^L$, which have intersecting error balls with $\mathcal{S}$, that is,*

$$V^{\mathbb{T}}(\mathcal{S}, s, t, u) = \{\widetilde{\mathcal{S}} \in \mathcal{X}_M^L : B^{\mathbb{T}}(\mathcal{S}, s, t, u) \cap B^{\mathbb{T}}(\widetilde{\mathcal{S}}, s, t, u) \neq \emptyset\}.$$

Hereby, $|V^{\mathbb{T}}(\mathcal{S}, s, t, u)|$ is called the degree of $\mathcal{S}$. The average degree of all sets is denoted by

$$\overline{V}^{\mathbb{T}}(s, t, u) = \frac{1}{|\mathcal{X}_M^L|} \sum_{\mathcal{S} \in \mathcal{X}_M^L} |V^{\mathbb{T}}(\mathcal{S}, s, t, u)|.$$

The generalized Gilbert-Varshamov bound (cf. [GF93; Tol97]) is derived using a graph representation of an error-correcting code. We will use this representation to find the generalized Gilbert-Varshamov bound for the DNA storage channel. Consider the simple graph G with the set of vertices $\mathcal{X}_M^L$. Two vertices $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{X}_M^L$ are connected if they cannot be confused after transmission over the DNA storage channel, i.e., if $B^{\mathbb{T}}(\mathcal{S}_1, s, t, u) \cap B^{\mathbb{T}}(\mathcal{S}_2, s, t, u) = \emptyset$, or equivalently, $\mathcal{S}_2 \notin V^{\mathbb{T}}(\mathcal{S}_1, s, t, u)$. Note that this definition is slightly different from [GF93; Tol97] due to the lack of a distance measure in our case. By construction, a *clique* in G (collection of vertices, where each pair of vertices is connected) is an $(s,t,u)_{\mathbb{T}}$-correcting code. Now, it can directly be shown that the total number of edges G coincides with [Tol97, eq. (2)]. Analogously to [Tol97], it is therefore possible to establish a lower bound on the size of a clique in G (and therefore an $(s,t,u)_{\mathbb{T}}$-correcting code).

Table 3.1: Lower and upper bounds on the redundancy of optimal $(s,t,u)_{\mathbb{T}}$-correcting codes. Low order terms are omitted.

Error correction	Gilbert-Varshamov bound		Sphere packing bound	
$(s,t,\bullet)_{\mathbb{L}}$	$(s+2t)L+(s+2t)\log M$	[Thm. 3.12]	$(s+t)L+t\log M$	[Cor. 3.18]
$(\sigma M,\tau M,\bullet)_{\mathbb{L}}$	$(\sigma+2\tau)(L-\log M)$	[Thm. 3.12]	$(\sigma+\tau)M(L-\log M)$	[Cor. 3.18]
$(s,t,u)_{\mathbb{I}}$	$sL+(s+t)\log M+2tu\log L$	[Thm. 3.14]	$sL+tu\log L$	[Thm. 3.20]
$(s,t,u)_{\mathbb{D}}$	$sL+(s+t)\log M+2tu\log(L/2)$	[Thm. 3.16]	$sL+tu\log L$	[Thm. 3.27]
$(s,t,u)_{\mathbb{S}}$	$sL+(s+2t)\log M+2tu\log L$	[Thm. 3.13]	$sL+t\log M+tu\log L$	[Thm. 3.23]
$(s,M-s,u)_{\mathbb{D}}$	$2Mu\log L$	[Thm. 3.16]	$Mu\log L$	[Thm. 3.28]
$(s,M-s,u)_{\mathbb{S}}$	$2Mu\log L$	[Thm. 3.13]	$Mu\log L$	[Thm. 3.24]

Theorem 3.11 (cf. [GF93; Tol97])**.** *There exists an $(s,t,u)_{\mathbb{T}}$-correcting code $\mathcal{C}\subseteq\mathcal{X}_M^L$ of size at least*

$$|\mathcal{C}|\geq\frac{\binom{2^L}{M}}{\overline{V}^{\mathbb{T}}(s,t,u)}.$$

Such a code can be constructed by successively selecting words $\mathcal{S}^{(i)}$ with minimum degree from $\mathcal{X}_M^L$ as codewords and removing all words $V^{\mathbb{T}}(\mathcal{S}^{(i)},s,t,u)$ as possible candidates for the succeeding codewords. Bounding the denominator in Theorem 3.11 from above will be the main challenge throughout this section.

3.3.1 Arbitrary Number of Edit Errors per Sequence

We start with the case of a loss of s sequences and an arbitrary number of edit errors in at most t sequences. The main theorem for this case is proven in the following.

Theorem 3.12. *There exists an $(s,t,\bullet)_{\mathbb{L}}$-correcting code $\mathcal{C}\subseteq\mathcal{X}_M^L$ of cardinality at least*

$$|\mathcal{C}|\geq\frac{\binom{2^L}{M}}{\binom{M}{s+2t}\binom{2^L}{s+2t}}.$$

Hence, for fixed $s,t\in\mathbb{N}_0$ and fixed $0<\beta<1$, there exists an $(s,t,\bullet)_{\mathbb{L}}$-correcting code $\mathcal{C}\subseteq\mathcal{X}_M^L$ with redundancy

$$r(\mathcal{C})\leq(s+2t)L+(s+2t)\log M-\log((s+2t)!^2)+o(1),$$

when $M\to\infty$ with $M=2^{\beta L}$.

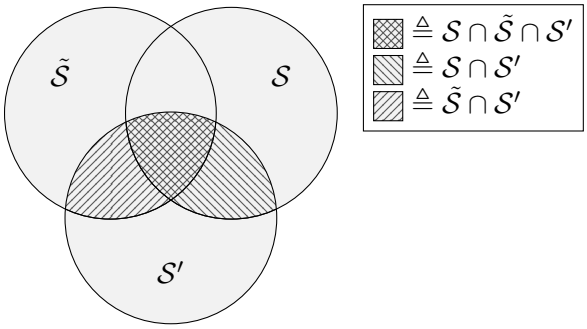


Figure 3.4: Illustration for the proof of Theorem 3.12

Proof. We will find an upper bound on $\overline{V}^{\mathbb{L}}(s,t,\bullet)$ by bounding $|V^{\mathbb{L}}(\mathcal{S},s,t,\bullet)|$ from above for all $\mathcal{S} \in \mathcal{X}_M^L$. In the following, let $\widetilde{\mathcal{S}} \in V^{\mathbb{L}}(\mathcal{S},s,t,\bullet) \subseteq \mathcal{X}_M^L$ be a set which has an intersecting error ball with $\mathcal{S}$. Start by observing that for any such $\widetilde{S}$, there exists $\mathcal{S}' \in B^{\mathbb{L}}(\mathcal{S},s,t,\bullet) \cap B^{\mathbb{L}}(\widetilde{\mathcal{S}},s,t,\bullet)$ with $|\mathcal{S}'| \leq M-s$, since $B^{\mathbb{L}}(\mathcal{S},s,t,\bullet) \cap B^{\mathbb{L}}(\widetilde{\mathcal{S}},s,t,\bullet) \neq \emptyset$ and for all $\mathcal{S}'' \in B^{\mathbb{L}}(\mathcal{S},s,t,\bullet) \cap B^{\mathbb{L}}(\widetilde{\mathcal{S}},s,t,\bullet)$ with $|\mathcal{S}''| > M-s$ it is possible to construct $\mathcal{S}' \in B^{\mathbb{L}}(\mathcal{S},s,t,\bullet) \cap B^{\mathbb{L}}(\widetilde{\mathcal{S}},s,t,\bullet)$ with $|\mathcal{S}'| = M-s$ by removing any $|\mathcal{S}''|-M+s$ sequences from $\mathcal{S}''$. By Definition 3.3, $|\mathcal{S} \cap \mathcal{S}'| \geq M-s-t$ and also $|\widetilde{\mathcal{S}} \cap \mathcal{S}'| \geq M-s-t$. Further, for any such $\mathcal{S}'$,

$$\begin{aligned} |\mathcal{S} \cap \widetilde{\mathcal{S}}| &\geq |\mathcal{S} \cap \widetilde{\mathcal{S}} \cap \mathcal{S}'| \overset{(a)}{\geq} |\mathcal{S} \cap \mathcal{S}'| + |\widetilde{\mathcal{S}} \cap \mathcal{S}'| - |\mathcal{S}'| \\ &\geq 2(M-s-t) - (M-s) = M-s-2t, \end{aligned}$$

where we used in (a) that $|\mathcal{S} \cap \widetilde{\mathcal{S}} \cap \mathcal{S}'| = |\mathcal{S} \cap \mathcal{S}'| + |\widetilde{\mathcal{S}} \cap \mathcal{S}'| - |(\mathcal{S} \cup \widetilde{\mathcal{S}}) \cap \mathcal{S}'| \geq |\mathcal{S} \cap \mathcal{S}'| + |\widetilde{\mathcal{S}} \cap \mathcal{S}'| - |\mathcal{S}'|$ (for an illustration, refer to Figure 3.4). Therefore, any $\widetilde{\mathcal{S}}$ has an intersection of size at least $M-s-2t$ with $\mathcal{S}$. Note that for $2^L \geq M+s+2t$ this bound is tight, i.e., it is possible to find sets $\mathcal{S}, \widetilde{\mathcal{S}} \in \mathcal{X}_M^L$ with $B^{\mathbb{L}}(\mathcal{S},s,t,\bullet) \cap B^{\mathbb{L}}(\widetilde{\mathcal{S}},s,t,\bullet) \neq \emptyset$ and $|\mathcal{S} \cap \widetilde{\mathcal{S}}| = M-s-2t$. Each $\widetilde{\mathcal{S}}$ can thus be constructed by removing $s+2t$ sequences from $\mathcal{S}$ and adding $s+2t$ arbitrary sequences. The total number of elements $\widetilde{\mathcal{S}}$ is thus at most $|V^{\mathbb{L}}(\mathcal{S},s,t,\bullet)| \leq \binom{M}{s+2t}\binom{2^L}{s+2t}$. It follows by Theorem 3.11 that there exists an $(s,t,\bullet)_{\mathbb{L}}$-correcting code $\mathcal{C}$ of size at least

$$|\mathcal{C}| \geq \frac{\binom{2^L}{M}}{\binom{M}{s+2t}\binom{2^L}{s+t}}.$$

By Definition 3.6, there exists an $(s,t,\bullet)_{\mathbb{L}}$-correcting code $\mathcal{C}$ with redundancy at most

$$\begin{aligned} r(\mathcal{C}) &= \log\binom{2^L}{M} - \log|\mathcal{C}| \leq \log\binom{M}{s+2t}\binom{2^L}{s+2t} \\ &\overset{(a)}{=} (s+2t)L + (s+2t)\log M - \log((s+2t)!^2) + o(1), \end{aligned}$$

where in equation (a) we used Lemma A.2 from Appendix A.1.

□

3.3.2 Substitution Errors

We will now establish the existence of a code for the case of a loss of s sequences and a fixed number of u substitution errors in t sequences. As before, we will use Theorem 3.11, however,

bounding $|V^{\mathbb{S}}(\mathcal{S}, s, t, u)|$ is slightly more involved in this case. In principle our arguments will be as follows. First, we bound the number of sets $\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, s, t, u)$. Then, we will bound the number $|\{\widetilde{\mathcal{S}} \in \mathcal{X}_M^L : \mathcal{S}' \in B^{\mathbb{S}}(\widetilde{\mathcal{S}}, s, t, u)\}|$ of sets that contain $\mathcal{S}'$ in their error ball for all such $\mathcal{S}'$. Multiplying these two quantities, one obtains an upper bound on the number of sets $|V^{\mathbb{S}}(\mathcal{S}, s, t, u)|$. This procedure is formalized in the following theorem.

Theorem 3.13. *There exists an $(s, t, u)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ with cardinality at least*

$$|\mathcal{C}| \geq \frac{\binom{2^L}{M}}{\binom{M+s}{s}\binom{M}{t}\binom{M+t-1}{t}\binom{2^L}{s}(B^{\mathbb{S}}(L,u))^{2t}}.$$

Hence, for fixed $s, t, u \in \mathbb{N}_0$ and fixed $0 < \beta < 1$, there exists an $(s, t, u)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ with redundancy

$$r(\mathcal{C}) \leq sL + (s + 2t)\log M + 2tu \log L - \log(s!^2 t!^2 u!^{2t}) + o(1),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. We will find an upper bound on $|V^{\mathbb{S}}(\mathcal{S}, s, t, u)|$ for all $\mathcal{S} \in \mathcal{X}_M^L$. We can directly bound

$$|B^{\mathbb{S}}(\mathcal{S}, s, t, u)| \overset{(a)}{\leq} \sum_{i=0}^{s} \binom{M}{i}\binom{M-i}{t}(B^{\mathbb{S}}(L,u))^t \overset{(b)}{\leq} \binom{M+s}{s}\binom{M}{t}(B^{\mathbb{S}}(L,u))^t,$$

where (a) holds as we can choose at most s out of M sequences to be lost, t out of the remaining sequences to be erroneous and there are $B^{\mathbb{S}}(L, u)$ error patterns for each erroneous sequence. Inequality (b) follows from the fact that $\binom{M-i}{t} \leq \binom{M}{t}$ for all $i \geq 0$ and the bound $\sum_{i=0}^{s}\binom{M}{i} \leq \binom{M+s}{s}$. Given $\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, s, t, u)$, we count the number of possible $\widetilde{\mathcal{S}}$ with $\mathcal{S}' \in B^{\mathbb{S}}(\widetilde{\mathcal{S}}, s, t, u)$ as follows. For each of the t erroneous sequences it is possible to either add u errors to a sequence $\boldsymbol{x} \in \mathcal{S}'$ or to create a new sequence inside the error ball $B^{\mathbb{S}}(\boldsymbol{x}, u)$. There are $\binom{M+t-1}{t}(B^{\mathbb{S}}(L,u))^t$ possible error patterns for this procedure. Finally, the s lost sequences can be arbitrary sequences $\boldsymbol{x} \in \Sigma_2^L$, and there are at most $\binom{2^L}{s}$ choices for these sequences. Thus,

$$|V^{\mathbb{S}}(\mathcal{S}, s, t, u)| \leq \binom{M+s}{s}\binom{M}{t}(B^{\mathbb{S}}(L,u))^t\binom{M+t-1}{t}(B^{\mathbb{S}}(L,u))^t\binom{2^L}{s},$$

for all $\mathcal{S} \in \mathcal{X}_M^L$. Therefore, also the average degree $\overline{V}^{\mathbb{S}}(s, t, u)$ is bounded by the same quantity and we can apply Theorem 3.11. By Definition 3.6, there exists an $(s, t, u)_{\mathbb{S}}$-correcting code $\mathcal{C}$ with redundancy at most

$$\begin{aligned} r(\mathcal{C}) &= \log\binom{2^L}{M} - \log|\mathcal{C}| \leq \log\binom{M+s}{s}\binom{M}{t}\binom{M+t-1}{t}(B^{\mathbb{S}}(L,u))^{2t}\binom{2^L}{s} \\ &\overset{(a)}{=} sL + (s+2t)\log M + 2tu\log L - \log(s!^2 t!^2 u!^{2t}) + o(1), \end{aligned}$$

where the equality (a) follows from Lemma A.2 in the Appendix A.1. □

3.3.3 Insertion Errors

We now turn to the case of insertions and deletions. We can use similar arguments as in the case of substitution errors to obtain the following result for insertion errors.

Theorem 3.14. *There exists an $(s,t,u)_{\mathbb{I}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ with cardinality at least*

$$|\mathcal{C}| \geq \frac{\binom{2^L}{M}}{\binom{M}{s}\binom{M-s}{t}\binom{2^L}{s}(S^{\mathbb{I}}(L,u))^t\binom{L}{u}^t}.$$

Hence, for fixed $s,t,u \in \mathbb{N}_0$ and fixed $0<\beta<1$, there exists an $(s,t,u)_{\mathbb{I}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ with redundancy

$$r(\mathcal{C}) \leq sL + (s+t)\log M + 2tu\log L - \log(s!^2 t! u!^{2t}) + o(1),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. We will find an upper bound on $|V^{\mathbb{I}}(\mathcal{S},s,t,u)|$ for all $\mathcal{S} \in \mathcal{X}_M^L$. Let $\mathcal{S}' \in B^{\mathbb{I}}(\mathcal{S},s,t,u)$ be such that *exactly* s sequences were lost and there were *exactly* u insertions in each of t sequences. The number of such elements $\mathcal{S}'$ is at most $\binom{M}{s}\binom{M-s}{t}(S^{\mathbb{I}}(L,u))^t$, as we can choose s out of M sequences to be lost, t out of the remaining $M-s$ sequences to be erroneous and there are $S^{\mathbb{I}}(L,u)$ error patterns for each erroneous sequence. Given $\mathcal{S}' \in B^{\mathbb{I}}(\mathcal{S},s,t,u)$, we count possible $\widetilde{\mathcal{S}}$ with $\mathcal{S}' \in B^{\mathbb{I}}(\widetilde{\mathcal{S}},s,t,u)$ as follows. From each of the t erroneous sequences we can delete u symbols. There are $\binom{L}{u}^t$ possible deletion patterns. Then, the s lost sequences can be arbitrary sequences $\boldsymbol{x} \in \Sigma_2^L$, and there are at most $\binom{2^L}{s}$ choices for these sequences. Thus,

$$|V^{\mathbb{I}}(\mathcal{S},s,t,u)| \leq \binom{M}{s}\binom{M-s}{t}(S^{\mathbb{I}}(L,u))^t\binom{2^L}{s}\binom{L}{u}^t$$

for all $\mathcal{S} \in \mathcal{X}_M^L$.[3] Therefore, also the average $\overline{V}^{\mathbb{I}}(s,t,u)$ is bounded by the same quantity and we can apply Theorem 3.11 to prove existence of an $(s,t,u)_{\mathbb{I}}$-correcting code $\mathcal{C}$ with size as given in the theorem statement. The result on the redundancy follows from Definition 3.6 and a repeated application of Lemma A.2 from Appendix A.1. □

3.3.4 Deletion Errors

For the case of deletion errors, we slightly adapt our arguments since the size of the deletion sphere is non-uniform [Lev66]. As stated in Theorem 3.11, it is sufficient to find an upper bound on the average degree $\overline{V}^{\mathbb{D}}(s,t,u)$. We will therefore show how this can be used to derive a bound that depends only on the average deletion sphere size, defined as follows.

Definition 3.15. *The average of the t-th power of the deletion sphere size $|S^{\mathbb{D}}(\boldsymbol{x},u)|$ is defined to be*

$$\overline{S}^{\mathbb{D},t}(u) = \frac{1}{2^L}\sum_{\boldsymbol{x}\in\Sigma_2^L}|S^{\mathbb{D}}(\boldsymbol{x},u)|^t.$$

[3] We note that here we chose only those sets $\mathcal{S}'$ that have been obtained by a loss of *exactly* s sequences and by *exactly* u errors in each of t sequences. This choice can be justified by the fact that it can be shown that for any $\mathcal{S},\widetilde{\mathcal{S}} \in \mathcal{X}_M^L$ with $B^{\mathbb{I}}(\mathcal{S},s,t,u) \cap B^{\mathbb{I}}(\widetilde{\mathcal{S}},s,t,u) \neq \emptyset$ there exists a word $\mathcal{S}' \in B^{\mathbb{I}}(\mathcal{S},s,t,u) \cap B^{\mathbb{I}}(\widetilde{\mathcal{S}},s,t,u)$ that is obtained by a loss of *exactly* s sequences and by *exactly* u errors in each of t sequences. Therefore, we count all sets $\widetilde{\mathcal{S}} \in V^{\mathbb{I}}(\mathcal{S},s,t,u)$ using the arguments in this proof.

Based on this definition we can formulate the following theorem about the existence of $(s,t,u)_{\mathbb{D}}$-correcting codes.

Theorem 3.16. *There exists an $(s,t,u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ with cardinality at least*

$$|\mathcal{C}| \geq \frac{\binom{2^L}{M}}{\binom{M}{s}\binom{M-s}{t}\binom{2^L}{s}(S^{\mathbb{I}}(L-u,u))^t \overline{S}^{\mathbb{D},t}(u)}.$$

Hence, for fixed $s,t,u \in \mathbb{N}_0$ and fixed $0<\beta<1$, there exists an $(s,t,u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ with redundancy

$$r(\mathcal{C}) \leq sL + (s+t)\log M + 2tu\log L - tu - \log(s!^2 t! u!^{2t}) + o(1),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. We derive an upper bound on the average degree $\overline{V}^{\mathbb{D}}(s,t,u)$. Recall that $S^{\mathbb{D}}(\mathcal{S},s,t,u)$ is the set of all words $\mathcal{S}' \in B^{\mathbb{D}}(\mathcal{S},s,t,u)$ obtained from $\mathcal{S}$ by a loss of *exactly* s sequences and *exactly* u deletions in t sequences[4]. The number of such words is at most

$$|S^{\mathbb{D}}(\mathcal{S},s,t,u)| \leq \sum_{\mathcal{S}_{\mathsf{E}} \subseteq \mathcal{S}, |\mathcal{S}_{\mathsf{E}}|=t} \prod_{\boldsymbol{x} \in \mathcal{S}_{\mathsf{E}}} |S^{\mathbb{D}}(\boldsymbol{x},u)| \binom{M-t}{s}.$$

This can be illustrated by the following consideration. First, fix $\mathcal{S}_{\mathsf{E}} \subseteq \mathcal{S}$ with $|\mathcal{S}_{\mathsf{E}}| = t$. There are $|S^{\mathbb{D}}(\boldsymbol{x},u)|$ possible error patterns for each $\boldsymbol{x} \in \mathcal{S}_{\mathsf{E}}$ and $\binom{M-t}{s}$ choices of s lost sequences among the remaining $M-t$ error-free sequences. Summing over all possible choices $\mathcal{S}_{\mathsf{E}} \subseteq \mathcal{S}$ of erroneous sequences yields the bound. Then, for each such set $\mathcal{S}'$, there are at most $\binom{2^L}{s}(S^{\mathbb{I}}(L-u,u))^t$ sets $\widetilde{\mathcal{S}}$ with $\mathcal{S}' \in B^{\mathbb{D}}(\widetilde{\mathcal{S}},s,t,u)$. This is because each erroneous sequence $\boldsymbol{x}' \in \mathcal{S}'$ has length $L-u$ and requires exactly u insertions to become a sequence of length L and the s lost sequences can be arbitrary words in $\widetilde{\mathcal{S}}$. Therefore,

$$|V^{\mathbb{D}}(\mathcal{S},s,t,u)| \leq |S^{\mathbb{D}}(\mathcal{S},s,t,u)| \binom{M-t}{s}\binom{2^L}{s}(S^{\mathbb{I}}(L-u,u))^t.$$

Taking the average of $|S^{\mathbb{D}}(\mathcal{S},s,t,u)|$ over all sets $\mathcal{S} \in \mathcal{X}_M^L$ yields

$$\begin{aligned}\sum_{\mathcal{S}\in\mathcal{X}_M^L} \frac{|S^{\mathbb{D}}(\mathcal{S},s,t,u)|}{\binom{2^L}{M}} &\leq \frac{\binom{M-t}{s}}{\binom{2^L}{M}} \sum_{\mathcal{S}\in\mathcal{X}_M^L} \sum_{\mathcal{S}_{\mathsf{E}}\subseteq\mathcal{S},|\mathcal{S}_{\mathsf{E}}|=t} \prod_{\boldsymbol{x}\in\mathcal{S}_{\mathsf{E}}} |S^{\mathbb{D}}(\boldsymbol{x},u)| \overset{(a)}{=} \frac{\binom{M}{t}\binom{M-t}{s}}{\binom{2^L}{t}} \sum_{|\mathcal{S}_{\mathsf{E}}|=t} \prod_{\boldsymbol{x}\in\mathcal{S}_{\mathsf{E}}} |S^{\mathbb{D}}(\boldsymbol{x},u)| \\ &\overset{(b)}{\leq} \frac{\binom{M}{s}\binom{M-s}{t}}{\binom{2^L}{t}} \sum_{|\mathcal{S}_{\mathsf{E}}|=t} \sum_{\boldsymbol{x}\in\mathcal{S}_{\mathsf{E}}} \frac{|S^{\mathbb{D}}(\boldsymbol{x},u)|^t}{t} \overset{(c)}{=} \binom{M}{s}\binom{M-s}{t}\overline{S}^{\mathbb{D},t}(u),\end{aligned}$$

where the sum over $|\mathcal{S}_{\mathsf{E}}| = t$ runs over all sets $\mathcal{S}_{\mathsf{E}} \subseteq \Sigma_2^L$ with $|\mathcal{S}_{\mathsf{E}}| = t$. Here, for the equality (a) we used that each set $\mathcal{S}_{\mathsf{E}}$ with $|\mathcal{S}_{\mathsf{E}}| = t$ is contained in exactly $\binom{2^L-t}{M-t}$ sets $\mathcal{S} \in \mathcal{X}_M^L$ together with the equality $\binom{2^L-t}{M-t}\binom{2^L}{t} = \binom{M}{t}\binom{2^L}{M}$. Further, in inequality (b), a combination of the arithmetic-geometric mean inequality and Jensen inequality [Jen06], [CT06, Thm. 2.6.2] has been used

[4] Regarding the restriction to sets $\mathcal{S}'$ obtained by a loss of *exactly* s sequences and by *exactly* u errors in each of t sequences, we are using the analog argument as in the proof of Theorem 3.14, adapted to the case of deletions.

to show that for any non-negative $a_1, \ldots, a_t \geq 0$ it holds that $a_1 \cdot \ldots \cdot a_t \leq \frac{1}{t}(a_1^t + \ldots + a_t^t)$. Equality (c) follows from the fact that each $\boldsymbol{x} \in \Sigma_2^L$ is contained in $\binom{2^L-1}{t-1}$ sets $\mathcal{S}_\mathsf{E} \in \mathcal{X}_t^L$. Finally, to bound $\overline{S}^{\mathbb{D},t}(u)$ from above, we use the well-known result $|S^{\mathbb{D}}(\boldsymbol{x}, u)| \leq \binom{||\boldsymbol{x}||+u-1}{u} \leq \frac{(||\boldsymbol{x}||+u-1)^u}{u!}$ from Levenshtein [Lev66], which results in

$$\overline{S}^{\mathbb{D},t}(u) \leq \frac{1}{2^L} \sum_{\boldsymbol{x} \in \Sigma_2^L} \frac{(||\boldsymbol{x}|| + u - 1)^{tu}}{u!^t} \overset{(a)}{=} \frac{1}{u!^t} \sum_{i=0}^{L-1} \frac{\binom{L-1}{i}(i+u)^{tu}}{2^{L-1}} \overset{(b)}{=} \frac{1}{u!^t} \left(\frac{L}{2}\right)^{ut} (1 + o(1)).$$

In equation (a), we used that the number of words $\boldsymbol{x} \in \Sigma_2^L$ with $||\boldsymbol{x}|| = i$ is $2\binom{L-1}{i-1}$. For the asymptotic approximation (b), we identify the sum as the $(tu)-$th decentralized moment of a binomial distribution with $L-1$ trials and success probability $\frac{1}{2}$. Combining [Kno08, eq. (4.1)] and [Kno08, eq. (4.6)], one obtains the asymptotic behavior for fixed u, and t, when $L \to \infty$. □

3.4 Sphere-Packing Lower Bounds

A well-known method to find upper bounds on the cardinality of error-correcting codes is the sphere-packing bound. In this section we derive sphere-packing bounds for $(s, t, u)_\mathbb{T}$-correcting codes. These bounds directly imply lower bounds on the redundancy of such codes. One particular observation of the considered DNA storage channel is that it is non-uniform, i.e. the sizes of the error balls $B^\mathbb{T}(\mathcal{S}, s, t, u)$ depend on the channel input $\mathcal{S}$ for all types of errors $\mathbb{T}$, which hinders the computation of sphere packing bounds. A practical method to find sphere packing bounds for non-uniform error balls is the generalized sphere packing bound [FVY15; KK13]. However, due to the complex expressions of the error ball sizes, this method does not yield tractable expressions for the considered channel. Another possibility is to derive the sphere packing bound by finding an upper bound on the error ball size, which we will do in Section 3.4.1. We will also show that for large M most of the error balls have a similar size, which allows to formulate tighter asymptotic sphere packing bounds in Sections 3.4.3 and 3.4.4. Note that together with the lower bounds on the achievable size of $(s, t, u)_\mathbb{T}$-correcting codes from the previous section and concrete code constructions in Section 3.5, it can be shown that the sphere packing bounds are asymptotically tight for many channel parameters. Even when the bounds are not tight, they can provide important insights into the nature of the DNA channel as well as allow to evaluate coding schemes.

3.4.1 Arbitrary Number of Edit Errors per Sequence

We start by finding an upper bound for $(s, t, \bullet)_\mathbb{L}$-correcting codes, which depicts the case of a loss of s sequences and an arbitrary number of insertion, deletion, and substitution errors in each of t erroneous sequences.

Theorem 3.17. *The cardinality of any $(s, t, \bullet)_\mathbb{L}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies*

$$|\mathcal{C}| \leq \frac{\binom{2^L}{M-s}}{\binom{M}{t+s}\binom{2^L-M}{t}}.$$

In particular, the redundancy of any $(s, t, \bullet)_\mathbb{L}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is therefore at least

$$r(\mathcal{C}) \geq (s+t)\log(2^L - M - t) + t\log(M - s - t) - \log(t!(s+t)!).$$

Proof. We prove the theorem by finding a subset of $B^{\mathbb{L}}(\mathcal{S}, s, t, \bullet)$, which gives a lower bound on the sphere size $|B^{\mathbb{L}}(\mathcal{S}, s, t, \bullet)|$ for all $\mathcal{S} \in \mathcal{X}_M^L$. Let $\mathcal{S}' \in B^{\mathbb{L}}(\mathcal{S}, s, t, \bullet)$ with $\mathcal{S}' \subseteq \Sigma_2^L$ denote an element from the error ball of $\mathcal{S}$, which contains only sequences of length L and let $\mathcal{S}_{\mathsf{C}}, \mathcal{S}'_{\mathsf{E}}$ denote the corresponding error-free, respectively erroneous outcomes of the sequences, i.e. $\mathcal{S}' = \mathcal{S}_{\mathsf{C}} \cup \mathcal{S}'_{\mathsf{E}}$, according to Definition 3.3. We construct such distinct $\mathcal{S}'$ in the following way. Choose $M - s - t$ error-free sequences $\mathcal{S}_{\mathsf{C}} \subseteq \mathcal{S}$ and choose the t erroneous sequences in $\mathcal{S}'_{\mathsf{E}}$ to be distinct elements out of the $2^L - M$ sequences in $\Sigma_2^L \setminus \mathcal{S}$ and let $\mathcal{S}' = \mathcal{S}_{\mathsf{C}} \cup \mathcal{S}'_{\mathsf{E}}$. For any such $\mathcal{S}_{\mathsf{C}} \subseteq \mathcal{S}$ and $\mathcal{S}'_{\mathsf{E}} \subseteq \Sigma_2^L \setminus \mathcal{S}$ one obtains a unique element from the error ball $B^{\mathbb{L}}(\mathcal{S}, s, t, \bullet)$, since $\mathcal{S}' = \mathcal{S}_{\mathsf{C}} \cup \mathcal{S}'_{\mathsf{E}}$ and $\mathcal{S}_{\mathsf{C}}, \mathcal{S}'_{\mathsf{E}}$ are both subsets of two distinct sets. There are in total $\binom{M}{s+t}$ ways to choose the set $\mathcal{S}_{\mathsf{C}}$ and $\binom{2^L-M}{t}$ ways to choose $\mathcal{S}'_{\mathsf{E}}$ and thus $|B^{\mathbb{L}}(\mathcal{S}, s, t, \bullet)| \geq \binom{M}{s+t}\binom{2^L-M}{t}$. All such constructed received sets have $|\mathcal{S}'| = |\mathcal{S}_{\mathsf{C}}| + |\mathcal{S}'_{\mathsf{E}}| = M - s$ sequences of length L. By Definition 3.5 of an $(s, t, \bullet)_{\mathbb{L}}$-correcting code $\mathcal{C}$, for any two $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ with $\mathcal{S}_1 \neq \mathcal{S}_2$, we need to have $B^{\mathbb{L}}(\mathcal{S}_1, s, t, \bullet) \cap B^{\mathbb{L}}(\mathcal{S}_2, s, t, \bullet) = \emptyset$. We thus obtain by a sphere packing argument, that any $(s, t, \bullet)_{\mathbb{L}}$-correcting code $\mathcal{C}$ satisfies

$$|\mathcal{C}| \leq \frac{\binom{2^L}{M-s}}{\binom{M}{t+s}\binom{2^L-M}{t}}.$$

Therefore, the redundancy is at least

$$\begin{aligned} r(\mathcal{C}) &= \log\binom{2^L}{M} - \log|\mathcal{C}| \geq \log\frac{(2^L - M + s)!(M - s)!}{(2^L - M - t)!(M - s - t)!(s + t)!t!} \\ &\overset{(a)}{\geq} (s + t)\log(2^L - M - t) + t\log(M - t - s) - \log(t!(s + t)!). \end{aligned}$$

where in (a) we used that for any $a, b \in \mathbb{N}$ with $a \leq b$, it holds that $\frac{a!}{b!} = \frac{1}{b(b-1)\ldots(a+1)} \geq \frac{1}{b^{b-a}}$. □

This non-asymptotic bound directly implies an asymptotic bound, when $M \to \infty$ and $M = 2^{\beta L}$ for fixed $0 < \beta < 1$.

Corollary 3.18. *For fixed $s, t \in \mathbb{N}_0$ and fixed $0 < \beta < 1$, the redundancy of any $(s, t, \bullet)_{\mathbb{L}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is asymptotically at least*

$$r(\mathcal{C}) \geq (s + t)L + t\log M - \log(t!(s + t)!) + o(1),$$

when $M \to \infty$ and $M = 2^{\beta L}$. Further, for any fixed σ, τ with $\sigma > 0$, $\tau > 0$ and $\sigma + \tau < 1$, the redundancy of any $(\sigma M, \tau M, \bullet)_{\mathbb{L}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies

$$r(\mathcal{C}) \geq (\sigma + \tau)M(L - \log M) + O(M).$$

Proof. The first statement directly follows from Theorem 3.17,

$$\begin{aligned} r(\mathcal{C}) &\geq (s + t)\log(2^L - M - t) + t\log(M - t - s) - \log(t!(s + t)!) \\ &= (s + t)L + (s + t)\log(1 - (M + t)/2^L) + t\log M + t\log(1 - (t + s)/M) - \log(t!(s + t)!) \\ &\overset{(a)}{\geq} (s + t)L + t\log M - \log(t!(s + t)!) + o(1), \end{aligned}$$

where in inequality (a) we used $\log(1+x) \geq \frac{x}{1+x}$ for any $x > -1$. The second statement is proven as follows. We start from the upper bound on the cardinality of any $(\sigma M, \tau M, \bullet)_{\mathbb{L}}$-correcting code from Theorem 3.17 and obtain

$$\begin{aligned} r(\mathcal{C}) = \log \binom{2^L}{M} - \log|\mathcal{C}| &\geq \log \binom{2^L}{M}\binom{M}{(\sigma+\tau)M}\binom{2^L-M}{\tau M} - \log \binom{2^L}{(1-\sigma)M} \\ &\overset{(a)}{\geq} \log \binom{2^L}{M}\binom{2^L-M}{\tau M} - \log \binom{2^L}{(1-\sigma)M} \\ &\overset{(b)}{=} M \log \frac{\mathrm{e}2^L}{M} + \tau M \log \frac{\mathrm{e}(2^L-M)}{\tau M} - (1-\sigma)M \log \frac{\mathrm{e}2^L}{(1-\sigma)M} + o(M) \\ &\overset{(c)}{=} (\sigma+\tau)M(L - \log M) + O(M), \end{aligned}$$

where for inequality (a) we dropped the term $\log \binom{M}{(\sigma+\tau)M}$ since it is positive and also asymptotically negligible. Equality (b) follows from applying Lemma A.3 to each of the individual terms. Finally, to prove equality (c), we identified the terms of order $O(M)$ together with an application of Lemma A.1 on the term $\log(2^L - M) = L + \log(1 - M/2^L)$. □

This result is particularly interesting, due to the following consideration. Both lost sequences and erroneous sequences do not carry any useful information, since the erroneous sequences can be distorted by an arbitrary number of errors. However, unlike a lost sequence, the erroneous sequence cannot directly be detected by the decoder and therefore, compared to a loss of sequence, requires additional redundancy of roughly $\log M$ bits to be corrected. This result is analogous to the case of standard binary substitution-correcting block-codes of length n, where erasures require a redundancy of only a single symbol, and errors require roughly $\log n$ symbols of redundancy to be corrected. This analogy becomes particularly visible when sequences are indexed and protected by a standard substitution-correcting code, similarly to Construction 3.30 (see Section 3.5.1), but also holds for the general case of any $(s, t, \bullet)_{\mathbb{L}}$-correcting code. However, this seems to be not the case, when the number of lost sequences and erroneous sequences scales with M, since in that case the redundancy only depends on $\sigma + \tau$.

3.4.2 Insertion Errors

In the following, we find code size upper bounds for the case of having a combination of a loss of s sequences and only u insertion errors inside t arbitrary sequences. The sphere packing bound is derived in the following theorem.

Theorem 3.19. *The cardinality of any $(s, t, u)_{\mathbb{I}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies*

$$|\mathcal{C}| \leq \frac{\binom{2^L}{M-s-t}\binom{2^{L+u}}{t}}{\binom{M}{s+t}\binom{S^{\mathbb{I}}(L,u)}{t}}.$$

Proof. We prove the theorem by bounding the error ball size $|B^{\mathbb{I}}(\mathcal{S}, s, t, u)|$ from below for all $\mathcal{S}$, which yields an upper bound on the cardinality of $(s, t, u)_{\mathbb{I}}$-correcting codes by a sphere packing argument. Distinct elements $\mathcal{S}' \in B^{\mathbb{I}}(\mathcal{S}, s, t, u)$ of the error ball can be found in the following way. First, choose two distinct sets $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}} = \{\boldsymbol{x}_{f_1}, \ldots, \boldsymbol{x}_{f_t}\} \subseteq \mathcal{S}$ with $|\mathcal{S}_{\mathsf{L}}| = s$ and $|\mathcal{S}_{\mathsf{E}}| = t$. Further choose the set of erroneous sequences $\mathcal{S}'_{\mathsf{E}} = \{\boldsymbol{x}'_{f_1}, \ldots, \boldsymbol{x}'_{f_t}\}$ such that $\boldsymbol{x}'_{f_i} \in S^{\mathbb{I}}(\boldsymbol{x}_{f_i}, u)$. The

received set $\mathcal{S}'$ is then constructed by $\mathcal{S}' = \mathcal{S}_{\mathsf{C}} \cup \mathcal{S}'_{\mathsf{E}}$, where $\mathcal{S}_{\mathsf{C}} = \mathcal{S} \setminus (\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}})$ are the error-free sequences, as in Definition 3.3. Let $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}}, \mathcal{S}'_{\mathsf{E}}$ and $\widetilde{\mathcal{S}}_{\mathsf{L}}, \widetilde{\mathcal{S}}_{\mathsf{E}}, \widetilde{\mathcal{S}}'_{\mathsf{E}}$ be chosen according to the above described procedure and let $\mathcal{S}'$ and $\widetilde{\mathcal{S}}'$ be the corresponding received sets. We will show that if $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}} \neq \widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}}$ or $\mathcal{S}_{\mathsf{E}} \neq \mathcal{S}'_{\mathsf{E}}$, then $\mathcal{S}' \neq \widetilde{\mathcal{S}}'$. We discuss first the case when $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}} \neq \widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}}$. Here it directly follows that $\mathcal{S}' \neq \widetilde{\mathcal{S}}'$, since the error-free sequences $\mathcal{S}_{\mathsf{C}} = \mathcal{S} \setminus (\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}})$ and $\widetilde{\mathcal{S}}_{\mathsf{C}} = \mathcal{S} \setminus (\widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}})$ of length L are different. In the other case, if $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}} = \widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}}$, it follows that $\mathcal{S}'_{\mathsf{E}} \neq \widetilde{\mathcal{S}}'_{\mathsf{E}}$. Therefore, two different choices of the sets $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}}$ and $\mathcal{S}'_{\mathsf{E}}$ yield different elements in $B^{\mathbb{I}}(\mathcal{S}, s, t, u)$. The number of possible sets $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}}$ is $\binom{M}{s+t}$. For each $\boldsymbol{x}_{f_i} \in \mathcal{S}_{\mathsf{E}}$, we have $S^{\mathbb{I}}(L, u)$ choices for $\boldsymbol{x}'_{f_i} \in S^{\mathbb{I}}(\boldsymbol{x}_{f_i}, u)$. Note that in general the spheres $S^{\mathbb{I}}(\boldsymbol{x}_{f_i}, u)$ are not necessarily distinct over different i, however we can still bound the number of choices for the set $\mathcal{S}'_{\mathsf{E}}$ from below as follows. A conservative argument suggests that the smallest number of choices for $\mathcal{S}'_{\mathsf{E}}$ is attained when the $S^{\mathbb{I}}(\boldsymbol{x}_{f_i}, u)$ perfectly agree for all i and in this case, we have exactly $\binom{S^{\mathbb{I}}(L,u)}{t}$ choices for $\mathcal{S}'_{\mathsf{E}}$. Hence, in total, there are at least $\binom{M}{s+t}\binom{S^{\mathbb{I}}(L,u)}{t}$ ways to choose $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}}$ and $\mathcal{S}'_{\mathsf{E}}$ and therefore $|B^{\mathbb{I}}(\mathcal{S}, s, t, u)| \geq \binom{M}{s+t}\binom{S^{\mathbb{I}}(L,u)}{t}$ for all $\mathcal{S} \in \mathcal{X}_M^L$. Each such constructed received set $\mathcal{S}'$ consists of $M - s - t$ sequences of length L and t sequences of length $L + u$. There are in total $\binom{2^L}{M-s-t}\binom{2^{L+u}}{t}$ such sets, which yields the theorem by a sphere packing argument. □

Note that Theorem 3.19 provides a valid upper bound for any parameter M, L, s, t, u. For the case of deletion errors or combinations of insertions and deletions, formulating a sphere packing bound based on the minimum error ball size yields a weak bound, since the minimum deletion ball size is $|B^{\mathbb{D}}(\mathbf{0}, u)| = u + 1$, obtained by, e.g., the all-zero word $\mathbf{0}$. Therefore, a conservative analysis similar to Theorem 3.19 would yield unsatisfactory results. However, an asymptotic analysis, which yields asymptotically tighter bounds is possible, as we will see in Theorem 3.27. Further, Theorem 3.19 can be used to infer the following asymptotic statement.

Corollary 3.20. *For fixed $s, t \in \mathbb{N}_0$ and fixed $0 < \beta < 1$, the redundancy of any $(s, t, u)_{\mathbb{I}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is asymptotically at least*

$$r(\mathcal{C}) \geq sL + tu(\log L - 1) - \log(s + t)! + o(1),$$

when $M \to \infty$ and $M = 2^{\beta L}$.

Proof. The statement directly follows from Theorem 3.19 as follows.

$$r(\mathcal{C}) = \log|\mathcal{X}_M^L| - \log|\mathcal{C}| \geq \log\binom{2^L}{M}\binom{M}{s+t}\binom{S^{\mathbb{I}}(L,u)}{t} - \log\binom{2^L}{M-s-t}\binom{2^{L+u}}{t}$$

Thus, expanding the binomial coefficients $\binom{2^L}{M}$ and $\binom{2^L}{M-s-t}$, we obtain

$$r(\mathcal{C}) \geq \log\frac{(2^L - M + s + t)!(M - s - t)!}{(2^L - M)!M!} + \log\binom{M}{s+t}\binom{S^{\mathbb{I}}(L,u)}{t} - \log\binom{2^{L+u}}{t}.$$

We proceed with bounding the factorials as follows. It is immediate that for any $a, b \in \mathbb{N}$ with $a \leq b$, it holds that $\frac{a!}{b!} \geq \frac{1}{b^{b-a}}$ and when $a \geq b$, it holds that $\frac{a!}{b!} \geq b^{b-a}$. Therefore

$$\begin{aligned} r(\mathcal{C}) &\geq (s+t)\log\frac{(2^L - M)}{M} + \log\binom{M}{s+t}\binom{S^{\mathbb{I}}(L,u)}{t} - \log\binom{2^{L+u}}{t} \\ &\overset{(a)}{\geq} (s+t)(L - \log M) + (s+t)\log M + tu\log L - t(L+u) - \log(s+t)! + o(1) \\ &= tL + tu(\log L - 1) - \log(s+t)! \end{aligned}$$

where in inequality (a) we additionally used Lemma A.1 on the first summand and Lemma A.2 on the remaining binomial coefficients. □

3.4.3 Substitution Errors

We now derive asymptotic sphere packing bounds on the code size for $(s,t,u)_{\mathbb{S}}$-correcting codes. This depicts the case of only substitution errors inside the sequences. We will mainly focus our attention towards large number of sequences M. As discussed before, the error ball sizes depend on the center $\mathcal{S}$. However, as it turns out, asymptotically the error balls have similar sizes. We will start by finding a lower bound on the error ball size for a set $\mathcal{S}$.

Lemma 3.21. *Let $\mathcal{Y} \subseteq \mathcal{S} \in \mathcal{X}_M^L$ be an u-substitution-correcting code, i.e. $B^{\mathbb{S}}(\boldsymbol{y}_1,u)\cap B^{\mathbb{S}}(\boldsymbol{y}_2,u)=\emptyset$ for all $\boldsymbol{y}_1,\boldsymbol{y}_2\in\mathcal{Y}$ and $\boldsymbol{y}_1\neq\boldsymbol{y}_2$. Further, let $s+t\leq|\mathcal{Y}|$. Then,*

$$\left|\left\{\mathcal{S}'\in B^{\mathbb{S}}(\mathcal{S},s,t,u):|\mathcal{S}'|\leq M-s\right\}\right|\geq\binom{|\mathcal{Y}|}{s}\binom{|\mathcal{Y}|-s}{t}\left(B^{\mathbb{S}}(L,u)-1\right)^t.$$

Proof. The lower bound will be proven by identifying and counting specific patterns of a loss of sequences and errors in sequences of $\mathcal{S}$ that lead to distinct channel outputs $\mathcal{S}'\in B^{\mathbb{S}}(\mathcal{S},s,t,u)$ with $|\mathcal{S}'|\leq M-s$. Let $\mathcal{Y}=\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{|\mathcal{Y}|}\}$. The sets of stored sequences in the error balls around the elements in $\mathcal{Y}$ are denoted by $\mathcal{S}_i\triangleq\mathcal{S}\cap B^{\mathbb{S}}(\boldsymbol{y}_i,u)$. Similarly, the sets of received sequences in these error balls are $\mathcal{S}_i'\triangleq\mathcal{S}'\cap B^{\mathbb{S}}(\boldsymbol{y}_i,u)$. Note that the sets $B^{\mathbb{S}}(\boldsymbol{y}_i,u)$ and thus also the sets $\mathcal{S}_i$ are pairwise distinct, since $\mathcal{Y}$ is an u-substitution-correcting code. We further define the selector function for sequences $\boldsymbol{a},\boldsymbol{b},\boldsymbol{x}\in\Sigma_2^L$ as

$$\mathbb{I}_{\boldsymbol{x}}^{\mathcal{S}}(\boldsymbol{a},\boldsymbol{b})=\begin{cases}\boldsymbol{a}, & \text{if } \boldsymbol{x}\notin\mathcal{S}\\ \boldsymbol{b}, & \text{otherwise}\end{cases}.$$

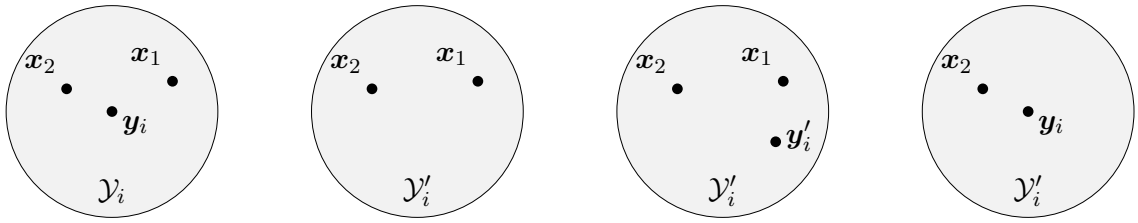
Distinct channel outputs $\mathcal{S}'\in B^{\mathbb{S}}(\mathcal{S},s,t,u)$ are obtained in the following manner. First, choose two distinct sets $\mathcal{L}=\{l_1,\ldots,l_s\}\subseteq[|\mathcal{Y}|]$ with $|\mathcal{L}|=s$ and $\mathcal{F}=\{f_1,\ldots,f_t\}\subseteq[|\mathcal{Y}|]$ with $|\mathcal{E}|=t$ and a collection of error vectors $\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_t$, where $\boldsymbol{e}_j\in\Sigma_2^L$ are non-zero vectors of weight at most $\mathsf{wt}_{\mathsf{H}}(\boldsymbol{e}_j)\leq u$. We will show that by for each choice of $\mathcal{L},\mathcal{F}$, $\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_t$ we can construct a unique word $\mathcal{S}'\in B^{\mathbb{S}}(\mathcal{S},s,t,u)$ in the following manner. First, all sequences $\mathcal{S}_{\mathsf{L}}=\{\boldsymbol{y}_{l_1},\ldots,\boldsymbol{y}_{l_s}\}$ are lost. Denote $\boldsymbol{y}'_{f_j}\triangleq\boldsymbol{y}_{f_j}+\boldsymbol{e}_j$, $1\leq j\leq t$. The set $\mathcal{S}_{\mathsf{E}}$ of erroneous sequences is then chosen as

$$\mathcal{S}_{\mathsf{E}}=\bigcup_{j=1}^{t}\left\{\mathbb{I}_{\boldsymbol{y}'_{f_j}}^{\mathcal{S}}(\boldsymbol{y}_{f_j},\boldsymbol{y}'_{f_j})\right\}.$$

In other words, if $\boldsymbol{y}'_{f_j}\notin\mathcal{S}$ we choose the sequence, which will be distorted by errors to be $\boldsymbol{y}_{f_j}$ and otherwise we choose it to be the sequence $\boldsymbol{y}'_{f_j}\in\mathcal{S}$. The erroneous outcomes of the sequences in $\mathcal{S}_{\mathsf{E}}$ are now constructed by

$$\mathcal{S}'_{\mathsf{E}}=\bigcup_{j=1}^{t}\left\{\mathbb{I}_{\boldsymbol{y}'_{f_j}}^{\mathcal{S}}(\boldsymbol{y}'_{f_j},\boldsymbol{y}_{f_j})\right\}.$$

That is if $\boldsymbol{y}'_{f_j}\notin\mathcal{S}$, we have $\boldsymbol{y}_{f_j}\in\mathcal{S}_{\mathsf{E}}$ and we add $\boldsymbol{e}_j$ to that sequence to obtain $\boldsymbol{y}'_j\in\mathcal{S}'_{\mathsf{E}}$. If $\boldsymbol{y}'_j\in\mathcal{S}$, $\boldsymbol{y}'_{f_j}\in\mathcal{S}_{\mathsf{E}}$ is the sequence which is distorted and we add $-\boldsymbol{e}_j$, resulting in $\boldsymbol{y}_{f_j}\in\mathcal{S}'_{\mathsf{E}}$. It is important to note that by this choice of error patterns, the erroneous sequence $\boldsymbol{y}'_{f_j}\in B^{\mathbb{S}}(\boldsymbol{y}_{f_j},u)$



(a) No errors. (b) $\boldsymbol{y}_i$ is lost. (c) $\boldsymbol{y}_i$ is erroneous, $\boldsymbol{y}'_i \neq \boldsymbol{x}_j$. (d) $\boldsymbol{x}_j \neq \boldsymbol{y}_i$ is erroneous, $\boldsymbol{x}'_j = \boldsymbol{y}_i$.

Figure 3.5: Cases for error patterns in Lemma 3.21.

and therefore will never be present in another error ball $B_u^{\mathbb{S}}(\boldsymbol{y}), \boldsymbol{y} \in \mathcal{Y} \setminus \{\boldsymbol{y}_{f_j}\}$, since $\mathcal{Y}$ is an u-substitution-correcting code. The received set is now $\mathcal{S}' = \mathcal{S}_{\mathsf{C}} \cup \mathcal{S}'_{\mathsf{E}}$, where $\mathcal{S}_{\mathsf{C}} = \mathcal{S} \setminus (\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}})$ are the error-free sequences, as in Definition 3.3. We will show now that two choices $\mathcal{L}, \mathcal{F}, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_t$ and $\widetilde{\mathcal{L}}, \widetilde{\mathcal{F}}, \widetilde{\boldsymbol{e}}_1, \ldots, \widetilde{\boldsymbol{e}}_t$ yield different received sets $\mathcal{S}'$ and $\widetilde{\mathcal{S}}'$, if (and only if) they differ in at least one of the components, i.e., $\mathcal{L} \neq \widetilde{\mathcal{L}}$, $\mathcal{F} \neq \widetilde{\mathcal{F}}$, or $\boldsymbol{e}_j \neq \widetilde{\boldsymbol{e}}_j$ for some j. For each $i \in [|\mathcal{Y}|]$, we distinguish between the following three different cases (visualized in Figure 3.5) and state the composition of the resulting received parts $\mathcal{S}'_i = \mathcal{S} \cap B^{\mathbb{S}}(\boldsymbol{y}_i, u)$.

- $i \notin (\mathcal{L} \cup \mathcal{F})$: $\mathcal{S}'_i = \mathcal{S}_i$
- $i \in \mathcal{L}$: $\mathcal{S}'_i = \mathcal{S}_i \setminus \{\boldsymbol{y}_i\}$
- $i \in \mathcal{F}$: $\mathcal{S}'_i = (\mathcal{S}_i \setminus \{\boldsymbol{y}_i\}) \cup \{\boldsymbol{y}'_i\}$ or $\mathcal{S}'_i = \mathcal{S}_i \setminus \{\boldsymbol{y}'_i\}$.

Here $\boldsymbol{y}' \in B^{\mathbb{S}}(\boldsymbol{y}_i, u) \setminus \mathcal{S}$ is the erroneous outcome of the sequence $\boldsymbol{y}_i$. Comparing the outputs $\mathcal{S}'_i$ for these three cases, it is verified that each case yields different $\mathcal{S}'_i$. Now, if $\mathcal{L} \neq \widetilde{\mathcal{L}}$ there is at least one i such that $i \in \mathcal{L}$ and $i \notin \widetilde{\mathcal{L}}$ and if $\mathcal{F} \neq \widetilde{\mathcal{F}}$ there is at least one i such that $i \in \mathcal{F}$ and $i \notin \widetilde{\mathcal{F}}$. Therefore, in both cases it follows that $\mathcal{S}'_i \neq \widetilde{\mathcal{S}}'_i$ for some i and consequently $\mathcal{S}' \neq \widetilde{\mathcal{S}}'$. Further, if both $\mathcal{L} = \widetilde{\mathcal{L}}$ and $\mathcal{F} = \widetilde{\mathcal{F}}$, there exists some j such that $\boldsymbol{e}_j \neq \widetilde{\boldsymbol{e}}_j$. Therefore, the corresponding results $\boldsymbol{y}'_{f_j}$ will be different and thus $\mathcal{S}'_{f_j} \neq \widetilde{\mathcal{S}}'_{f_j}$ according to the third case above. This proves that each $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}}, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_t$ yields a unique word in $B^{\mathbb{S}}(\mathcal{S}, s, t, u)$. Finally, by construction, all $\mathcal{S}'$ satisfy $|\mathcal{S}'| \leq M - s$ and there are $\binom{|\mathcal{Y}|}{s}\binom{|\mathcal{Y}|-s}{t}$ possible choices for the sets $\mathcal{L}$ and $\mathcal{F}$ and $(B^{\mathbb{S}}(L, u) - 1)^t$ non-zero error patterns $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t$. □

This means, that if a set $\mathcal{S} \in \mathcal{X}_M^L$ contains an u-substitution-correcting code $\mathcal{Y}$ with cardinality $|\mathcal{Y}|$, the error ball has size at least $|B^{\mathbb{S}}(\mathcal{S}, s, t, u)| \geq \binom{|\mathcal{Y}|}{s}\binom{|\mathcal{Y}|-s}{t}(B^{\mathbb{S}}(L, u) - 1)^t$. Interestingly, for an appropriate choice of parameters, most of the sets $\mathcal{S} \in \mathcal{X}_M^L$ have the property of containing a large u-error-correcting code of size that is close to M. To establish this fact, we need the following lemma that proves an upper bound on the number of sets that do not contain a large error-correcting code.

Lemma 3.22. *Let $\mathcal{Y} \subseteq \mathcal{S}$ be the largest u-error-correcting code (error type $\mathbb{T}$), with the property $B^{\mathbb{T}}(\boldsymbol{y}_1, u) \cap B^{\mathbb{T}}(\boldsymbol{y}_2, u) = \emptyset$ for all $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}$ and $\boldsymbol{y}_1 \neq \boldsymbol{y}_2$. The number of sets $\mathcal{S} \subseteq \mathcal{X}_M^L$ with $|\mathcal{Y}| \leq K$, denoted as $D(K)$, is at most*

$$D(K) \leq \binom{2^L}{K}\binom{KV_u^{\mathbb{T}}}{M-K},$$

where

$$V^{\mathbb{T}}(u) = \max_{\boldsymbol{x}\in\Sigma_2^L} |\{\boldsymbol{y} \in \Sigma_2^L : B^{\mathbb{T}}(\boldsymbol{x}, u) \cap B^{\mathbb{T}}(\boldsymbol{y}, u) \neq \emptyset\}|$$

is the maximum over the number of sequences $\boldsymbol{y} \in \Sigma_2^L$ *that have intersecting error balls with* $\boldsymbol{x} \in \Sigma_2^L$.

Proof. Consider the following procedure on a set $\mathcal{S} \in \mathcal{X}_M^L$ whose largest u-error-correcting subset $\mathcal{Y} \subseteq \mathcal{S}$ has size at most K. Write $\mathcal{S}^{(1)} \triangleq \mathcal{S}$. Take an arbitrary word $\boldsymbol{x}^{(1)} \in \mathcal{S}^{(1)}$ and remove all words $\boldsymbol{y} \in \Sigma_2^L$ with intersecting error balls, i.e. $B^{\mathbb{T}}(\boldsymbol{x}^{(1)}, u) \cap B^{\mathbb{T}}(\boldsymbol{y}, u) \neq \emptyset$ from $\mathcal{S}^{(1)}$. Then select an arbitrary sequence from the resulting set $\mathcal{S}^{(2)}$, and, again, remove all elements with intersecting error balls. Continue this procedure until $S^{(j+1)} = \emptyset$. This procedure will stop after at most $j \leq K$ steps, since otherwise $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{K+1}$ would form an u-error-correcting code. Hence, each such set $\mathcal{S}$ can be constructed by first selecting K arbitrary, distinct words $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ and then choosing the remaining $M - K$ words to have intersecting error balls with at least one of the $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$. □

While the bound from Lemma 3.22 may not seem particularly strong, it can be used to show that the number of sets that do not contain an u-substitution-correcting code of large size is negligible with respect to the sets that do contain an u-substitution-correcting code. We will elaborate this result and use it in the following to prove an upper bound on the size of $(s, t, u)_{\mathbb{S}}$-correcting codes.

Theorem 3.23. *For fixed* $s, t, u \in \mathbb{N}_0$ *and* $0 < \beta < 1$, *any* $(s, t, u)_{\mathbb{S}}$*-correcting code* $\mathcal{C} \subseteq \mathcal{X}_M^L$ *satisfies*

$$|\mathcal{C}| \leq \frac{\binom{2^L}{M-s}}{\binom{M}{s}\binom{M-s}{t}\binom{L}{u}^t}(1 + o(1)),$$

when $M \to \infty$ *with* $M = 2^{\beta L}$. *The redundancy is at least*

$$r(\mathcal{C}) \geq sL + t \log M + tu \log L - \log\left(s!t!u!^t\right) + o(1),$$

Proof. Denote by $B(\mathcal{S}) \triangleq \{\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, s, t, u) : |\mathcal{S}'| \leq M - s\}$ the set of possible received words with at most $M - s$ sequences. Denote further by $\mathcal{X}_{\mathsf{e}} \subseteq \mathcal{X}_M^L$ the set of all $\mathcal{S} \in \mathcal{X}_M^L$, which contain some u-substitution-correcting code $\mathcal{Y} \subseteq \mathcal{S}$ of size larger than $|\mathcal{Y}| > M - y(M)$, where we define $y(M) = M/\log M$. The remaining sets are comprised in $\mathcal{X}_{\mathsf{e}}^{\mathsf{c}} = \mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{e}}$. With the partition $\mathcal{X}_{\mathsf{e}} \cup \mathcal{X}_{\mathsf{e}}^{\mathsf{c}} = \mathcal{X}_M^L$, it follows that the cardinality of any $(s, t, u)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is at most

$$|\mathcal{C}| = |\mathcal{C} \cap \mathcal{X}_{\mathsf{e}}| + |\mathcal{C} \cap \mathcal{X}_{\mathsf{e}}^{\mathsf{c}}| \leq \frac{\left|\bigcup_{\mathcal{S}\in\mathcal{X}_{\mathsf{e}}} B(\mathcal{S})\right|}{\min_{\mathcal{S}\in\mathcal{X}_{\mathsf{e}}} |B(\mathcal{S})|} + |\mathcal{X}_{\mathsf{e}}^{\mathsf{c}}|.$$

The first term follows from a sphere packing bound on all sets $\mathcal{S} \in \mathcal{X}_{\mathsf{e}}$. The numerator counts the total number of possible channel outputs, when an arbitrary $\mathcal{S} \in \mathcal{X}_{\mathsf{e}}$ is the input. The denominator is a lower bound on the error ball size for all sets $\mathcal{S} \in \mathcal{X}_{\mathsf{e}}$. Since each channel output is a set of sequences of size $M - s - t$ up to $M - s$, we have

$$\left|\bigcup_{\mathcal{S}\in\mathcal{X}_{\mathsf{e}}} B(\mathcal{S})\right| \leq \sum_{i=0}^{t} \binom{2^L}{M - s - i}.$$

From Lemma 3.21 it is known that

$$\min_{\mathcal{S}\in\mathcal{X}_{\mathsf{e}}}|B(\mathcal{S})| \geq \binom{M-y(M)}{s,t}(B^{\mathbb{S}}(L,u)-1)^t,$$

and applying Lemma 3.22, we find that $|\mathcal{X}_{\mathsf{e}}^{\mathsf{c}}| \leq D(M-y(M))$. It follows that

$$\begin{aligned}|\mathcal{C}| &\leq \frac{\sum_{i=0}^{t}\binom{2^L}{M-s-i}}{\binom{M-y(M)}{s}\binom{M-y(M)-s}{t}(B^{\mathbb{S}}(L,u)-1)^t} + D(M-y(M))\\ &= \frac{\sum_{i=0}^{t}\binom{2^L}{M-s-i}}{\binom{M-y(M)}{s}\binom{M-y(M)-s}{t}(B^{\mathbb{S}}(L,u)-1)^t}(1+\Delta),\end{aligned}$$

where Δ accounts for $D(M-y(M))$ and is defined implicitly as in the following equation. We will show that for our choice of $y(M)$, the first summand dominates the bound, i.e. $\Delta \to 0$ for $M \to \infty$. We obtain

$$\begin{aligned}\log\Delta = \log\frac{D(M-y(M))\binom{M-y(M)}{s}\binom{M-y(M)-s}{t}(B^{\mathbb{S}}(L,u)-1)^t}{\sum_{i=0}^{t}\binom{2^L}{M-s-i}} &\overset{(a)}{\leq} \log\frac{D(M-y(M))}{\sum_{i=0}^{t}\binom{2^L}{M-s-i}} + O(L)\\ &\overset{(b)}{\leq} \frac{\binom{2^L}{M-y(M)}\binom{(M-y(M))B^{\mathbb{S}}(L,2u)}{y(M)}}{\binom{2^L}{M-s}} + O(L) \overset{(c)}{\leq} -\frac{1-\beta}{\beta}M + o(M),\end{aligned}$$

where for inequality (a) we used $\log\binom{M-y(M)}{s}\binom{M-y(M)-s}{t} = O(\log M) = O(L)$ and further the fact that $t\log(B^{\mathbb{S}}(L,u)-1) = O(\log L)$. Inequality (b) follows from Lemma 3.22 together with $V^{\mathbb{S}}(u) = B^{\mathbb{S}}(L,2u)$. Inequality (c) can be shown by an application of Lemma A.6 with $z(L) = 2^L/((M-y(M))B^{\mathbb{S}}(L,2u))$. Therefore, $\Delta \to 0$, as $M \to \infty$ and henceforth $D(M-y(M))$ is asymptotically negligible. We obtain for any $(s,t,u)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$

$$|\mathcal{C}| \leq \frac{\sum_{i=0}^{t}\binom{2^L}{M-s-i}}{\binom{M-y(M)}{s}\binom{M-y(M)-s}{t}(B^{\mathbb{S}}(L,u)-1)^t}(1+o(1)) = \frac{\binom{2^L}{M-s}}{\binom{M}{s}\binom{M-s}{t}\binom{L}{u}^t}(1+o(1)).$$

The redundancy is asymptotically at least

$$\begin{aligned}r(\mathcal{C}) = \log\frac{\binom{2^L}{M}}{|\mathcal{C}|} &\geq \log\frac{\binom{2^L}{M}\binom{M}{s}\binom{M-s}{t}\binom{L}{u}^t}{\binom{2^L}{M-s}} + o(1)\\ &\geq s\log(2^L-M) + t\log\left(ML^u\right) - \log(s!t!u!^t) + o(1)\\ &= sL + t\log M + tu\log L - \log\left(s!t!u!^t\right) + o(1),\end{aligned}$$

where we used Lemma A.2 to simplify the asymptotic behavior of the binomial coefficients. □

In particular, for $s=0$ and $u=1$, the redundancy of any $(0,t,1)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is at least $t\log(ML) - \log t!$ bits. Note that this coincides with the results from [SRB21] for $t=1$. Comparing the bound on the redundancy stated in Theorem 3.23 with the well known sphere packing bound for conventional u-substitution-correcting block codes, $\log B^{\mathbb{S}}(L,u)$, yields an interesting interpretation of the $(0,t,1)_{\mathbb{S}}$ channel. While it seems intuitive that the redundancy required is at least $t\log(ML) - \log t!$ bits, since there are t errors inside a total of ML symbols,

it is interesting that from a sphere packing point of view, the fact the sequences are not ordered does appear to require as much redundancy as not knowing the distribution of the errors in an ordered array. While Theorem 3.23 is formulated for a fixed number of errors s, t, we will find a bound for the case, when number of erroneous sequences t is scaling with M in the following.

Theorem 3.24. *For fixed $s, u \in \mathbb{N}_0$ and fixed $0 < \beta < 1$, any $(s, M-s, u)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies*

$$r(\mathcal{C}) \geq Mu \log L + O(M),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. We follow a similar outline as in the proof for Theorem 3.23. Denote the set of possible received words with at most $M-s$ sequences by $B(\mathcal{S}) \triangleq \{\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, s, t, u) : |\mathcal{S}'| \leq M - s\}$. Further denote $\mathcal{X}_{\mathsf{e}} \subseteq \mathcal{X}_M^L$ as the set of all $\mathcal{S} \in \mathcal{X}_M^L$, which contain an u-substitution-correcting code $\mathcal{Y} \subseteq \mathcal{S}$ of size $|\mathcal{Y}| > M - y(M)$, where we define $y(M) = M/\log\log M$ and $\mathcal{X}_{\mathsf{e}}^{\mathsf{c}} = \mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{e}}$. Allowing only $t = M - s - y(M)$ erroneous sequences, we can apply Lemma 3.21 and obtain

$$|B(\mathcal{S})| \geq \binom{M - y(M)}{s} (B^{\mathbb{S}}(L, u) - 1)^{M - y(M) - s},$$

for all $\mathcal{S} \in \mathcal{X}_{\mathsf{e}}$. It follows that

$$|\mathcal{C}| \leq \frac{\sum_{i=0}^{M-y(M)-s} \binom{2^L}{M-s-i}}{\binom{M-y(M)}{s}(B^{\mathbb{S}}(L,u)-1)^{M-y(M)-s}}(1+\Delta),$$

using arguments analogous to those in the proof of Theorem 3.23. We will show that $\Delta \to 0$ for $M \to \infty$. We obtain

$$\begin{aligned}
\log \Delta &= \log \frac{\binom{M-y(M)}{s}(B^{\mathbb{S}}(L,u)-1)^{M-y(M)-s} D(M-y(M))}{\sum_{i=0}^{M-y(M)-s}\binom{2^L}{M-s-i}} \\
&\overset{(a)}{\leq} \log \frac{\binom{2^L}{M-y(M)}\binom{(M-y(M))B^{\mathbb{S}}(L,2u)}{y(M)}}{\binom{2^L}{M-s}} + Mu\log L + O(L) \\
&\overset{(b)}{\leq} -\frac{ML(1-\beta)}{\log\log M} + Mu\log L + o\left(\frac{M}{\log\log M}\right) = -\frac{ML(1-\beta)}{\log(\beta L)} + O(M\log L),
\end{aligned}$$

where in inequality (a) we used $\log\binom{M-y(M)}{s} = O(L)$. For inequality (b) we applied Lemma A.6 with $z(L) = 2^L/((M-y(M))B^{\mathbb{S}}(L,2u))$. Therefore, $\Delta \to 0$, as $M \to \infty$. We obtain for any $(s, M-s, u)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$

$$|\mathcal{C}| \leq \frac{\sum_{i=0}^{M-y(M)-s}\binom{2^L}{M-s-i}}{\binom{M-y(M)}{s}(B^{\mathbb{S}}(L,u)-1)^{M-y(M)-s}}(1+o(1)) \leq \frac{\binom{2^L+M}{M-s}}{\binom{M}{s}\binom{L}{u}^{M-y(M)-s}}(1+o(1)),$$

where we used that $\sum_{i=0}^{m}\binom{n}{m} \leq \binom{n+m}{m}$ for any $n, m \in \mathbb{N}$ to bound the numerator. Therefore, the redundancy satisfies

$$\begin{aligned}
r(\mathcal{C}) &= \log\frac{\binom{2^L}{M}}{|\mathcal{C}|} \geq \log\frac{\binom{2^L}{M}\binom{M}{s}\binom{L}{u}^{M-y(M)-s}}{\binom{2^L+M}{M-s}} + o(1) \\
&= \log\frac{(2^L)!(2^L+s)!}{(2^L-M)!s!(2^L+M)!} + (M-y(M)-s)\log\binom{L}{u}.
\end{aligned}$$

In order to bound the first term, we use that for any $a, b \in \mathbb{N}$ with $a \leq b$, it holds that $\frac{a!}{b!} \geq \frac{1}{b^{b-a}}$ and when $a \geq b$, it holds that $\frac{a!}{b!} \geq b^{b-a}$. We obtain

$$r(\mathcal{C}) \geq M \log \frac{2^L - M}{2^L + M} + s \log(2^L + M) + (M - y(M) - s) \log \binom{L}{u}.$$

Finally, applying Lemma A.1, we see that the first term is of order $o(M)$ and using that $y(M) \log L = O(M)$, we obtain

$$r(\mathcal{C}) \leq Mu \log L + O(M).$$

□

3.4.4 Deletion Errors

We will now turn to derive an asymptotic bound on the cardinality of $(s, t, u)_{\mathbb{D}}$-correcting codes. Note that in general it is also possible to use the technique that we present here for insertion errors. However, we already obtained a bound in Theorem 3.19 and therefore focus on the case of deletions in the sequel. Since the deletion ball is non-uniform, i.e., the balls around different words can have different sizes and it is thus not directly possible to use an analogue of Lemma 3.21 as in Theorem 3.23. We will therefore slightly adapt our arguments and use the fact that, although the deletion ball size is non-uniform, most of the deletion balls have a similar size. In, particular, it has been shown in [Lev66] that

$$|S^{\mathbb{D}}(\boldsymbol{x}, u)| \geq \binom{||\boldsymbol{x}|| - u + 1}{u}$$

and most words $\boldsymbol{x} \in \Sigma_2^L$ have roughly $L/2$ runs. We will elaborate this result in the following.

Lemma 3.25. *Let $\rho \in \mathbb{N}$. The number of words with less than $L/2 - \rho$ runs satisfies*

$$\left| \left\{ \boldsymbol{x} \in \Sigma_2^L : ||\boldsymbol{x}|| < \frac{L}{2} - \rho \right\} \right| \leq \frac{2^L}{\mathrm{e}^{\frac{2\rho^2}{L}}}.$$

Proof. The number of words $\boldsymbol{x} \in \Sigma_2^L$ with exactly i runs, i.e., $||\boldsymbol{x}|| = i$ is given by $2\binom{L-1}{i-1}$. Therefore, the number of words with less than $L/2 - \rho$ runs is given by

$$|\{\boldsymbol{x} \in \Sigma_2^L : ||\boldsymbol{x}|| < L/2 - \rho\}| = 2 \sum_{i=1}^{L/2-\rho-1} \binom{L-1}{i-1} \overset{(a)}{\leq} \sum_{i=1}^{L/2-\rho} \binom{L}{i} \overset{(b)}{\leq} \frac{2^L}{\mathrm{e}^{\frac{2\rho^2}{L}}},$$

where we used $\binom{L-1}{i-1} \leq \frac{1}{2}\binom{L}{i}$ for $i \leq \frac{L}{2}$ in inequality (a) and Hoeffding's inequality [Hoe63] (see [Ver18, Thm 2.2.2] for an application to binomial tails) on the binomial sum in (b). □

Next, we find a lower bound on the ball size $B^{\mathbb{D}}(\mathcal{S}, s, t, u)$, for sets, which contain a deletion-correcting code.

Lemma 3.26. *Let $\mathcal{Y} \subseteq \mathcal{S} \in \mathcal{X}_M^L$ be an u-deletion-correcting code, i.e. $B^{\mathbb{D}}(\boldsymbol{y}_1, u) \cap B^{\mathbb{D}}(\boldsymbol{y}_2, u) = \emptyset$ for all $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}$ and $\boldsymbol{y}_1 \neq \boldsymbol{y}_2$. Further, let $s + t \leq |\mathcal{Y}|$. Then,*

$$|B^{\mathbb{D}}(\mathcal{S}, s, t, u)| \geq \sum_{\substack{\mathcal{S}_{\mathsf{E}}, \mathcal{S}_{\mathsf{L}} \subseteq \mathcal{Y}, \mathcal{S}_{\mathsf{E}} \cap \mathcal{S}_{\mathsf{L}} = \emptyset, \\ |\mathcal{S}_{\mathsf{L}}| = s, |\mathcal{S}_{\mathsf{E}}| = t}} \prod_{\boldsymbol{y} \in \mathcal{S}_{\mathsf{E}}} |S^{\mathbb{D}}(\boldsymbol{y}, u)|,$$

Proof. We will find a lower bound on the number of words inside the error ball $|B^{\mathbb{D}}(\mathcal{S}, s, t, u)|$ by counting distinct elements $\mathcal{S}' \in B^{\mathbb{D}}(\mathcal{S}, s, t, u)$ in the following way. Choose two arbitrary distinct sets $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}} = \{\boldsymbol{y}_{e_1}, \ldots, \boldsymbol{y}_{e_t}\} \subseteq \mathcal{Y}$ with $|\mathcal{S}_{\mathsf{L}}| = s$ and $|\mathcal{S}_{\mathsf{E}}| = t$ and choose a set of erroneous outcomes $\mathcal{S}'_{\mathsf{E}} = \{\boldsymbol{y}'_{e_1}, \ldots, \boldsymbol{y}'_{e_t}\}$, where $\boldsymbol{y}'_{e_i} \in S^{\mathbb{D}}(\boldsymbol{y}_{e_i}, u)$. Note that we delete exactly u symbols from each $\boldsymbol{y}_{e_i}$ and thus $\boldsymbol{y}'_{e_i} \in \Sigma_2^{L-u}$. Denote by $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}}, \mathcal{S}'_{\mathsf{E}}$ and $\widetilde{\mathcal{S}}_{\mathsf{L}}, \widetilde{\mathcal{S}}_{\mathsf{E}}, \widetilde{\mathcal{S}}'_{\mathsf{E}}$ two different choices of error realizations and let $\mathcal{S}'$ and $\widetilde{\mathcal{S}}'$ be the corresponding received sets. If $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}} \neq \widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}}$, then $\mathcal{S}' \neq \widetilde{\mathcal{S}}'$, as the resulting error-free sequences in $\mathcal{S}'$ and $\widetilde{\mathcal{S}}$ of length L are different. In the case $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}} = \widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}}$ and $\mathcal{S}_{\mathsf{E}} \neq \widetilde{\mathcal{S}}_{\mathsf{E}}$, it follows that $\mathcal{S}'_{\mathsf{E}} \neq \widetilde{\mathcal{S}}'_{\mathsf{E}}$, as the erroneous outcomes are chosen out of the radius u deletion spheres from an u-deletion-correcting code. Finally, if $\mathcal{S}_{\mathsf{L}} \cup \mathcal{S}_{\mathsf{E}} = \widetilde{\mathcal{S}}_{\mathsf{L}} \cup \widetilde{\mathcal{S}}_{\mathsf{E}}$ and $\mathcal{S}_{\mathsf{E}} = \widetilde{\mathcal{S}}_{\mathsf{E}}$ it follows that $\mathcal{S}_{\mathsf{L}} = \widetilde{\mathcal{S}}_{\mathsf{L}}$ and thus $\mathcal{S}'_{\mathsf{E}} \neq \widetilde{\mathcal{S}}'_{\mathsf{E}}$ as we chose $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}}, \mathcal{S}'_{\mathsf{E}}$ and $\widetilde{\mathcal{S}}_{\mathsf{L}}, \widetilde{\mathcal{S}}_{\mathsf{E}}, \widetilde{\mathcal{S}}'_{\mathsf{E}}$ to be different. Hence, for each choice of error realizations $\mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}}, \mathcal{S}'_{\mathsf{E}}$, we obtain a unique element in $B^{\mathbb{D}}(\mathcal{S}, s, t, u)$. Counting the number of choices yields the lemma. □

This allows to formulate the following theorem.

Theorem 3.27. *For fixed $s, t, u \in \mathbb{N}_0$ and $0 < \beta < 1$, any $(s, t, u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies*

$$|\mathcal{C}| \leq \frac{\binom{2^L}{M-s-t}\binom{2^{L-u}}{t}}{\binom{M}{s}\binom{M-s}{t}\binom{L/2}{u}^t}(1 + o(1))$$

when $M \to \infty$ with $M = 2^{\beta L}$. The redundancy is thus at least

$$r(\mathcal{C}) \geq sL + tu \log L - \log(s!u!^t) + o(1).$$

Proof. We abbreviate by $B(\mathcal{S}) \triangleq \{\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, s, t, u) : |\mathcal{S}' \cap \Sigma_2^L| = M - s - t, |\mathcal{S}' \cap \Sigma_2^{L-u}| = t\}$ the set of possible received words that consist of $M - s - t$ sequences of length L and t sequences of length $L - u$. Denote by $\mathcal{X}_{\mathsf{r}} \subseteq \mathcal{X}_M^L$, the set of all $\mathcal{S} \in \mathcal{X}_M^L$, which contain more than $M - y(M)$ sequences $\boldsymbol{x} \in \Sigma_2^L$ with $||\boldsymbol{x}|| \geq L/2 - \rho(L)$, where we choose $y(M) = M/\log M$ and $\rho(L) = \sqrt{L \ln L}$. Further, let $\mathcal{X}_{\mathsf{e}} \subseteq \mathcal{X}_M^L$ be all sets $\mathcal{S} \in \mathcal{X}_M^L$ that contain an u-deletion-correcting code $\mathcal{Y} \subseteq \mathcal{S}$ of size $|\mathcal{Y}| > M - y(M)$ and let $\mathcal{X} = \mathcal{X}_{\mathsf{r}} \cap \mathcal{X}_{\mathsf{e}}$. The remaining sets are comprised in $\mathcal{X}^{\mathsf{c}} = \mathcal{X}_M^L \setminus \mathcal{X}$. Since $\mathcal{X}$ and $\mathcal{X}^{\mathsf{c}}$ partition $\mathcal{X}_M^L$, every $(s, t, u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies

$$|\mathcal{C}| = |\mathcal{C} \cap \mathcal{X}| + |\mathcal{C} \cap \mathcal{X}^{\mathsf{c}}| \leq \frac{\left|\bigcup_{\mathcal{S} \in \mathcal{X}} B(\mathcal{S})\right|}{\min_{\mathcal{S} \in \mathcal{X}} |B(\mathcal{S})|} + |\mathcal{X}^{\mathsf{c}}|.$$

The number of received sets after a loss of exactly s sequences and t sequences with exactly u deletions each is at most

$$\left|\bigcup_{\mathcal{S} \in \mathcal{X}} B(\mathcal{S})\right| \leq \binom{2^L}{M - s - t}\binom{2^{L-u}}{t},$$

as each received set consists of $M - s - t$ error-free sequences and t sequences of length $L - u$. Each $\mathcal{S} \in \mathcal{X}$ contains less than $y(M)$ sequences, which do not belong to the u-deletion-correcting code $\mathcal{Y}$ and less than $y(M)$ (possibly different) sequences with $||\boldsymbol{x}|| < L/2 - \rho(L)$. Thus, at least $M - 2y(M)$ sequences form an u-deletion-correcting code and satisfy $||\boldsymbol{x}|| \geq L/2 - \rho(L)$ and by Lemma 3.26, we have

$$|B(\mathcal{S})| \geq \binom{M - 2y(M)}{s}\binom{M - 2y(M) - s}{t}\binom{L/2 - \rho(L) - u}{u}^t$$

for each $\mathcal{S} \in \mathcal{X}$. The number of remaining sets $\mathcal{S} \notin \mathcal{X}$ is $|\mathcal{X}^{\mathsf{c}}| = |\mathcal{X}_M^L \setminus \mathcal{X}| \leq |\mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{r}}| + |\mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{e}}|$. Each such set in $\mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{r}}$ contains at least $y(M)$ sequences with $||\boldsymbol{x}|| < L/2 - \rho(L)$ and each set in $\mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{e}}$ does not contain an u-deletion-correcting code of size more than $M - y(M)$. By Lemma 3.25, we have that

$$|\mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{r}}| \leq \binom{2^L}{M - y(M)} \binom{2^L/L^2}{y(M)},$$

as each $\mathcal{S} \in \mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{r}}$ can be constructed by choosing $y(M)$ sequences to have less than $L/2 - \rho(L)$ runs and the remaining sequences are chosen arbitrarily. Further, using Lemma 3.22, it follows that

$$|\mathcal{X}_M^L \setminus \mathcal{X}_{\mathsf{e}}| \leq \binom{2^L}{M - y(M)} \binom{(M - y(M))V^{\mathbb{D}}(u)}{y(M)},$$

where $V^{\mathbb{D}}(u) = \max_{\boldsymbol{x} \in \Sigma_2^L} |\{\boldsymbol{y} \in \Sigma_2^L : B^{\mathbb{D}}(\boldsymbol{x}, u)(\boldsymbol{x}) \cap B^{\mathbb{D}}(\boldsymbol{y}, u) \neq \emptyset\}|$. It follows that the size of any $(s,t,u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is at most

$$\begin{aligned}
|\mathcal{C}| \leq & \frac{\binom{2^L}{M-s-t}\binom{2^{L-u}}{t}}{\binom{M-2y(M)}{s}\binom{M-2y(M)-s}{t}\binom{L/2-\rho(L)-u}{u}^t} + \binom{2^L}{M - y(M)} \left(\binom{2^L/L^2}{y(M)} + \binom{(M - y(M))V^{\mathbb{D}}(u)}{y(M)} \right) \\
= & \frac{\binom{2^L}{M-s-t}\binom{2^{L-u}}{t}}{\binom{M-2y(M)}{s}\binom{M-2y(M)-s}{t}\binom{L/2-\rho(L)-u}{u}^t} (1 + \Delta_{\mathsf{r}} + \Delta_{\mathsf{c}}),
\end{aligned}$$

where Δ_{r} and Δ_{c} are defined implicitly by the above equation. We will show now that $\Delta_{\mathsf{r}} \to 0$ for $M \to \infty$. First, by an application of Lemma A.2 we find that $\log \binom{M-2y(M)}{s} = O(L)$, $\log \binom{M-2y(M)-s}{t} = O(L)$, $\log \binom{2^{L-u}}{t} = O(L)$, and $\log \binom{L/2-\rho(L)-u}{u}^t = O(\log L)$. It follows that

$$\log \Delta_{\mathsf{r}} = \log \frac{\binom{2^L/L^2}{y(M)}\binom{2^L}{M-y(M)}}{\binom{2^L}{M-s-t}} + O(L).$$

An application of Lemma A.6 with $z(L) = L^2$ then implies that

$$\log \Delta_{\mathsf{r}} \leq -\frac{M}{\log M} \log \log M + O\left(\frac{M}{\log M}\right).$$

Hence, $\Delta_{\mathsf{r}} \to 0$ for $M \to \infty$. We now turn to Δ_{c}. First, note that $V^{\mathbb{D}}(u) \leq \binom{L}{u} S^{\mathbb{I}}(L-u, u)$, which can be explained as follows. For any $\boldsymbol{x} \in \Sigma_2^L$ an arbitrary $\boldsymbol{y} \in \Sigma_2^L$ with $B^{\mathbb{D}}(\boldsymbol{x}, u) \cap B^{\mathbb{D}}(\boldsymbol{y}, u) \neq \emptyset$ can be constructed from $\boldsymbol{x}$ by removing u symbols from $\boldsymbol{x}$ and inserting u symbols in the result. Together with the trivial fact that $S^{\mathbb{D}}(\boldsymbol{x}, u) \leq \binom{L}{u}$, the statement follows. Analogous to the proof of Theorem 3.23, it can then be shown that $\Delta_{\mathsf{c}} \to 0$ for $M \to \infty$. We obtain for the maximum size of a $(s,t,u)_{\mathbb{D}}$-correcting code,

$$|\mathcal{C}| \leq \frac{\binom{2^L}{M-s-t}\binom{2^{L-u}}{t}}{\binom{M}{s}\binom{M-s}{t}\binom{L/2}{u}^t}(1 + o(1)).$$

The redundancy is consequently at least

$$r(\mathcal{C}) = \log \binom{2^L}{M} - \log |\mathcal{C}| \geq \log \frac{\binom{2^L}{M}\binom{M}{s}\binom{M-s}{t}\binom{L/2}{u}^t}{\binom{2^L}{M-s-t}\binom{2^{L-u}}{t}} + o(1) = sL + tu \log L - \log(s!u!^t) + o(1).$$

□

The result of Theorem 3.27 is particularly interesting when compared with Theorem 3.23, which depicts the case of substitution errors inside the sequences. It can be seen that correcting substitutions requires $t \log M - \log t!$ more bits of redundancy as compared to insertion or deletion errors only. While this seems surprising, there is a practical reason for this phenomena. For the case of only insertion or only deletion errors, it is directly possible to identify erroneous sequences, by checking their length to be different from L. This is not possible for substitution errors, and erroneous sequences can be confused with correct sequences, which means that additional redundancy is required for detecting the erroneous sequences. In fact, we will show in Construction 3.50, how to constructively exploit the identification of erroneous sequences for the case of $(0,1,1)_{\mathbb{D}}$ deletion errors and obtain a code that asymptotically achieves the bound from Theorem 3.27. In the following we derive a sphere packing bound for the case when the number of erroneous sequences scales with M.

Theorem 3.28. *For fixed $s, u \in \mathbb{N}_0$ and fixed $0 < \beta < 1$, any $(s, M-s, u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ satisfies*

$$r(\mathcal{C}) \geq Mu \log L + O(M),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. The proof is similar to that of Theorem 3.27 and we use the same notation of received words $B(\mathcal{S}) \triangleq \{\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, s, M-s, u) : |\mathcal{S}' \cap \Sigma_2^L| = 0, |\mathcal{S}' \cap \Sigma_2^{L-u}| = M - s\}$ with no sequences of length L and with $M-s$ sequences of length $L-u$. Further, we adopt the notation $\mathcal{X} = \mathcal{X}_{\mathsf{r}} \cap \mathcal{X}_{\mathsf{e}}$ for sets that contain an u-deletion-correcting code of size $|\mathcal{Y}| > M - y(M)$ and more than $M - y(M)$ sequences with at least $||\boldsymbol{x}|| \geq L/2 - \rho(L)$ runs, where $y(M) = M/\log\log M$ and $\rho(L) = \sqrt{L/2 \ln L \log^2 L}$. With Lemma 3.26, it follows

$$|B(\mathcal{S})| \geq \binom{M - 2y(M)}{s} \binom{L/2 - \rho(L) - u}{u}^{M-2y(M)-s}$$

for all $\mathcal{S} \in \mathcal{X}$. By the same sphere packing argument as in Theorem 3.27, it follows that the size of any $(s,t,u)_{\mathbb{D}}$-correcting code $\mathcal{C} \subseteq \mathcal{X}_M^L$ is at most

$$|\mathcal{C}| \leq \frac{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}}{\binom{M-2y(M)}{s} \binom{L/2-\rho(L)-u}{u}^{M-2y(M)-s}} (1 + \Delta_{\mathsf{r}} + \Delta_{\mathsf{c}}),$$

where Δ_{r} and Δ_{c} are given by

$$\Delta_{\mathsf{r}} = \frac{\binom{M-2y(M)}{s} \binom{L/2-\rho(L)-u}{u}^{M-y(M)-s} \binom{2^L/L^{\log^2 L}}{y(M)} \binom{2^L}{M-y(M)}}{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}},$$

$$\Delta_{\mathsf{c}} = \frac{\binom{M-2y(M)}{s} \binom{L/2-\rho(L)-u}{u}^{M-y(M)-s} \binom{M-y(M)V^{\mathbb{D}}(u)}{y(M)} \binom{2^L}{M-y(M)}}{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}}.$$

We will show now that $\Delta_{\mathsf{r}} \to 0$ and $\Delta_{\mathsf{c}} \to 0$ for $M \to \infty$. By Lemma A.2 it is immediate that

$\log \binom{M-2y(M)}{s} = O(M)$ and we obtain

$$\log \Delta_{\mathsf{r}} \leq \log \frac{\binom{L/2}{u}^M \binom{2^L/L^{\log^2 L}}{y(M)} \binom{2^L}{M-y(M)}}{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}} + O(M) = \log \frac{\binom{2^L/L^{\log^2 L}}{y(M)} \binom{2^L}{M-y(M)}}{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}} + Mu \log L + O(M)$$
$$\overset{(a)}{\leq} \log \frac{\binom{2^L/L^{\log^2 L}}{y(M)} \binom{2^L}{M-y(M)}}{\binom{2^L}{M-s}} + Mu \log L + O(M) \overset{(b)}{\leq} -\frac{M \log^3 L}{\log(\beta L)} + O(M \log L)$$

where for inequality (a) we used that

$$\log \frac{\binom{2^L}{M-s}}{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}} = O(M)$$

and applied Lemma A.6 in inequality (b). Hence, $\Delta_{\mathsf{r}} \to 0$ for $M \to \infty$. Analogous to the proof of Theorem 3.24, it can be shown that $\Delta_{\mathsf{c}} \to 0$ for $M \to \infty$. We obtain for the maximum size of a $(s, M-s, u)_{\mathbb{D}}$-correcting code

$$|\mathcal{C}| \leq \frac{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}}{\binom{M}{s} \binom{L/2-\rho(L)-u}{u}^{M-y(M)-s}} (1 + o(1)).$$

The redundancy is consequently at least

$$r(\mathcal{C}) \geq \log \frac{\binom{2^L}{M} \binom{M}{s} \binom{L/2-\rho(L)-u}{u}^{M-y(M)-s}}{\binom{2^L}{2y(M)} \binom{2^{L-u}}{M-2y(M)-s}} + o(1) \geq Mu \log L + O(M).$$

□

3.5 Code Constructions over Sets of DNA Sequences

In this section we will present several different code constructions that will be suitable for different error types and for different parameter regimes. First, in Sections 3.5.1 and 3.5.2 we start by presenting constructions that use indices to combat the loss of ordering and then employing an outer maximum-distance-separable (MDS) code whose symbols are the DNA sequences. Such schemes are particularly suitable for the case when there are many errors per sequence, as, independently of the number of errors occurring in one sequence, it only accounts for a single error inside the MDS code. We proceed by constructing codes using a binary constant-weight representation of sets in Section 3.5.3. This construction comes at the advantage of improved redundancy due to avoiding explicit indexing at the cost of increased complexity. We then proceed with concatenated and tensor-product code constructions, achieving optimal (up to lower order terms) redundancy in Sections 3.5.4 and 3.5.5. A summary of the redundancy achieved by the code constructions in this section can be found in Table 3.2.

Table 3.2: Redundancies achieved by the construction presented in this chapter. Low order terms are omitted.

Error corr.	Construction		Sphere packing bound	
$(s,t,\bullet)_{\mathbb{L}}$	$M\log \mathrm{e} + (s+2t)(L-\log M)$	[Const. 3.30]	$(s+t)L + t\log M$	[Cor. 3.18]
	$\frac{(1-c)}{2}M^c \log M$ $+(s+2t)M^{1-c}(L-\log M)$	[Const. 3.33]		
	$(s+2t)L$	[Const. 3.38]		
$(0,M,1)_{\mathbb{D}}$	$M\log L$	[Const. 3.44]	$M\log L$	[Thm. 3.28]
$(0,M,u)_{\mathbb{S}}$	$Mu\log L$	[Const. 3.47]	$Mu\log L$	[Thm. 3.24]
$(0,1,1)_{\mathbb{D}}$	$\log L$	[Const. 3.50]	$\log L$	[Thm. 3.27]

3.5.1 Index-Based Construction using MDS Codes

We start by presenting code constructions that use an MDS code[5] of length M over the sequences. That is, each sequence $\boldsymbol{x}_i$ is a symbol of the MDS codeword. In order to restore the ordering information of the sequences, or, equivalently, the MDS codeword symbols at the receiver, we prepend an index to each sequence that holds the position of the sequence in the codeword.

The following function, which collects all indices of a set of sequences, will be useful for our constructions that are based on indexing.

Definition 3.29. *For any set $\mathcal{S} \subseteq \Sigma_2^L$ we define*

$$\boldsymbol{I}(\mathcal{S}) = \bigcup_{\boldsymbol{x}\in\mathcal{S}} \{\mathsf{pref}_{\lceil \log M\rceil}(\boldsymbol{x})\}$$

to be the set of indices of the sequences in $\mathcal{A}$.

The following construction is based on adding an index in front of all sequences $\boldsymbol{x}_i$ and using a Reed-Solomon code, or more generally an MDS code, over the M sequences for error correction. For all M and k, where $k \leq M$ we denote by $\mathsf{MDS}[M,k]$ an MDS code over some field of size at least $M-1$. We do not explicitly specify the field over which the code is defined as it will be clear from the context.

In the following Construction 3.30, the sequences $\boldsymbol{x}_i = (\boldsymbol{I}(i), \boldsymbol{u}_i)$ of each codeword set are constructed by writing a binary representation of the index, $\boldsymbol{I}(i)$, of length $\lceil \log M\rceil$ in the first

[5] While the construction works in principle with any MDS code, in practice one would employ a Reed-Solomon code for which many different efficient decoding algorithms are known. For a background on MDS and in particular Reed-Solomon codes, we refer the reader to [Rot06, Ch. 5 and 11].

part of each sequence. Then, the remaining part $\boldsymbol{u}_i$ is viewed as a symbol over the finite field $\mathbb{F}_{2^{L-\lceil \log M \rceil}}$, and $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_M)$ will form a codeword in a MDS code.[6]

Construction 3.30. *For all M, L, and a positive integer δ, let $\mathcal{C}_1(M, L, \delta)$ be the code defined by*

$$\mathcal{C}_{\mathsf{MDS}}(M, L, \delta) = \{\mathcal{S} \in \mathcal{X}_M^L : \boldsymbol{x}_i = (\boldsymbol{I}(i), \boldsymbol{u}_i), (\boldsymbol{u}_1, \dots, \boldsymbol{u}_M) \in \mathsf{MDS}[M, M - \delta]\}.$$

This code provides a direct construction to correct a loss of sequences and erroneous sequences with an arbitrary amount of errors each. The error correction capability for several types of errors is summarized in the following statement.

Proposition 3.31. *For all M, L, δ, the code $\mathcal{C}_{\mathsf{MDS}}(M, L, \delta)$ is*

- *$(s, t, \bullet)_{\mathbb{L}}$-correcting for all $s + 2t \leq \delta$,*
- *$(s, t, \bullet)_{\mathbb{I}}$-correcting for all $s + t \leq \delta$,*
- *$(s, t, \bullet)_{\mathbb{D}}$-correcting for all $s + t \leq \delta$.*

Proof. Let $\mathcal{S} \in \mathcal{C}_{\mathsf{MDS}}(M, L, \delta)$ denote the transmitted set and $\mathcal{S}'$ be the received set after a loss of sequences and errors. We will prove the lemma by presenting a decoding algorithm and start with proving the lemma for the case of arbitrary edit errors. According to Definition 3.3, we write $\mathcal{S}_{\mathsf{C}}, \mathcal{S}_{\mathsf{L}}, \mathcal{S}_{\mathsf{E}}$ as the sets of error-free, lost, and erroneous sequences, and $\mathcal{S}'_{\mathsf{E}}$ are the erroneous outcomes of the sequences in $\mathcal{S}_{\mathsf{E}}$. First, we observe that if we can recover the MDS codeword $(\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_M)$, we can also recover $\mathcal{S}$ by prepending the index $\boldsymbol{I}(i)$ in front of each $\boldsymbol{u}_i$. Given $\mathcal{S}'$, the decoder creates the estimate $(\boldsymbol{u}'_1, \boldsymbol{u}'_2, \dots, \boldsymbol{u}'_M)$ by declaring all positions i with

$$|\{\boldsymbol{x}' \in \mathcal{S}' : \mathsf{pref}_{\lceil \log M \rceil}(\boldsymbol{x}') = \boldsymbol{I}(i)\}| \neq 1,$$

i.e., for which there is not exactly one index in $\mathcal{S}'$, as erasures. For the remaining positions i there exists exactly one $\boldsymbol{x}' \in \mathcal{S}'$ with $\boldsymbol{x}' = (\boldsymbol{I}(i), \boldsymbol{u}'_i)$ and the corresponding symbols $\boldsymbol{u}'_i$ are filled as symbols into the MDS codeword. We will show that the number of erasures s' and the number of errors t' in $(\boldsymbol{u}'_1, \boldsymbol{u}'_2, \dots, \boldsymbol{u}'_M)$ satisfy $s' + 2t' \leq \delta$ by the following consideration. Intuitively, each lost sequence accounts for an erasure and each erroneous sequence either accounts for one error, if its erroneous index is not present in the received set yet or for at most two erasures, if its index agrees with one of the indices present in the set. We will elaborate on and rigorously prove this intuition in the following. Recall that according to Definition 3.29, $\boldsymbol{I}(\mathcal{S}_{\mathsf{C}}), \boldsymbol{I}(\mathcal{S}_{\mathsf{L}})$, and $\boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})$ are the sets of indices of the correctly received sequences, lost sequences, and erroneous sequences, respectively. Then the set of erroneous positions $\mathcal{T}'$[7] in $(\boldsymbol{u}'_1, \boldsymbol{u}'_2, \dots, \boldsymbol{u}'_M)$ is a subset $\mathcal{T}' \subseteq \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}}) \setminus \boldsymbol{I}(\mathcal{S}_{\mathsf{C}})$, as the erroneous sequences that have an index in $\boldsymbol{I}(\mathcal{S}_{\mathsf{C}})$ result in erasures. Further, the set of positions with erasures $\mathcal{E}'$ is a subset $\mathcal{E}' \subseteq \boldsymbol{I}(\mathcal{S}_{\mathsf{L}}) \cup \boldsymbol{I}(\mathcal{S}_{\mathsf{E}}) \cup \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})$. Using that $\mathcal{T}'$ and $\mathcal{E}'$ have to be distinct by definition, it follows that

$$s' + 2t' = |\mathcal{E}'| + 2|\mathcal{T}'| = |\mathcal{E}' \cup \mathcal{T}'| + |\mathcal{T}'| \leq |\boldsymbol{I}(\mathcal{S}_{\mathsf{L}}) \cup \boldsymbol{I}(\mathcal{S}_{\mathsf{E}}) \cup \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})| + |\boldsymbol{I}(\mathcal{S}'_{\mathsf{E}}) \setminus \boldsymbol{I}(\mathcal{S}_{\mathsf{C}})|.$$

[6] We assume that $M \leq \sqrt{2^L}$ or, equivalently, $\beta \leq \frac{1}{2}$, in this section to guarantee the existence of the MDS code [Rot06, Ch. 11]. However, the case $M > \sqrt{2^L}$ can always be realized by employing non-MDS codes.

[7] Although the elements in $\mathcal{T}'$ are binary vectors of length at most $\lceil \log M \rceil$, we will treat them as their corresponding decimal integer numbers, according to the mapping $\boldsymbol{I}(i)$, in the sequel.

We now make use of the fact that $\mathcal{S}_{\mathsf{E}}$ and $\mathcal{S}_{\mathsf{L}}$ are distinct by definition and we arrive at

$$\begin{aligned} s' + 2t' &\leq |\boldsymbol{I}(\mathcal{S}_{\mathsf{L}})| + |\boldsymbol{I}(\mathcal{S}_{\mathsf{E}})| + |\boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})| - |\boldsymbol{I}(\mathcal{S}_{\mathsf{E}}) \cap \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})| - |\boldsymbol{I}(\mathcal{S}_{\mathsf{L}}) \cap \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})| + |\boldsymbol{I}(\mathcal{S}'_{\mathsf{E}}) \setminus \boldsymbol{I}(\mathcal{S}_{\mathsf{C}})| \\ &= s + 2t - |\boldsymbol{I}(\mathcal{S}_{\mathsf{E}}) \cap \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})| - |\boldsymbol{I}(\mathcal{S}_{\mathsf{L}}) \cap \boldsymbol{I}(\mathcal{S}'_{\mathsf{E}})| + |\boldsymbol{I}(\mathcal{S}'_{\mathsf{E}}) \cap (\boldsymbol{I}(\mathcal{S}_{\mathsf{L}}) \cup \boldsymbol{I}(\mathcal{S}_{\mathsf{E}}))| = s + 2t \leq \delta. \end{aligned}$$

The correction capability then follows from employing any standard unique erasure-error decoding algorithm [LC04, Ch. 7.7] on the estimate $(\boldsymbol{u}'_1, \boldsymbol{u}'_2, \ldots, \boldsymbol{u}'_M)$.

For the case of only insertion ($\mathbb{I}$) and only deletion ($\mathbb{D}$) errors, it is possible to identify the erroneous sequences by checking their length to be larger (respectively smaller) than L. If these sequences are discarded, there are in total $s + t$ erasures inside the MDS codeword, which can be corrected, if $s + t \leq \delta$. □

For the practically important case of a loss of sequences and combinations of substitution and deletion errors, $\mathcal{C}_{\mathsf{MDS}}(M, L, \delta)$ can correct all errors, if $s + 2t_{\mathbb{S}} + t_{\mathbb{D}} \leq \delta$, where $t_{\mathbb{S}}$ is the number of sequences suffering from substitution errors only and $t_{\mathbb{D}}$ is the number of sequences with deletion errors. The same also holds for combinations of substitution and insertion errors. However, this is not true for combinations of substitutions, insertions *and* deletions as a sequence that contains insertions and deletions might have length exactly L and therefore cannot be erased. In this case, as elaborated in the proof, $s + 2t \leq \delta$ has to hold. More generally, the MDS codeword requires two redundancy symbols to correct erroneous sequences, which have length exactly L, and requires only a single redundancy symbol for sequences, which have a length that is different from L, as these can be detected as erroneous and thus they can be erased.

The redundancy of Construction 3.30 is stated in the following theorem.

Theorem 3.32. *For all M, L, δ, the redundancy of the code $\mathcal{C}_{\mathsf{MDS}}(M, L, \delta)$ is*

$$r(\mathcal{C}_{\mathsf{MDS}}(M, L, \delta)) = r(\mathcal{I}_M^L) + \delta(L - \lceil \log M \rceil).$$

Proof. First, indexing the sequences requires a redundancy of $r(\mathcal{I}_M^L)$, which is derived in Lemma 4.3 below. Second, the MDS code has δ redundant symbols over a field of size $2^{L - \lceil \log M \rceil}$, which corresponds to a total of $\delta(L - \lceil \log M \rceil)$ additional redundancy bits. □

While the redundancy of Construction 3.30 can be large, especially when $M \gg L$, it provides some very useful features. First, it is possible to efficiently encode and decode this code using standard encoders and decoders for MDS codes. Second, it is not necessary to design the code for a specific number of errors s and t, but rather their sum $s + 2t$, which allows for a flexible decoding procedure.

3.5.2 Construction with Shortened Indices and MDS Codes

Construction 3.30, presented in the previous section, uses indexing and is beneficial for its simplicity in the encoding and decoding procedures and flexibility in the types of errors that occur. However, especially for small δ the redundancy is away from the sphere-packing lower bound in Corollary 3.18 by a term that scales linearly in M. This is due to the use of an index, which has by itself a redundancy of roughly $M \log \mathrm{e}$ as shown in Lemma 4.3. We therefore will now propose a construction that uses less bits for indexing and thus allocates multiple sequences with the same index. This reduces the redundancy required for indexing and allows a trade-off in redundancy with respect to L and M, as we will show in the following. To start with, we

define the shortened indices $\boldsymbol{I}_c(i) \in \Sigma_2^{c \log M}$ to be the binary representation of $\lfloor \frac{i-1}{2^{(1-c)M}} \rfloor$, which has length $c \log M$. Note that by this definition $\boldsymbol{I}_1(i) = \boldsymbol{I}(i)$. To simplify notation, we assume in the sequel that $c \log M$ is integer.

Construction 3.33. *For $1 \leq i \leq M^c$, abbreviate by the set of distinct sequences with the same index $\boldsymbol{I}_c(i)$, where $\boldsymbol{u}_j \in \Sigma_2^{L - c \log M}$. For $\delta \geq 0$, let $\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)$ be the code defined by*

$$\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta) = \{\mathcal{S} \in \mathcal{X}_M^L : \boldsymbol{x}_i = (\boldsymbol{I}_c(i), \boldsymbol{u}_i), (\mathcal{U}_1, \ldots, \mathcal{U}_{M^c}) \in \mathsf{MDS}[M^c, M^c - \delta]^{8}, \\ \mathcal{U}_i = \{\boldsymbol{u}_{(i-1)M^{1-c}+1}, \ldots, \boldsymbol{u}_{iM^{1-c}}\}\}.$$

To guarantee existence of the MDS code, we require $M^c \leq \binom{2^L M^{-c}}{M^{1-c}}$. For $M = 2^{\beta L}$, for example $c \leq 1 + (\log \frac{1-\beta}{\beta})/(\beta L)$ is sufficient.

Note that $\mathcal{U}_i = \{\boldsymbol{u}_{(i-1)M^{1-c}+1}, \ldots, \boldsymbol{u}_{iM^{1-c}}\}$ comprises all vectors that share the same shortened index. In total, there are M^c groups of sequences which use the same index and each group contains M^{1-c} sequences. For $c = 1$ this construction is equal to the one presented in the previous section. The error-correction capability is summarized in the following proposition.

Proposition 3.34. *For all M, L, δ, the code $\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)$ is*

- *$(s, t, \bullet)_{\mathbb{L}}$-correcting for all $s + 2t \leq \delta$,*
- *$(s, t, \bullet)_{\mathbb{I}}$-correcting for all $s + t \leq \delta$,*
- *$(s, t, \bullet)_{\mathbb{D}}$-correcting for all $s + t \leq \delta$.*

Proof. The proof follows the same idea as that for Proposition 3.31. We will show that the MDS codeword $\mathcal{U} = (\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_{M^c})$ can be recovered from $\mathcal{U}' = (\mathcal{U}'_1, \mathcal{U}'_2, \ldots, \mathcal{U}'_{M^c})$, where $\mathcal{U}'_i = \{\mathsf{suff}_{L - c \log M}(\boldsymbol{x}') : \boldsymbol{x}' \in \mathcal{S}', \mathsf{pref}_{c \log M}(\boldsymbol{x}') = \boldsymbol{I}_c(i)\}$ collects all sequences in $\mathcal{S}'$ which have the same index i. Given $\mathcal{S}'$, we create the received estimate word $\mathcal{U}'$ by declaring all positions i with $|\mathcal{U}'_i| \neq M^{1-c}$ as erasures. The remaining positions in $\mathcal{U}'$ are filled with the corresponding symbols $\mathcal{U}'_i$. Proving that the number of erasures s' and the number of errors t' in $\mathcal{U}'$ satisfy $s' + 2t' \leq \delta$ is completely analogous to Proposition 3.31 and is omitted for brevity.

For the case of only insertion ($\mathbb{I}$) and only deletion ($\mathbb{D}$) errors, it is possible to identify the erroneous groups by checking the length of the respective sequences to be larger (respectively smaller) than L. If these sequences are discarded, which results in erasures in the corresponding positions, there are in total $s + t$ erasures inside the MDS codeword. These can be corrected by a standard erasure correction decoding algorithm, if $s + t \leq \delta$. □

The redundancy of Construction 3.33 is stated in the following theorem.

Theorem 3.35. *The redundancy of Construction 3.33 is given by*

$$r(\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)) = \log \binom{2^L}{M} - (M^c - \delta) \log \binom{2^L M^{-c}}{M^{1-c}}.$$

[8] We treat each set $\mathcal{U}_i$ as an element of a field of size $\binom{2^L M^{-c}}{M^{1-c}}$ and for simplicity we assume that this number is a prime power such that the finite field exists. For the case when this size is not exactly a prime power, it is possible to use a slightly smaller field and restrict the choices for the sequences inside $\mathcal{U}_i$ accordingly, causing only a negligible loss in redundancy.

For fixed $0 < c < 1$, $\delta \in \mathbb{N}_0$ *and* $0 < \beta < 1$, *the redundancy of* $\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)$ *is asymptotically*

$$r(\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)) = \frac{(1-c)}{2} M^c \log M + \frac{\log 2\pi}{2} M^c + \delta M^{1-c} (L - \log M + \log \mathrm{e}) + o(M^c + M^{1-c}),$$

when $M \to \infty$ *with* $M = 2^{\beta L}$.

Proof. The cardinality of Construction 3.33 can be computed as follows. Each group $\mathcal{U}_i$ consists of M^{1-c} unordered, distinct sequences, which share the same index $\boldsymbol{I}_c(i)$. In total, there are $M^c - \delta$ information groups, since δ groups are redundancy symbols of the MDS codeword. Therefore, the redundancy is

$$r(\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)) = \log \binom{2^L}{M} - \log \binom{2^L M^{-c}}{M^{1-c}}^{M^c - \delta}.$$

Applying Stirling's approximation [Rob55] onto the binomial coefficients yields

$$\begin{aligned} r(\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)) &= \log \binom{2^L}{M} - (M^c - \delta) \log \binom{2^L M^{-c}}{M^{1-c}} \\ &= \frac{1-c}{2} M^c \log M + \frac{M^c - 1}{2} \log \left(1 - \frac{M}{2^L}\right) - \gamma_2 M^c + \gamma_1 \\ &\quad + \delta \left(M^{1-c} L - M^{1-c} \log M - \frac{1-c}{2} \log M - \left(2^L M^{-c} - M^{1-c} + \frac{1}{2}\right) \log \left(1 - \frac{M}{2^L}\right) + \gamma_2 \right), \end{aligned}$$

where $\gamma_1 = -\log \sqrt{2\pi} + o(1)$ and $\gamma_2 = -\log \sqrt{2\pi} + o(1)$, when $c < 1$. Note that it can be verified that for $c = 1$, γ_2 has a different asymptotic behavior, i.e., $\gamma_2 = -\log \mathrm{e} + o(1)$. Therefore, for $c = 1$, the expression for $r(\mathcal{C}_3(M, L, c, \delta))$ yields the same redundancy as in Theorem 3.32. Employing Lemma A.1 onto the two logarithmic terms yields

$$r(\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)) = \frac{1-c}{2} M^c \log M + \frac{\log 2\pi}{2} M^c + \delta M^{1-c} (L - \log M + \log \mathrm{e}) + o(M^c + M^{1-c}).$$

□

Note that the last summand in the asymptotic expression for $\mathcal{C}_{\mathsf{SIMDS}}(M, L, c, \delta)$ in Theorem 3.35 quantifies the redundancy from the MDS construction, since it is multiplied by δ, the number of redundant symbols of the MDS code. The two remaining terms therefore quantify the redundancy required for indexing. This shows that, asymptotically, for $c > 0.5$ the redundancy needed for indexing dominates, as the terms for indexing scale as M^c and the term for the MDS construction scales as M^{1-c} and for $c < 0.5$ the redundancy from the MDS construction dominates the redundancy of the overall construction.

3.5.3 Construction of Set Codes with Constant-Weight Codes

We continue with a construction that uses the fact that a set can be represented by a binary indicator vector. Employing constant-weight error-correcting codes, we will then construct a code that efficiently corrects errors within the DNA sequences. To this end, we impose an ordering

(e.g., lexicographic) onto the sequences in Σ_2^L. Thus, every data set $\mathcal{S} \in \mathcal{X}_M^L$ can be represented by a binary vector $\boldsymbol{v}(\mathcal{S})$ of length 2^L, where each non-zero entry in $\boldsymbol{v}(\mathcal{S})$ indicates that a specific sequence is contained in the set $\mathcal{S}$. The set of possible data sets $\mathcal{X}_M^L$ can therefore be represented[9] by constant-weight binary vectors of length 2^L

$$\mathcal{V}_M^L = \{\boldsymbol{v} \in \{0,1\}^{2^L} : \mathsf{wt_H}(\boldsymbol{v}) = M\},$$

where $\mathsf{wt_H}(\boldsymbol{v})$ denotes the *Hamming weight* of $\boldsymbol{v}$, i.e., the number of non-zero entries inside the vector $\boldsymbol{v}$. That is, the mapping $\boldsymbol{v}$ defines a bijection between $\mathcal{X}_M^L$ and $\mathcal{V}_M^L$ and thus $\boldsymbol{v}^{-1}$ is well-defined. Using this representation, a loss of a sequence $\boldsymbol{x} \in \mathcal{S}$ corresponds to an asymmetric $1 \to 0$ error inside $\boldsymbol{v}(\mathcal{S})$ at the position corresponding to $\boldsymbol{x}$. Substitution errors inside a sequence $\boldsymbol{x} \in \mathcal{S}$ translate a single $1 \to 0$ at the corresponding position of the original sequence $\boldsymbol{x}$, and a single $0 \to 1$ error at the position of its erroneous outcome $\boldsymbol{x}'$. In case, the erroneous outcome $\boldsymbol{x}'$ is already present in $\mathcal{S}'$, the $0 \to 1$ error is omitted and there is only a single asymmetric $1 \to 0$ error at the position of the original sequence $\boldsymbol{x}$, similar to a loss of a sequence. Note that the combination of a $1 \to 0$ and $0 \to 1$ error is sometimes called an error in the *Johnson graph*. For codes in the Johnson graph, the reader is referred to, e.g., [Bro+90; Joh72]. We start by presenting an example about this new representation.

Example 3.36. *Consider the following $M = 3$ stored sequences $\mathcal{S} = \{(001), (010), (110)\}$, each of length $L = 3$. We choose $\boldsymbol{v}(\mathcal{S})$ to map each sequence $\boldsymbol{x} \in \mathcal{S}$ to its decimal equivalent by standard base conversion and let $\boldsymbol{v}(\mathcal{S})$ be non-zero at exactly these indices. Hence, e.g., the sequence (110) is mapped to $1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6$ and thus $\boldsymbol{v}(\mathcal{S})$ will be non-zero at index 7. Note that we additionally add 1, since we index vectors starting by 1. Therefore, $\boldsymbol{v}(\mathcal{S}) = (01100010)$. Assume now, the set $\mathcal{S}$ is transmitted over a $(1,1,2)_{\mathbb{S}}$ channel, resulting in $\mathcal{S}' = \{(001), (111)\}$, where the sequence (110) was lost and the sequence (010) has been perturbed by two substitution errors. The corresponding binary representation is $\boldsymbol{v}(\mathcal{S}') = (01000001)$, where there was a single $1 \to 0$ at position 7 due to the loss of the sequence (110) and $1 \to 0$ and $0 \to 1$ errors at positions 3, respectively 8, since the sequence (010) was distorted to the sequence (111).*

With this principle in mind, we define a code that can correct asymmetric errors and errors in the Johnson graph.

Definition 3.37. *For all M, L and positive integers s, t, we define $\mathcal{V}_{\mathsf{CW}}(M, L, s, t) \subseteq \mathcal{V}_M^L$ to be a code of length 2^L that consists of codewords with constant Hamming weight M, which corrects s asymmetric $1 \to 0$ errors and t errors in the Johnson graph.*

With such a code $\mathcal{V}_{\mathsf{CW}}(M, L, s, t) \subseteq \mathcal{V}_M^L$ at hand that can correct asymmetric errors and errors in the Johnson graph, we can construct a code for the DNA storage channel.

Construction 3.38. *For all $M, L, s,$ and t we define the following code*

$$\mathcal{C}_{\mathsf{CW}}(M, L, s, t) = \{\mathcal{S} \in \mathcal{X}_M^L : \boldsymbol{v}(\mathcal{S}) \in \mathcal{V}_{\mathsf{CW}}(M, L, s, t)\}.$$

By this construction, given a constant-weight code $\mathcal{V}_{\mathsf{CW}}(M, L, s, t)$, we construct the DNA storage code $\mathcal{C}_2(M, L, s, t)$ by mapping each $\boldsymbol{c} \in \mathcal{V}_{\mathsf{CW}}(M, L, s, t)$ to its corresponding set $\mathcal{S} = \boldsymbol{v}^{-1}(\boldsymbol{c})$. Note that this mapping can be efficiently implemented, by, e.g., a decimal to binary mapping of the non-zero positions in $\boldsymbol{c}$, as illustrated in Example 3.36. The correctness of the construction is established in the following statement.

[9] A similar representation has been used as a proof technique in [Hec+17].

Proposition 3.39. *For all M, L and positive integers s, t, the code $\mathcal{C}_{\mathsf{CW}}(M, L, s, t)$ is an $(s, t, \bullet)_{\mathbb{L}}$-correcting code.*

Proof. Denote by $\mathcal{S}'$ the received set after a loss of at most s sequences and errors in at most t sequences. Let s' be the number of asymmetric errors and t' be the number of errors in the Johnson graph that occurred in $\boldsymbol{v}(\mathcal{S})$. Clearly, $s' + t' \leq s + t$ and $t' \leq t$. Note that $s' = M - \mathsf{wt}_{\mathsf{H}}(\boldsymbol{v}(\mathcal{S}'))$ is detectable by the decoder. If $s' \leq s$, then the decoder can directly decode the loss of $s' \leq s$ sequences and $t' \leq t$ errors in the Johnson graph. If $s' > s$, the decoder adds $s' - s$ (arbitrarily placed) ones to $\boldsymbol{v}(\mathcal{S}')$, resulting in exactly s asymmetric errors and at most $t' + s' - s \leq t$ errors in the Johnson graph. □

To obtain a code based on Construction 3.38, we use the fact that an asymmetric error can be represented by a single substitution error and an error in the Johnson graph can be represented by two substitution errors. Having a code with an appropriate minimum Hamming distance, it is therefore possible to employ standard codes, which will be done in the following theorem.

Theorem 3.40. *There exists a construction of the code $\mathcal{C}_{\mathsf{CW}}(M, L, s, t)$ with redundancy at most*

$$r(\mathcal{C}_{\mathsf{CW}}(M, L, s, t)) \leq (s + 2t)L.$$

Proof. By Proposition 3.39, it is sufficient to find a sufficiently large M-constant-weight code which can correct $s + 2t$ substitution errors. This is since each loss in $\mathcal{S}$ causes an $1 \to 0$ asymmetric error in $\boldsymbol{v}(\mathcal{S})$ and can be represented as a single substitution error and every error in a sequence in $\mathcal{S}$ will cause at most one $1 \to 0$ and one $0 \to 1$ error in $\boldsymbol{v}(\mathcal{S})$ and thus can be represented by two substitution errors. Next, it is known, that there exists a τ-substitution-correcting binary alternant code of length 2^L and dimension $2^L - \tau L$, cf. [Rot06, Ch. 5.5]. Due to the pigeonhole principle and since the alternant code has at most $2^{\tau L}$ cosets, there is one coset of the alternant code that contains at least $\binom{2^L}{M}/2^{\tau L}$ words with constant weight M, and therefore there exists a code $\mathcal{C}_2(M, L, s, t)$ of cardinality at least $\binom{2^L}{M}/2^{\tau L}$. With this alternant code, the redundancy of Construction 3.38 is therefore at most

$$r(\mathcal{C}_{\mathsf{CW}}(M, L, s, t)) \leq \log \binom{2^L}{M} - \log \frac{\binom{2^L}{M}}{2^{(s+2t)L}} = (s + 2t)L.$$

Using $\tau = s + 2t$ yields the theorem. □

3.5.4 Concatenated Constructions

We now proceed with a discussion of concatenating codes to obtain codes for the combinatorial DNA storage channel. We further prove error-correction capabilities of the concatenation based on the properties of the two component codes. Assume we are given two codes, an outer code $\mathcal{C}_{\mathsf{o}} \subseteq \mathcal{X}_M^{L_{\mathsf{o}}}$, where $L_{\mathsf{o}} < L$, and an inner code $\mathcal{C}_{\mathsf{i}} \subseteq \Sigma_2^L$ of dimension L_{o} and length L. Notably the outer code is a code over sets and the inner code is a code over vectors. We then concatenate the two codes by encoding each sequence $\boldsymbol{x}_{\mathsf{o}} \in \mathcal{S}_{\mathsf{o}}$ of a codeword $\mathcal{S}_{\mathsf{o}} \in \mathcal{C}_{\mathsf{o}}$ of the outer code with the inner code, resulting in sequences $\boldsymbol{x} \in \Sigma_2^L$ of length L such that the resulting set $\mathcal{S} \in \mathcal{X}_M^L$. This procedure is formalized in the following construction.

Construction 3.41. *For all M, L, $L_{\mathsf{o}} < L$ and positive integers s, t, let $\mathcal{C}_{\mathsf{o}} \subseteq \mathcal{X}_M^{L_{\mathsf{o}}}$ be an outer code and $\mathcal{C}_{\mathsf{i}} \subseteq \Sigma_2^L$ be a standard block-code of dimension L_{o} and length L. Further, $\mathsf{enc}(\cdot) : \Sigma_2^{L_{\mathsf{o}}} \mapsto \Sigma_2^L$ is an encoder of the code $\mathcal{C}_{\mathsf{i}}$. We define the concatenated construction as*

$$\mathcal{C}_{\mathsf{Con}}(M, L, \mathcal{C}_{\mathsf{i}}, \mathcal{C}_{\mathsf{o}}) = \left\{ \mathcal{S} \in \mathcal{X}_M^L : \mathcal{S} = \bigcup_{\boldsymbol{x}_{\mathsf{o}} \in \mathcal{S}_{\mathsf{o}}} \mathsf{enc}(\boldsymbol{x}_{\mathsf{o}}), \mathcal{S}_{\mathsf{o}} \in \mathcal{C}_{\mathsf{o}} \right\}.$$

As outer code $\mathcal{C}_{\mathsf{o}}$ it is in principle possible to use any set code over $\mathcal{X}_M^{L_{\mathsf{o}}}$. Using the proposed Construction 3.30, 3.33, or 3.38 it is possible to enhance the inner code to additionally correct a loss of sequences. This is done as follows.

Proposition 3.42. *Let $\mathcal{C}_{\mathsf{o}} \subseteq \mathcal{X}_M^{L_{\mathsf{o}}}$ be an $(s, 0, 0)$-correcting code and $\mathcal{C}_{\mathsf{i}} \subseteq \Sigma_2^L$ be a block-code that can correct u errors of type $\mathbb{T}$. Then, $\mathcal{C}_{\mathsf{Con}}(M, L, \mathcal{C}_{\mathsf{i}}, \mathcal{C}_{\mathsf{o}})$ is a $(s, M - s, u)_{\mathbb{T}}$-correcting code.*

Proof. The proof is immediate, since the inner code can correct all errors of type $\mathbb{T}$ inside the sequences. After correcting these errors, the lost sequences can be corrected using the outer code. □

Note that such concatenated constructions are highly relevant in practice, as there might be a few sequences, which experienced more than u errors, which can then be corrected by the outer code, since Construction 3.30, 3.33, and 3.38 can correct both a loss of sequences and errors in sequences, as long as $s + 2t \leq \delta$. Such a construction has been used for example in [Gra+15], where a Reed-Solomon code has been used as inner code and an indexed Reed-Solomon code has been used as outer code.

We now present two constructions that use the trivial outer code $\mathcal{C}_{\mathsf{o}} = \mathcal{X}_M^{L_{\mathsf{o}}}$, which is a $(0, 0, 0)$-correcting code. The first construction is applicable to single insertion or deletion errors and uses the well-known Varshamov-Tenengolt's (VT) [VT65] code as inner code. The VT code is defined as all binary vectors of length L which admit the same *checksum* a, that is defined as follows.

Definition 3.43. *The Varshamov-Tenegolts checksum $s_{\mathsf{VT}}(\boldsymbol{x})$ of $\boldsymbol{x} \in \Sigma_2^L$ is defined by*

$$s_{\mathsf{VT}}(\boldsymbol{x}) = \sum_{i=1}^{L} i x_i.$$

It is well-known [Lev66] that the knowledge of $s_{\mathsf{VT}}(\boldsymbol{x})$ modulus $L+1$ is sufficient to reconstruct $\boldsymbol{x}$ from a single insertion or deletion. For a comprehensive survey on VT codes, see [Slo00]. Using this VT checksum, we propose the following construction, which we will prove to be $(0, M, 1)_{\mathbb{ID}}$-correcting. That is, the code can correct a single deletion or insertion in every sequence.

Construction 3.44. *Let $a \in \mathbb{N}_0$, with $0 \leq a \leq L$. Then,*

$$\mathcal{C}_{\mathsf{MID}}(M, L, a) = \{\mathcal{S} \in \mathcal{X}_M^L : s_{\mathsf{VT}}(\boldsymbol{x}_i) \equiv a \bmod (L+1), \forall\, 1 \leq i \leq M\}.$$

Based on this construction we can immediately prove its following correction capabilities.

Proposition 3.45. *The code $\mathcal{C}_{\mathsf{MID}}(M, L, a)$ is n $(0, M, 1)_{\mathbb{ID}}$-correcting code.*

Proof. All erroneous sequences can be detected by checking their length to be either $L+1$ or $L-1$. If a sequence is erroneous, it can be corrected by decoding in the VT code with checksum a. Note that two distinct sequences cannot have the same erroneous outcome since they are different and belong to a single-deletion-correcting code. □

By Construction 3.44, all sequences $\boldsymbol{x}_i$ have the same checksum a, which allows to correct a single insertion or a single deletion in each sequence. The redundancy of Construction 3.44 is computed in the following lemma.

Theorem 3.46. *For fixed $0 < \beta < 1$, the redundancy of the code $\mathcal{C}_{\mathsf{MID}}(M, L, 0)$ satisfies asymptotically*

$$r(\mathcal{C}_{\mathsf{MID}}(M, L, 0)) \leq M \log(L+1) + o(M),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. It is known [Mar96; Slo00] that the number of binary words $\boldsymbol{x} \in \Sigma_2^L$ that satisfy $s_{\mathsf{VT}}(\boldsymbol{x}) = 0 \bmod (L+1)$ is at least $2^L/(L+1)$. Each codeword of $\mathcal{C}_{\mathsf{MID}}(M, L, 0)$ is a subcode of size M of the VT code $\{\boldsymbol{x} \in \Sigma_2^L : s_{\mathsf{VT}}(\boldsymbol{x}) \equiv 0 \bmod (L+1)\}$. Therefore the redundancy of Construction 3.44 with $a = 0$ is at most

$$\begin{aligned} r(\mathcal{C}_{\mathsf{MID}}(M, L, 0)) &\leq \log \binom{2^L}{M} - \log \binom{\frac{2^L}{L+1}}{M} \leq ML - M \log \left(\frac{2^L}{L+1} - M \right) \\ &\overset{(a)}{\leq} M \log(L+1) + \frac{M^2 \log \mathrm{e}}{2^L/(L+1) - M}, \end{aligned}$$

where we factored out $2^L/(L+1)$ from the logarithm and used the inequality $\log(1-x) \geq -\frac{x \log \mathrm{e}}{1-x}$ for all $x < 1$ to prove inequality (a). For $M = 2^{\beta L}$, $0 < \beta < 1$ the second term is $o(M)$, which concludes the proof. □

Interestingly, as we have shown in Theorem 3.28, the redundancy of this construction is asymptotically optimal in terms of scaling with the parameters M and L. Note that the proof of Theorem 3.46 above contains a non-asymptotic expression for the redundancy.

The second construction uses a similar concept and can be used to correct u substitution errors in each sequence.

Construction 3.47. *Let $\mathcal{C}_{\mathsf{sub}}(L, u) \subseteq \Sigma_2^L$ be a binary u-substitution-correcting code of length L. For all L, u, and $M \leq |\mathcal{C}_{\mathsf{sub}}(L, u)|$ we define the code*

$$\mathcal{C}_{\mathsf{MS}}(M, L, u) = \{\mathcal{S} \in \mathcal{X}_M^L : \mathcal{S} \subseteq \mathcal{C}_{\mathsf{sub}}(L, u)\}.$$

Proposition 3.48. *The code $\mathcal{C}_{\mathsf{MS}}(M, L, u)$ is a $(0, M, u)_{\mathbb{S}}$-correcting code.*

The proof is immediate, since every sequence is a codeword of a code that can correct u substitutions. Using binary alternant codes, it is possible to find a lower bound on the redundancy of Construction 3.47.

Theorem 3.49. *There exists a construction for the code $\mathcal{C}_{\mathsf{MS}}(M, L, u)$ with fixed $u \in \mathbb{N}_0$ and $0 < \beta < 1$ which has an asymptotic redundancy of at most*

$$r(\mathcal{C}_{\mathsf{MS}}(M, L, u)) \leq Mu\lceil \log L \rceil + o(M),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. For $\mathcal{C}_{\mathsf{sub}}(L,u)$ in Construction 3.47 we use a binary u-substitution-correcting alternant code of length L, which has redundancy at most $u\lceil \log L \rceil$, cf. [Rot06, Ch. 5.5] and thus obtain a code $\mathcal{C}_{\mathsf{MS}}(M,L,u)$ with redundancy at most

$$\begin{aligned} r(\mathcal{C}_7(M,L,u)) &\leq \log \binom{2^L}{M} - \log \binom{2^{L-u\lceil \log L \rceil}}{M} \leq ML - M \log \left(2^{L-u\lceil \log L \rceil} - M\right) \\ &\leq Mu\lceil \log L \rceil + \frac{M^2 \log \mathrm{e}}{2^{L-u\lceil \log L \rceil} - M}, \end{aligned}$$

where we factored out $2^{L-u\lceil \log L \rceil}$ from the logarithm and used the inequality $\log(1-x) \geq -\frac{x \log \mathrm{e}}{1-x}$ for all $x < 1$ to prove inequality (a). For $M = 2^{\beta L}$, $0 < \beta < 1$ the second term is $o(M)$, which concludes the proof. □

Note that Theorem 3.24 implies that for fixed u this construction is close to optimality.

3.5.5 Tensor-Product Construction for a Single Insertion or Deletion

We now present a construction that is capable of correcting a single insertion or deletion in the whole set $\mathcal{S}$. The following $(0,1,1)_{\mathbb{ID}}$-correcting code is based on VT codes, which have been introduced in Definition 3.43. Our construction now employs the idea of using a single-erasure-correcting code over the checksums of all sequences. The insertion or deletion can then be corrected by first recovering the checksum of the distorted sequence and then using this checksum to correct the insertion/deletion. Note that this idea is similar to the concept of tensor product codes [Wol06].

Construction 3.50. *For an integer a, with $0 \leq a \leq L$, the code construction $\mathcal{C}_{\mathsf{SID}}(M,L,a)$ is given by*

$$\mathcal{C}_{\mathsf{SID}}(M,L,a) = \left\{ \mathcal{S} \in \mathcal{X}_M^L : \sum_{i=1}^{M} s_{\mathsf{VT}}(\boldsymbol{x}_i) \equiv a \bmod (L+1) \right\}.$$

Note that the code can be extended to an arbitrary alphabet size q by applying non-binary VT codes [Ten84].

Proposition 3.51. *For all M, L, a, the code $\mathcal{C}_{\mathsf{SID}}(M,L,a)$ is a $(0,1,1)_{\mathbb{ID}}$-correcting code.*

Proof. We prove the proposition by presenting an appropriate decoding algorithm. Assume $\mathcal{S} \in \mathcal{C}_{\mathsf{SID}}(M,L,a)$ has been transmitted and $\mathcal{S}'$ has been received with a single insertion or deletion in the k-th sequence, for $1 \leq k \leq M$. After the reading process, the $M-1$ error-free sequences $\mathcal{S}_{\mathsf{C}}$ can be identified as they have length exactly L. The checksum of the erroneous sequence $\boldsymbol{x}_k$ can therefore be computed by

$$s_{\mathsf{VT}}(\boldsymbol{x}_k) = a - \sum_{i \in \mathcal{S}_{\mathsf{C}}} s_{\mathsf{VT}}(\boldsymbol{x}_i) \bmod (L+1).$$

The error in $\boldsymbol{x}_k$ is corrected by decoding in the VT code with checksum $s_{\mathsf{VT}}(\boldsymbol{x}_k)$. □

The redundancy of Construction 3.50 is established in the following theorem.

Theorem 3.52. *There exists $0 \leq a \leq L$ such that the redundancy of Construction 3.50 is at most*

$$r(\mathcal{C}_{\mathsf{SID}}(M, L, a)) \leq \log(L+1).$$

Proof. The codes $\mathcal{C}_{\mathsf{SID}}(M, L, a)$, $0 \leq a \leq L$ form a partition over $\mathcal{X}_M^L$ since for each set $\mathcal{S} \in \mathcal{X}_M^L$, the sum over the VT checksums of the individual sequences admits precisely one value. Since there are $L+1$ distinct values for a, based on the pigeonhole principle there exists $0 \leq a \leq L$ such that the cardinality of the code $\mathcal{C}_{\mathsf{SID}}(M, L, a)$ satisfies $|\mathcal{C}_{\mathsf{SID}}(M, L, a)| \geq \binom{2^L}{M}/(L+1)$ and thus its redundancy is at most $\log(L+1)$. □

We have shown in Theorem 3.27 that the redundancy of any $(0, 1, 1)_{\mathbb{ID}}$-correcting code is at least $\log(L) + o(1)$, and thus Construction 3.50 is asymptotically optimal. Note that a generalization to $t > 1$ sequences, each with a single insertion or deletion error, is non-trivial. This is because the VT checksum of a single erroneous sequence can be retrieved even without knowing the order of the remaining sequences. However, for multiple sequences this is not necessarily the case anymore, since standard erasure correcting codes require the knowledge of the ordering of the symbols. Finally, even when the VT checksums could be retrieved, it is not obvious how to assign the checksums with the erroneous sequences. We therefore conclude with the remark that this case remains an interesting open problem for now.

3.6 Conclusion

In this chapter we have introduced a novel combinatorial channel that models the relationship between synthesized and sequenced strands in DNA-based data storage systems. Upper and lower bounds on the optimal size of zero-error codes have been derived using Gilbert-Varshamov-type and sphere-packing arguments. We also proposed several constructions which can be either with or without indices or a reduced version of the indices. Lastly, we derived several more special constructions for a specific set of parameters. It has been illustrated that many of the proposed constructions are close to optimal, such as for the case of substitution, respectively single insertion or deletion errors inside all of the strands. We further have proposed several constructions that can cope with combinations of a loss of sequences and errors inside the sequences. By analyzing the sphere packing bounds and comparing them to our constructions, we have found important insights about the nature of the DNA storage channel. These include the maybe surprising fact that for zero-error codes and a fixed number of errors, correcting insertions or deletions requires less redundancy than correcting substitution errors inside the sequences.

Following our initial publication [Len+18] there appeared several follow-up papers treating similar channel models. Codes over sets of sequences that can correct a given number of arbitrarily placed substitutions haven been constructed recently [SRB21]. Another slight adaptation of the combinatorial channel model has been proposed in [SCSI19], where the sequence-subset distance has been introduced and analyzed and Singleton-like and Plotkin-like code size upper bounds have been derived. More recently, Wei and Schwartz [WS21] derived new converse bounds and proposed several new code constructions for different parameter ranges, using, among others, novel insights for codes correcting insertions and deletions [SB19].

Despite this progress there remain several interesting open questions on this channel. First, the case, where the number of errors scales linearly with the sequence numbers and sequence length has, apart from Corollary 3.18, barely been discussed until now. Second, a construction to

combat the fully general case of $(s, t, u)_{\mathbb{T}}$ is still to be found. While most cases with $s = 0$ are relatively well covered, incorporating error correction against an additional loss of sequences is challenging. Finally, generalizations of this channel model, such as allowing different number of errors in different strands or incorporating other error types, could be of interest.

Chapter 4

Error Correction in Indexed Sets

Several modern storage and communication systems face the particularity that information is conveyed over several sequences, whose order may be arbitrarily permuted. For example, unless equipped with specifically designed primers, the sequences in DNA-based data storage are stored and retrieved in an unordered manner. Similarly, transmitting several information packets over computer networks can result in permutations of the packets. One efficient and practical way to combat the loss of ordering of sequences is to prepend an index to each sequences that denotes the position of the strand in the archive. This approach has been discussed in different settings, such as codes over multisets [KT18b]. For a probabilistic DNA storage channel [Hec+17; SH21] over unordered sequences that are perturbed by substitutions, it has been shown that a simple indexing and error correction scheme over the individual strands is asymptotically rate-optimal. In the context of random access for DNA-based data storage, robust primers with incorporated indices have been developed [Yaz+18]. These primers were designed to meet various properties, such as a large mutual Hamming distance, balanced GC content, and avoiding mutual correlation.

In the sequel we will analyze the approach of indexing sequences in the presence of errors inside the strands. Note that the employment of indices is not a necessity and the more general setup of storing an arbitrary set of sequences has been analyzed in Chapter 3. However, the discussion of index-based schemes is practically important due to its simplicity. We start the discussion by refining the channel model from Section 3.1 to differentiate between errors in the indices and in the remaining part of the sequences in Section 4.1. Based on this refinement, we derive Gilbert-Varshamov (Section 4.2) and sphere-packing bounds (Section 4.3) that provide rigorous evidence that, for certain channel parameters, correcting errors inside the indices (asymptotically) requires less redundancy as compared to errors inside the remaining sequence.[1] We propose a new construction in Section 4.4 that efficiently copes with errors in indexed sets. To this end, we introduce a novel mechanism, called *anchoring*, and show that it is possible to combat the ordering loss of sequences using indices that are protected with only a small amount of redundancy. This allows to use standard coding techniques, such as tensor-product codes to correct errors within the sequences. We focus here on the case of substitution errors, while insertion and deletion errors are deferred for future work. As the model defined in this section builds on that from Chapter 3, we encourage the reader to familiarize with Section 3.1 before reading this chapter.

The results in this chapter have previously been reported in [Len+19b; Len+20d].

[1] Note that such a statement is not directly transferable to probabilistic channels with random errors or combinatorial channels, where the number of errors scales linearly with the archive dimension, as shown in [SH19].

4.1 Combinatorial DNA Storage Channel with Indexed Sets

Consider the combinatorial DNA storage channel from Section 3.1. We now refine that channel model to distinguish between errors in different parts of each strand. This is motivated, first, by the fact that sequencing technologies often report different error rates, depending on the location inside the sequence and, second, by this distinction, we can analyze the effect of errors depending on where they appear. To get a suitable model for the transmission of codes using indices, we particularly distinguish between errors in the first $\log M$ symbols[2] of each sequence and in the remaining part. Denoting by $\boldsymbol{I}(i)$ the binary representation of $i-1$ of length $\log M$, the set of all indexed sets is given by

$$\mathcal{I}_M^L = \{\mathcal{S} = \{(\boldsymbol{I}(1), \boldsymbol{u}_1), (\boldsymbol{I}(2), \boldsymbol{u}_2), \ldots, (\boldsymbol{I}(M), \boldsymbol{u}_M)\} : \boldsymbol{u}_i \in \Sigma_2^{L-\log M}, i \in [M]\},$$

with sequences $\boldsymbol{x}_i = (\boldsymbol{I}(i), \boldsymbol{u}_i) \in \Sigma_2^L$. We will call the sets $\mathcal{S} \in \mathcal{I}_M^L$ *indexed sets*. Each sequence in an indexed set consists of two parts. It starts with a prefix $\boldsymbol{I}(i) \in \Sigma_2^{\log M}$, also referred to as *index*, of length $\log M$ and ends with a suffix $\boldsymbol{u}_i \in \Sigma_2^{L-\log M}$. The prefix is a unique binary representation of the index i and designates the position of this specific sequence in the data set $\mathcal{S}$. The second part of each sequence, $\boldsymbol{u}_i \in \Sigma_2^{L-\log M}$, is referred to as the *data* part of a sequence and can be filled arbitrarily by either user information or redundancy from an error-correcting code, as illustrated later. Accordingly, we refine the combinatorial channel to distinguish between errors in the index and suffix as follows. Let $\mathcal{S} \in \mathcal{X}_M^L$ be the channel input. Let $(\mathcal{S}_\mathsf{C}, \mathcal{S}_\mathsf{L}, \mathcal{S}_\mathsf{E})$ form a partition of the sequences in $\mathcal{S}$ with $|\mathcal{S}_\mathsf{L}| \leq s$ and $|\mathcal{S}_\mathsf{E}| \leq t$ as in Definition 3.3. Then, for each erroneous sequence $\boldsymbol{x}_i \in \mathcal{S}_\mathsf{E}$ the prefix $\boldsymbol{x}_i^\mathsf{I} \triangleq \mathsf{pref}_{\log M}(\boldsymbol{x}_i)$ of length $\log M$ is affected by at most u_1 errors and the suffix $\boldsymbol{x}_i^\mathsf{D} \triangleq \mathsf{suff}_{L-\log M}(\boldsymbol{x}_i)$ of length $L - \log M$ is affected by at most u_2 errors of type $\mathbb{T}$. The received set $\mathcal{S}'$ is thus obtained by

$$\mathcal{S}' = \bigcup_{i=1}^{M} \begin{cases} \{\boldsymbol{x}_i\}, & \text{if } \boldsymbol{x}_i \in \mathcal{S}_\mathsf{C}, \\ \emptyset, & \text{if } \boldsymbol{x}_i \in \mathcal{S}_\mathsf{L}, \\ \{(\boldsymbol{y}_i^\mathsf{I}, \boldsymbol{y}_i^\mathsf{D})\}, & \text{if } \boldsymbol{x}_i \in \mathcal{S}_\mathsf{E} \end{cases},$$

where $\boldsymbol{y}_i^\mathsf{I} \in B^\mathbb{T}(\boldsymbol{x}_i^\mathsf{I}, u_1)$ and $\boldsymbol{y}_i^\mathsf{D} \in B^\mathbb{T}(\boldsymbol{x}_i^\mathsf{D}, u_2)$ are the erroneous outcomes of the prefix and suffix of $\boldsymbol{x}_i$. With this definition, the $(s, t, u_1, u_2)_\mathbb{T}$ channel will refer to the entity which, given an input set $\mathcal{S} \in \mathcal{X}_M^L$, outputs a received set $\mathcal{S}'$ resulting from arbitrary $\mathcal{S}_\mathsf{C}, \mathcal{S}_\mathsf{L}, \mathcal{S}_\mathsf{E}$ and $\boldsymbol{y}_i^\mathsf{I}, \boldsymbol{y}_i^\mathsf{D}$ as described above. This set of all possible channel outputs is denoted by $B^\mathbb{T}(\mathcal{S}, s, t, u_1, u_2)$, analogous to Definition 3.3. Note that the erroneous received sequences are not necessarily distinct from each other or from the error-free sequences and in this case these sequences adjoin and appear as a single sequence at the receiver. Therefore the number of received sequences $|\mathcal{S}'|$ can be less than $M - s$, i.e., $M - t - s \leq |\mathcal{S}'| \leq M$. An illustration of the channel is depicted in Fig. 4.1.

We will discuss codes that prepend an index of length $\log M$ to each sequence within this section. The corresponding zero-error codes are defined as follows.

Definition 4.1. *A code $\mathcal{C} \subseteq \mathcal{I}_M^L$ is called an $(s, t, u_1, u_2)_\mathbb{T}$-**correcting indexed code**, if it can correct a loss of s (or fewer) sequences and u_1, u_2 (or fewer) errors of type $\mathbb{T}$ within the prefix and suffix of each of t (or fewer) sequences, i.e., for any pair $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ with $\mathcal{S}_1 \neq \mathcal{S}_2$, it holds that*

$$B^\mathbb{T}(\mathcal{S}_1, s, t, u_1, u_2) \cap B^\mathbb{T}(\mathcal{S}_2, s, t, u_1, u_2) = \emptyset.$$

[2]Within this section we assume for simplicity that $\log M$ is integer. The case of non-integer $\log M$ can be dealt with analogously.

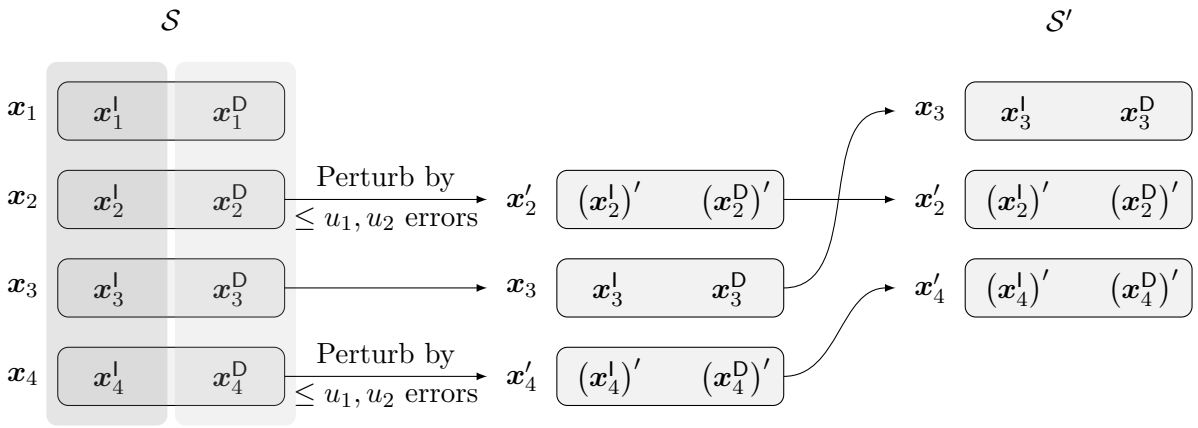


Figure 4.1: Illustration of the $(s, t, u_1, u_2)_{\mathbb{T}}$ channel model. Out of the M input sequences, t sequences are perturbed by at most u_1 errors of type $\mathbb{T}$ in $\boldsymbol{x}_i^{\mathsf{I}}$ and at most u_2 errors of type $\mathbb{T}$ in $\boldsymbol{x}_i^{\mathsf{D}}$. Out of the remaining sequences, s are lost and not observed in the received set. In this example $s = 1, t = 2$.

4.1.1 Relationship to Non-Indexed Codes

The notable differences between Definition 3.5 and 4.1 are that the indexed codes are subsets of $\mathcal{I}_M^L$ and they are defined over the slightly refined channel model, that distinguishes between errors in the prefix and suffix of the sequences. By this definition, an indexed-set code is a set of codewords for which, for each channel output $\mathcal{S}'$, there exists at most one codeword which could have resulted in this exact channel output $\mathcal{S}'$. We distinguish between errors in the index of sequences and data part of the sequences due to the following reasons. It is observed that the sequencing error rates at the beginning of DNA strands are lower with several sequencing technologies [EZ17; HMG19; Org+18]. Second, from a theoretical point of view, errors inside the indices have a different character than those in the data part, as they do not affect data directly but hinder the correct identification of the strand order. We will also elaborate that, for a moderate number of errors, the redundancy required to correct errors in the indices is significantly smaller than that in the data part of sequences. Finally, the channel model is strongly connected to the model presented in Section 3.1 as follows.

Proposition 4.2. *For any $s, t \in \mathbb{N}_0$ the following statements hold.*

1. *Every $(s, t, u)_{\mathbb{T}}$-correcting code is a $(s, t, u_1, u_2)_{\mathbb{T}}$-correcting code, if $u_1 + u_2 \leq u$.*
2. *Every $(s, t, u_1, u_2)_{\mathbb{T}}$-correcting code is a $(s, t, u)_{\mathbb{T}}$-correcting code, if $u \leq \min(u_1, u_2)$.*

Proof. The lemma directly follows from Definitions 3.5 and 4.1. □

4.1.2 Redundancy of Indexing

Since each sequence starts with an index of $\lceil \log M \rceil$ bits that cannot contain any information, the maximum number of information bits that can be stored this way is $M(L - \lceil \log M \rceil)$, assuming no redundancy from error correction. While this solution is attractive for its simplicity, it introduces already a redundancy, which increases linearly in M, which is stated in the following lemma.

Lemma 4.3. *For fixed $0 < \beta < \frac{1}{2}$, the redundancy required for indexing sequences is given by*

$$r(\mathcal{I}_M^L) = M(\lceil \log M \rceil - \log M + \log \mathrm{e}) - \frac{1}{2}\log M + o(1),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. From $M = 2^{\beta L}$ with $0 < \beta < \frac{1}{2}$, we have that $M = o(2^L)$ and $M = \omega(1)$, when $M \to \infty$. Therefore, we can use Lemma A.3 from Appendix A.1 to approximate the binomial coefficient. In fact, we use a slightly stronger result obtained from the proof of Lemma A.3 to arrive at

$$\begin{aligned} r(\mathcal{I}_M^L) &= \log \binom{2^L}{M} - M(L - \lceil \log M \rceil) = M \log \frac{\mathrm{e}2^L}{M} - \frac{1}{2}\log(2\pi M) + O\left(\frac{M^2}{2^L}\right) \\ &\overset{(a)}{=} M(\lceil \log M \rceil - \log M + \log \mathrm{e}) - \frac{1}{2}\log(2\pi M) + o(1), \end{aligned}$$

where we used that $\frac{M^2}{2^L} = 2^{(2\beta-1)L} = o(1)$, if $\beta < \frac{1}{2}$. □

This means that every construction which uses indexing already incurs a redundancy of at best roughly $M \log \mathrm{e}$ bits. Note that this amount can be significant, as the number of sequences M is usually significantly larger than their length L, as explained in Remark 3.7. However, in terms of code rate, it has been shown in [Hec+17] that for the case of no errors inside the sequences, the indexing approach is capacity achieving for a probabilistic version of the DNA storage channel.

4.2 Gilbert-Varshamov Bound for Indexed Codes Under Substitution Errors

We start with deriving lower bounds on the achievable size of error-correcting indexed-set codes based on Gilbert-Varshamov sphere covering arguments. Note that we cannot use results from Section 3.3, as indexed codes are defined as subsets of $\mathcal{I}_M^L$ instead of $\mathcal{X}_M^L$, where $\mathcal{I}_M^L \subseteq \mathcal{X}_M^L$ and thus involve a larger redundancy. Therefore we now derive Gilbert-Varshamov bounds for the case when the codes are subsets of $\mathcal{I}_M^L$. We hereby focus on the case $s = 0$ and substitution errors. For convenience, in the following we denote by $V^{\mathbb{T}}(\mathcal{S}, s, t, u_1, u_2)$ the set of indexed sets $\widetilde{\mathcal{S}} \in \mathcal{I}_M^L$ which have intersecting errors ball with a given $\mathcal{S} \in \mathcal{I}_M^L$, i.e., $B^{\mathbb{T}}(\mathcal{S}, s, t, u_1, u_2) \cap B^{\mathbb{T}}(\widetilde{\mathcal{S}}, s, t, u_1, u_2) \neq \emptyset$.

Theorem 4.4. *There exists a $(0, t, u_1, u_2)_{\mathbb{S}}$-correcting indexed code $\mathcal{C} \subseteq \mathcal{I}_M^L$ with cardinality at least*

$$|\mathcal{C}| \geq \frac{2^{M(L - \log M)}}{\binom{M}{t}^2 (B^{\mathbb{S}}(L - \log M, u_2))^{2t}(t!^2 + \frac{t}{M-t}(B^{\mathbb{S}}(\log M, u_1) + t)^{2t})}.$$

Therefore, for fixed t, u_1, u_2, $0 < \beta < 1$, there exists a $(0, t, u_1, u_2)_{\mathbb{S}}$-correcting indexed code $\mathcal{C} \subseteq \mathcal{I}_M^L$ with redundancy at most

$$r(\mathcal{C}) \leq r(\mathcal{I}_M^L) + 2t \log M + 2tu_2 \log L - 2t \log u_2! + o(1),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. Analogous to Theorem 3.11 it can be shown that there exists a $(0, t, u_1, u_2)_{\mathbb{S}}$-correcting indexed code $\mathcal{C} \subseteq \mathcal{I}_M^L$ with

$$|\mathcal{C}| \cdot \max_{\mathcal{S} \in \mathcal{I}_M^L} |V^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2)| \geq |\mathcal{I}_M^L|.$$

Bounding $|V^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2)|$ from above for all $\mathcal{S} \in \mathcal{I}_M^L$ will be the main task in the following. Abbreviate $B_{\mathsf{I}}(\mathcal{S}) \triangleq B^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2) \cap \mathcal{I}_M^L$ as the set of erroneous sets which are indexed sets and $B_{\mathsf{N}}(\mathcal{S}) \triangleq B_{\mathsf{I}}(\mathcal{S}) \triangleq B^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2) \setminus B_{\mathsf{I}}(\mathcal{S})$ as the remaining possible received words. Further distinguish between sets that have an intersection with $B_{\mathsf{I}}(\mathcal{S})$, i.e., $V_{\mathsf{I}}(\mathcal{S}) \triangleq \{\widetilde{\mathcal{S}} \in \mathcal{I}_M^L : B_{\mathsf{I}}(\mathcal{S}) \cap B^{\mathbb{S}}(\widetilde{\mathcal{S}}, 0, t, u_1, u_2) \neq \emptyset\}$ and sets which have an intersection with $B_{\mathsf{N}}(\mathcal{S})$, i.e., $V_{\mathsf{N}}(\mathcal{S}) \triangleq \{\widetilde{\mathcal{S}} \in \mathcal{I}_M^L : B_{\mathsf{N}}(\mathcal{S}) \cap B^{\mathbb{S}}(\widetilde{\mathcal{S}}, 0, t, u_1, u_2) \neq \emptyset\}$. We first bound $|V_{\mathsf{I}}(\mathcal{S})|$ from above. To begin with, $|B_{\mathsf{I}}(\mathcal{S})| \leq \binom{M}{t} t! (B^{\mathbb{S}}(L - \log M, u_2))^t$, as there are $\binom{M}{t}$ ways to choose the erroneous sequences $\mathcal{S}_{\mathsf{E}}$. For one fixed $\mathcal{S}_{\mathsf{E}}$, there are at most $t!$ error patterns for the errors in the indices that yield indexed sets, as only permutations of the erroneous sequences are potentially possible. For each such choice there are at most $(B^{\mathbb{S}}(L - \log M, u_2))^t$ ways to distribute the errors in the data fields of the t erroneous sequences. From each $\mathcal{S}' \in B_{\mathsf{I}}(\mathcal{S})$, there are again at most $|B_{\mathsf{I}}(\mathcal{S}')|$ ways to arrive at a valid set $\widetilde{\mathcal{S}} \in \mathcal{I}_M^L$ and thus $|V_{\mathsf{I}}(\mathcal{S})| \leq \binom{M}{t}^2 t!^2 (B^{\mathbb{S}}(L - \log M, u_2))^{2t}$. Next we bound $|V_{\mathsf{N}}(\mathcal{S})|$ from above. The number of elements in the error ball is at most $|B_{\mathsf{N}}(\mathcal{S})| \leq \binom{M}{t} B^{\mathbb{S}}(\log M, u_1) B^{\mathbb{S}}(L - \log M, u_2)$, as this is the number of possible error patterns. Let $\mathcal{S}' \in B_{\mathsf{N}}(\mathcal{S})$ and denote by τ' the number of indices that are not present in $\mathcal{S}'$. Then the number of sets $\widetilde{\mathcal{S}} \in \mathcal{I}_M^L$ with $\mathcal{S}' \in B_{\mathsf{N}}(\widetilde{\mathcal{S}})$ is at most $(B^{\mathbb{S}}(\log M, u_1) + t)^t \binom{M}{t-\tau'} (B^{\mathbb{S}}(L - \log M, u_2))^t$. This is because, first, τ' sequences must be distorted such that their indices match the missing indices and there are at most $B^{\mathbb{S}}(\log M, u_1) + t$ candidate sequences in $\mathcal{S}'$ for each missing index. Hence, there are are at most $(B^{\mathbb{S}}(\log M, u_1) + t)^{\tau'}$ possible choices for the errors in the indices to match to missing sequences. The remaining erroneous sequences can be chosen and distorted arbitrarily, resulting in at most $(B^{\mathbb{S}}(\log M, u_1) + t)^{t-\tau'} \binom{M}{t-\tau'} (B^{\mathbb{S}}(L - \log M, u_2))^t$ possibilities. Using $\tau' \geq 1$ for all $\mathcal{S}' \in B_{\mathsf{N}}(\mathcal{S})$ and $|\mathcal{I}_M^L| = 2^{M(L - \log M)}$ yields the first part of the theorem.

Therefore, there exists a $(0, t, u_1, u_2)_{\mathbb{S}}$-correcting code $\mathcal{C} \subseteq \mathcal{I}_M^L$ with redundancy at most

$$\begin{aligned} r(\mathcal{C}) &= \log \binom{2^L}{M} - \log |\mathcal{C}| \overset{(a)}{\leq} \log \frac{\binom{2^L}{M}}{2^{M(L - \log M)}} + \log \binom{M}{t}^2 (B^{\mathbb{S}}(L, u_2))^{2t} + 2 \log t! + o(1) \\ &\overset{(b)}{=} r(\mathcal{I}_M^L) + 2t \log M + 2t u_2 \log L - 2t \log u_2! + o(1), \end{aligned}$$

where inequality (a) holds, because $\frac{t}{M-t} (B^{\mathbb{S}}(\log M, u_1) + t)^{2t} = o(1)$ as $M \to \infty$. For equality (b) we used Lemmas A.2 and A.1 to prove the asymptotic behavior of $\binom{M}{t}$ and $B^{\mathbb{S}}(L, u_2)$. □

The redundancy in Theorem 4.4 is composed of the redundancy required for indexing $r(\mathcal{I}_M^L)$ and some terms for error correction. Interestingly, for the chosen parameter regime, the terms depending on the number of errors inside the indices, u_1 vanish as $M \to \infty$ and are thus asymptotically negligible. However, analyzing the non-asymptotic bound of Theorem 4.4, we see that for the case when t is not fixed, i.e., scales with M the terms depending on u_1 will become more apparent. Summing up, we conclude that for this existential construction, when the number t of erroneous sequences and the number of errors within the sequences is fixed, the redundancy required to correct errors inside the indices is asymptotically negligible. We will exhibit a similar behavior for the case of the converse sphere-packing bound presented in the next section.

4.3 Sphere Packing Bound for Indexed Codes under Substitution Errors

We now present a sphere-packing upper bound on the size of an indexed set code $\mathcal{C} \subseteq \mathcal{I}_M^L$. Note that in principle Theorem 3.23 provides a valid bound even for the case when $\mathcal{C} \subseteq \mathcal{I}_M^L$, however the bound can be strengthened using the restriction $\mathcal{C} \subseteq \mathcal{I}_M^L$. Further note that in contrast to Theorem 3.23 we present the results here on the refined channel model presented in Section 4.1. The main result is as follows.

Theorem 4.5. *The cardinality of any $(0, t, u_1, u_2)_{\mathbb{S}}$-correcting indexed code $\mathcal{C} \subseteq \mathcal{I}_M^L$ is at most*

$$|\mathcal{C}| \leq \frac{2^{M(L-\log M)}}{\binom{M}{t}(B^{\mathbb{S}}(L-\log M, u_2)-1)^t}.$$

Therefore, for fixed $t, u_1 \geq 0, u_2 \geq 1$, $0 < \beta < 1$, the redundancy is at least

$$r(\mathcal{C}) \geq r(\mathcal{I}_M^L) + t\log M + tu_2 \log(L-\log M) - \log(t!u_2!^t) + o(1),$$

when $M \to \infty$ with $M = 2^{\beta L}$.

Proof. Let $\mathcal{C} \subseteq \mathcal{I}_M^L$ be a $(0, t, u_1, u_2)_{\mathbb{S}}$-correcting indexed code. We consider received sets $\mathcal{S}'$ that have not experienced errors in the indices, corresponding to the case that $u_1 = 0$. Therefore all such erroneous outcomes are again indexed sets, i.e., $\mathcal{S}' \in \mathcal{I}_M^L$. Since the error balls of any two codewords in $\mathcal{C}$ must be distinct, every code $\mathcal{C} \subseteq \mathcal{I}_M^L$ satisfies

$$|\mathcal{C}| \leq \frac{|\mathcal{I}_M^L|}{\min\limits_{\mathcal{S}\in\mathcal{I}_M^L} |B^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2) \cap \mathcal{I}_M^L|}$$

Using this inequality we bound the code size $|\mathcal{C}|$ from above. Specifically, for all $\mathcal{S} \in \mathcal{I}_M^L$, we bound the number of erroneous outcomes which are again indexed sets, $|B^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2) \cap \mathcal{I}_M^L|$, from below. Distinct elements $\mathcal{S}' \in B^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2) \cap \mathcal{I}_M^L$ can be constructed as follows. For $u_1 = 0$ the indices of each sequence can be omitted and the stored set can be viewed as a binary array of M rows and L_M columns, where each row corresponds to one sequence. The number of possible error patterns is therefore

$$|B^{\mathbb{S}}(\mathcal{S}, 0, t, u_1, u_2) \cap \mathcal{I}_M^L| \geq \binom{M}{t}(B^{\mathbb{S}}(L-\log M, u_2)-1)^t,$$

as there are $\binom{M}{t}$ ways to choose the erroneous rows and $B^{\mathbb{S}}(L-\log M, u_2)-1$ possible non-zero substitution patterns per sequence. Finally, the case of no errors within the indices is possible, even when $u_1 > 0$, as there are *up to* u_1 errors inside the indices and thus the above bound also holds for arbitrary $u_1 > 0$. This concludes the proof. □

Note that by the definition of the channel it is possible that errors occur in the index of a sequence. However considering these errors for the sphere packing bound does not noticeably improve the bound, as we will illustrate in the following. Let us for simplicity assume that there has only been one error in the i-th sequence, and compare the two cases, where first, the error is in the data part, i.e., $t = 1, u_1 = 0$ and $u_2 = 1$, and second, the error is in the index, i.e., $t = 1, u_1 = 1$

and $u_2 = 0$. In the first case, it is easy to see that a redundancy of at least $\log(M(L - \log M))$ is necessary due to the standard sphere-packing bound on a vector of length $M(L - \log M)$. On the other hand, when the error occurs inside the index of sequence i, resulting in index j, the receiver will see two sequences with the same index j and no sequence with index i. In this case, the receiver only has to decide which of the two sequences with index j originates from position i. As this is merely a binary decision, from a sphere packing point of view, a redundancy of roughly a single bit is necessary to correct this error. This coincides with the previous observation that for a small number of erroneous sequences and errors per sequence, errors within the indices of sequences appear to be less harmful as compared to those inside the data fields.

Remark 4.6. *The fact that for certain channel parameters correcting errors in the indices requires less redundancy than in the data part can be rigorously concretized as follows. Fix $t = 1$, $\beta = \frac{1}{2}$ and an arbitrary positive integer u. Then, using Construction 3.30 with a Reed-Solomon code and $\delta = 1$ gives a $(0, 1, u, 0)_{\mathbb{S}}$-correcting indexed code with redundancy $r(\mathcal{I}_M^L) + \beta L/2$.[3] This redundancy is, for large L, smaller than this required for a $(0, 1, 0, u)_{\mathbb{S}}$-correcting indexed code, which is at least $r(\mathcal{I}_M^L) + \beta L/2 + u \log(L/2) - \log(u!) + o(1)$, by Theorem 4.5. This holds for any fixed u. However, other parameter regimes might behave differently.*

4.4 Anchor-based Codes

Constructing codes that can correct errors from the combinatorial DNA-storage channel, one faces the following challenge. To begin with, errors that are solely in the data part of the sequences can be corrected by standard error-correcting schemes, such as tensor-product codes [Wol06], which we will discuss in more detail later. However, errors in the indices of sequences corrupt the ordering of the sequences, which hinders the direct employment of tensor-product codes. We therefore construct a code that first allows us to reconstruct the correct ordering of the sequences using so called *anchors*, and then uses a tensor-product code to correct the errors in the data part of the sequences. As in the previous two sections, we will focus on the case of no losses of sequences, $s = 0$, and substitution errors inside the sequences. The anchors are defined as follows.

Definition 4.7. *Let $\ell, t, u_1, u_2 \in \mathbb{N}_0$ and $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M \in \Sigma_2^\ell$ be M vectors of length ℓ with $\ell \geq \log M$. The set of* anchor *vectors $\mathcal{A}(\ell, t, u_1, u_2)$ is defined to be*

$$\mathcal{A}(\ell, t, u_1, u_2) = \left\{ (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M) : \begin{array}{l} \boldsymbol{a}_i \in \Sigma_2^\ell, \\ \forall i, j \in [M] \text{ with } d_{\mathsf{H}}(\boldsymbol{I}(i), \boldsymbol{I}(j)) \leq 2u_1 : d_{\mathsf{H}}(\boldsymbol{a}_i, \boldsymbol{a}_j) > 2u_2, \\ (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M) \in \mathsf{MDS}[M, M - 2t] \end{array} \right\}.$$

That is, if two indices $\boldsymbol{I}(i), \boldsymbol{I}(j)$ have distance at most $2u_1$, the corresponding anchors $\boldsymbol{a}_i, \boldsymbol{a}_j$ have to be at distance more than $2u_2$. Further, the equivalents of the vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M$ in the finite field $\mathbb{F}_{2^\ell}$ are a codeword of an MDS code with minimum distance $2t + 1$.

[3] To verify the correctness of this statement, denote by i the index of the erroneous sequence, by j its corrupted index, and by $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M)$ the transmitted Reed-Solomon codeword. If $\boldsymbol{u}_i = \boldsymbol{u}_j$ the two sequences adjoin with one another and the decoder can reconstruct the original codeword by declaring an erasure at position i. If $\boldsymbol{u}_i \neq \boldsymbol{u}_j$, the decoder receives two sequences with index j and has to assign them to position i and j. He can uniquely decide for the correct assignment, due to the following. Denote the code locators of the Reed-Solomon code at positions i and j by α_i and α_j. Computing the syndromes, the codeword that is obtained by permuting $\boldsymbol{u}_i$ with $\boldsymbol{u}_j$ is only a codeword if $\alpha_i \boldsymbol{u}_i + \alpha_j \boldsymbol{u}_j = \alpha_i \boldsymbol{u}_j + \alpha_j \boldsymbol{u}_i$ (the vectors are treated as elements of the finite field $\mathbb{F}_{2^{L/2}}$ and the operations are within this field), which implies $\boldsymbol{u}_i = \boldsymbol{u}_j$, which is a contradiction. Consequently, only the correct association of $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ with the positions i and j yields a valid codeword.

This definition implies that the anchor vectors together with the sequence indices have both a large intra-anchor distance between sequences of one anchor, i.e., the distance between two strands $i \neq j$ is at least $d_{\mathsf{H}}\left((\boldsymbol{I}(i), \boldsymbol{a}_i), (\boldsymbol{I}(j), \boldsymbol{a}_j)\right) \geq 2\min(u_1, u_2)$, and a large inter-anchor distance between two anchors, i.e., two different anchor vectors $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M), (\boldsymbol{a}'_1, \ldots, \boldsymbol{a}'_M) \in \mathcal{A}(\ell, t, u_1, u_2)$ differ in at least $2t+1$ positions $i \in [M]$ due to the MDS code. Note that for $2u_1 = \log M$ and $t = 0$ this definition is equivalent to a standard error-correcting code, which corrects u_2 errors. The redundancy required to force such a constraint will be calculated later. For the case of $t = 0$, the set $\mathcal{A}(\ell, 0, u_1, u_2)$ is called clustering-correcting code, and explicit constructions which require only one bit of redundancy and can be encoded and decoded efficiently can be found in [Shi+19; Shi+22]. The anchoring property will be used to reconstruct the ordering of the sequences. After the ordering of sequences is restored, it is possible to correct the errors in the sequences using tensor-product codes [Wol06], which are defined as follows.

Definition 4.8. *Let $\mathcal{C}_1 \subseteq \Sigma_2^{L_1}$ be a linear binary code of length L_1, redundancy r_1 and parity-check matrix $\boldsymbol{H}_1 \in \Sigma_2^{r_1 \times L_1}$ and let $\mathcal{C}_2 \subseteq \mathbb{F}_{2^{r_1}}^{L_2}$ be a linear code over the field $\mathbb{F}_{2^{r_1}}$. The* tensor-product code *is then defined to be*

$$\mathsf{TPC}(\mathcal{C}_1, \mathcal{C}_2) = \left\{(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{L_2}) : \boldsymbol{u}_i \in \Sigma_2^{L_1}, (\boldsymbol{u}_1 \boldsymbol{H}_1^{\mathrm{T}}, \ldots, \boldsymbol{u}_{L_2} \boldsymbol{H}_1^{\mathrm{T}}) \in \mathcal{C}_2\right\},$$

i.e., the equivalents of the syndromes $\boldsymbol{s}_i = \boldsymbol{u}_i \boldsymbol{H}_1^{\mathrm{T}}$ in the finite field $\mathbb{F}_{2^{r_1}}$ are a codeword of $\mathcal{C}_2$. The overall redundancy of the tensor-product code is $r_1 r_2$ bits.

Correcting errors using the tensor-product code is done as follows [Wol06]. Assume that $\mathcal{C}_1$ can correct u_2 substitutions and $\mathcal{C}_2$ can correct t substitutions. Now a codeword $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{L_2})$ of the tensor product code is transmitted and the word $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_{L_2})$ is received, where at most t vectors $\boldsymbol{u}'_i$ have been affected by at most u_2 errors each. The receiver first computes the syndromes $\boldsymbol{s}'_i = \boldsymbol{u}'_i \boldsymbol{H}_1^{\mathrm{T}}$ of all vectors. Since there are at most t syndromes corrupted, the correct syndromes $\boldsymbol{s}_i$ can be recovered using the code $\mathcal{C}_2$. Now, in each row, u_2 errors can be corrected using the knowledge of the correct syndrome $\boldsymbol{s}_i$ and the code $\mathcal{C}_1$. Combining the anchoring property with the tensor-product code leads to the following construction.

Construction 4.9. *Let $\ell, t, u_1, u_2 \in \mathbb{N}$ with $\ell \geq \log M$. Denote by $\mathcal{C}_1$ a binary u_2-substitution-correcting code of length $L - \log M$ and redundancy r_1 and by $\mathcal{C}_2$ a t-substitution-correcting code of length M and redundancy r_2 over the field $\mathbb{F}_{2^{r_1}}$. We define the code $\mathcal{C}_{\mathsf{anc}}(M, L, \ell, t, u_1, u_2) \subseteq \mathcal{I}_M^L$ by*

$$\mathcal{C}_{\mathsf{anc}}(M, L, \ell, t, u_1, u_2) = \left\{ \begin{array}{l} \mathcal{S} = \{(\boldsymbol{I}(1), \boldsymbol{a}_1, \boldsymbol{v}_1), \ldots, (\boldsymbol{I}(M), \boldsymbol{a}_M, \boldsymbol{v}_M)\} : \\ \qquad (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M) \in \mathcal{A}(\ell, t, u_1, u_2), \\ \qquad ((\boldsymbol{a}_1, \boldsymbol{v}_1), \ldots, (\boldsymbol{a}_M, \boldsymbol{v}_M)) \in \mathsf{TPC}(\mathcal{C}_1, \mathcal{C}_2) \end{array} \right\}.$$

Note that Construction 4.9 depends on the explicit choice of the component codes $\mathcal{C}_1$ and $\mathcal{C}_2$, however we do not explicate this dependence for ease of notation. Regarding encoding into Construction 4.9, it is possible to impose the anchoring constraint and the tensor-product code on the vectors $(\boldsymbol{a_i}, \boldsymbol{v}_i)$ simultaneously by either a systematic encoding of the tensor-product code or by choosing appropriate cosets of $\mathcal{A}(\ell, t, u_1, u_2)$ and $\mathsf{TPC}(\mathcal{C}_1, \mathcal{C}_2)$. Notably, within Construction 4.9, the anchors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M$ also contain user data. The correctness of Construction 4.9 and its decoding algorithm are presented in the following.

Theorem 4.10. *Construction 4.9 is an $(s, t, u_1, u_2)_{\mathbb{S}}$-correcting indexed code.*

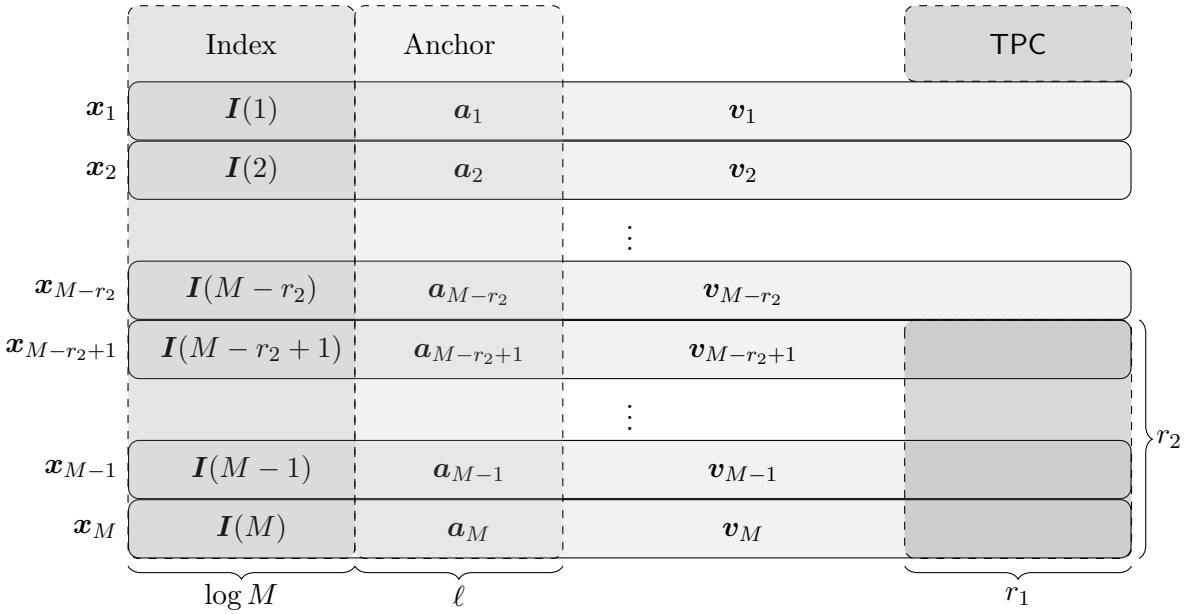


Figure 4.2: Schematic of Construction 4.9

Proof. We will prove the correctness of Construction 4.9 by providing an algorithm that can be used to correct errors from the $(s,t,u_1,u_2)_{\mathbb{S}}$ channel. The decoding algorithm can be split into the following two steps.

1. Retrieve the correct order of sequences using the anchoring property of $\boldsymbol{a}_1,\dots,\boldsymbol{a}_M$.

2. Correct errors inside the sequences using the tensor-product code $\mathsf{TPC}(\mathcal{C}_1,\mathcal{C}_2)$.

Assume $\mathcal{S}=\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_M\}\in\mathcal{C}_{\mathrm{A}}$ has been stored and $\mathcal{S}'=\{\boldsymbol{x}'_1,\dots\boldsymbol{x}'_M\}\in B^{\mathbb{S}}(\mathcal{S},s,t,u_1,u_2)$ has been received after transmission over the $(s,t,u_1,u_2)_{\mathbb{S}}$ channel. Hereby $\boldsymbol{x}'_i=(\boldsymbol{I}(i'),\boldsymbol{a}'_i,\boldsymbol{v}'_i)$ are the received sequences, which are either $\boldsymbol{x}'_i=\boldsymbol{x}_i$, if a sequence was received correctly, i.e., $\boldsymbol{x}_i\in\mathcal{S}_{\mathsf{C}}$, or $\boldsymbol{x}'_i=\boldsymbol{x}_i+\boldsymbol{e}_i$, if a sequence was received in error, i.e., $\boldsymbol{x}_i\in\mathcal{S}_{\mathsf{E}}$. This correct ordering of received sequences is however only used to simplify notation and is not known to the receiver, as the indices $\boldsymbol{I}(i')$ can be erroneous. Note that due to the anchoring property, it is guaranteed that an erroneous sequence can never adjoin with another sequence and therefore $|\mathcal{S}'|=M$.

The anchors can be fully recovered using their MDS property as follows. Declare all positions $i\in[M]$, where there is not exactly one index present, i.e., $i:|\{j:\boldsymbol{I}(j')=\boldsymbol{I}(i)\}|\neq 1$ as erasures, and fill all remaining positions with the corresponding anchors $\boldsymbol{a}'_i$. Although some anchors might have the wrong position, decoding the resulting vector of length M with a unique decoding algorithm yields the correct anchors $\boldsymbol{a}_1,\dots,\boldsymbol{a}_M$ (cf. Construction 3.30 and its proof of correctness in Proposition 3.31). Using the anchors, it is possible to assign each sequence $\boldsymbol{x}'_j$ to its correct position i by finding the single sequence $\boldsymbol{x}'_j\in\mathcal{S}'$ with $d_{\mathsf{H}}\left(\boldsymbol{I}(i),\boldsymbol{I}(j')\right)\leq u_1$ and $d_{\mathsf{H}}\left(\boldsymbol{a}_i,\boldsymbol{a}'_j\right)\leq u_2$. There is exactly one sequence $j=i$ with that property. Assume to the contrary, that there is more than one sequence (apart from the correct sequence $\boldsymbol{x}'_i$), which fulfills this property. Then, there would be a sequence $\boldsymbol{x}'_j$, $j\neq i$ with $d_{\mathsf{H}}\left(\boldsymbol{I}(i),\boldsymbol{I}(j')\right)\leq u_1$ and $d_{\mathsf{H}}\left(\boldsymbol{a}_i,\boldsymbol{a}'_j\right)\leq u_2$, which implies that

$d_{\mathsf{H}}(\boldsymbol{I}(i), \boldsymbol{I}(j)) \leq 2u_1$ and also $d_{\mathsf{H}}(\boldsymbol{a}_i, \boldsymbol{a}_j) \leq 2u_2$, which contradicts the anchoring property. We therefore can reconstruct the array $((\boldsymbol{a}'_1, \boldsymbol{v}'_1), \ldots, (\boldsymbol{a}'_M, \boldsymbol{v}'_M))$ in the correct order.

Since each row $(\boldsymbol{a}'_1, \boldsymbol{v}'_1)$ has at most u_2 errors, these errors can be corrected using the tensor-product code, which completes the proof of the correctness of Construction 4.9. □

The redundancy of Construction 4.9 can be decomposed into the redundancy required for the anchoring property and the redundancy of the tensor-product code and is given as follows.

Theorem 4.11. *For any ℓ, t, u_1, u_2, $\ell \geq \log M$ the redundancy of $\mathcal{C}_{\mathsf{anc}}(M, L, \ell, t, u_1, u_2)$ is*

$$r(\mathcal{C}_{\mathsf{anc}}(M, L, \ell, t, u_1, u_2)) \leq r(\mathcal{I}_M^L) + r_{\mathsf{anc}} + r_1 r_2,$$

where $r_{\mathsf{anc}} \leq 2t\ell - M \log(1 - 2^{-\ell} B^{\mathbb{S}}(\log M, 2u_1) B^{\mathbb{S}}(\ell, 2u_2))$. Therefore, for fixed t, u_1, u_2, when $M \to \infty$, the construction has redundancy

$$r(\mathcal{C}_{\mathsf{anc}}(M, L, \ell, t, u_1, u_2)) \leq r(\mathcal{I}_M^L) + 4t(\log M + (u_1+u_2)\log\log M) + 2tu_2\lceil \log(L - \log M)\rceil + 1 + o(1).$$

Proof. We start by computing the size of $\mathcal{A}(\ell, t, u_1, u_2)$. To begin with, the number of anchors without the MDS code constraint (corresponding to $t = 0$) is at least

$$|\mathcal{A}(\ell, 0, u_1, u_2)| \geq (2^\ell - B^{\mathbb{S}}(\log M, 2u_1) B^{\mathbb{S}}(\ell, 2u_2))^M.$$

This is because we can construct such anchors as follows. First, choose $\boldsymbol{a}_1$ arbitrarily. Then, successively choose $\boldsymbol{a}_i$, starting from $i = 2$, such that none of the previously selected sequences j with $d_{\mathsf{H}}(\boldsymbol{I}(i), \boldsymbol{I}(j)) \leq 2u_1$ satisfy $d_{\mathsf{H}}(\boldsymbol{a}_i, \boldsymbol{a}_l) \leq 2u_2$. As this removes at most $B^{\mathbb{S}}(\log M, 2u_1) B^{\mathbb{S}}(\ell, 2u_2)$ possible sequences, each such chosen sequence has at least $2\ell - B^{\mathbb{S}}(\log M, 2u_1) B^{\mathbb{S}}(\ell, 2u_2)$ options. For more details, see also [Shi+19]. Now, the MDS code with redundancy $2t$ over $\mathbb{F}_{2^\ell}$ has $2^{2t\ell}$ cosets. Since these cosets form a partition of the space $\mathbb{F}_{2^\ell}^M$, there exists one coset of the MDS code with

$$|\mathcal{A}(\ell, t, u_1, u_2)| \geq \frac{1}{2^{2t\ell}} |\mathcal{A}(\ell, 0, u_1, u_2)|$$

by the pigeonhole principle. From this follows the redundancy r_{anc} required for the anchoring property. Next, the redundancy of the tensor-product codes is $r_1 r_2$. Using alternant codes [Rot06, Ch. 5] $\mathcal{C}_1$ and $\mathcal{C}_2$, we obtain for that the redundancy of $\mathcal{C}_1$ is at most $r_1 \leq u_2 \lceil \log(L - \log M)\rceil$ as it is a binary code of length $L - \log M$ of minimum distance $2u_2 + 1$. The code $\mathcal{C}_2$ is of length M over the field $\mathbb{F}_{2^{r_1}}$ and has minimum distance at least $2t + 1$. Choosing it as a subfield subcode of a Reed-Solomon code (such codes are also known as alternant code [Rot06, Ch. 5.5]) of length M over a field of size $2^{\lceil \frac{\log M}{r_1}\rceil r_1}$, its redundancy is at most $r_2 \leq 2t\lceil \frac{\log M}{r_1}\rceil$, if $r_1 \leq \log M$. If $r_1 > \log M$, we can directly use an MDS code for $\mathcal{C}_2$ and obtain $r_2 = 2t$. Using $\lceil \frac{\log M}{r_1}\rceil \leq \frac{\log M}{r_1} + 1$, we obtain for the redundancy of the tensor-product code $r_1 r_2 \leq 2t \log M + r_1$. Therefore, using an appropriate coset for the MDS code of the anchors, as discussed above, yields

$$\begin{aligned} |\mathcal{C}_{\mathsf{anc}}(M, L, \ell, t, u_1, u_2)| \geq &= |\mathcal{A}(\ell, t, u_1, u_2)| \frac{2^{M(L - \log M - \ell)}}{2^{r_1 r_2}} \\ &\geq 2^{-2t\ell} (2^\ell - B^{\mathbb{S}}(\log M, 2u_1) B^{\mathbb{S}}(\ell, 2u_2))^M \frac{2^{M(L - \log M - \ell)}}{2^{2tu_2\lceil \log(L - \log M)\rceil + 2t \log M}}. \end{aligned}$$

The redundancy can then be computed to

$$r(\mathcal{C}_{\mathsf{anc}}(M,L,\ell,t,u_1,u_2)) \leq r(\mathcal{I}_M^L) + 2t\ell - M\log(1-2^{-\ell}B^{\mathbb{S}}(\log M, 2u_1)B^{\mathbb{S}}(\ell,2u_2)) \\ + 2tu_2\lceil\log(L-\log M)\rceil + 2t\log M.$$

Using $\ell = \log M + 2(u_1+u_2)\log\log M$, we can bound the third term by

$$-M\log(1-2^{-\ell}B^{\mathbb{S}}(\log M,2u_1)B^{\mathbb{S}}(\ell,2u_2)) \overset{(a)}{\leq} -M\log\left((1-2^{-\ell}(\log M+2u_1)^{2u_1}(\ell+u_2)^{2u_2})\right) \\ \overset{(b)}{=} M2^{-\ell}(\log M+2u_1)^{2u_1}(\ell+u_2)^{2u_2} + o(1) = 1+o(1),$$

where we additionally used $B^{\mathbb{S}}(L,u) \leq \binom{L+u}{u} \leq (L+u)^u/u!$ for any $L,u \in \mathbb{N}$ in inequality (a) and Lemma A.1 in equation (b). Finally, we obtain

$$r(\mathcal{C}_{\mathsf{anc}}(M,L,\ell,t,u_1,u_2)) \leq r(\mathcal{I}_M^L)+4t(\log M+(u_1+u_2)\log\log M)+2tu_2\lceil\log(L-\log M)\rceil+1+o(1),$$

as stated in the theorem. □

Note that for $t=1$, the construction can be improved by using a Hamming code for $\mathcal{C}$ and an $\mathsf{MDS}[M,1]$ code with redundancy 1 for the anchors is sufficient.

4.5 Conclusion

In this chapter, we have analyzed the approach of indexing sequences for transmission over the combinatorial DNA storage channel. We have refined the channel model to differentiate between errors in the indices and within the sequences, allowing to compare these types of error events and derived lower and upper bounds on the optimal size of indexed zero-error codes over this channel. We have discovered that for a fixed number of erroneous sequences and errors within the sequences, the errors within the indices of sequences appear to be less harmful as compared to those outside the sequences. This observation has been substantiated for a concrete example of channel parameters. We have further developed a novel code construction using anchors that allows to efficiently restore the correct ordering of the sequences, even in the presence of errors.

There are still several interesting open questions within this area of research. Those include the discussion of other error types, such as insertions and deletions and also the incorporation of a possible loss of sequences, as in Chapter 3.

Part II

Communication over Parallel Noisy Sequences

Chapter 5

Unordered Parallel Multinomial Channel

The unordered parallel multinomial channel is a probabilistic model that originates from information-theoretic studies of DNA-based data storage [Hec+17]. The channel has an input of many parallel sequences and comprises two stages. In the first stage, the input sequences are permuted with a uniformly random permutation. In the second stage, each of the resulting sequences is transmitted over an individual channel in parallel. Each such channel comprises a random selection of repetitions followed by a repeated transmission of the input sequence over a q-ary symmetric memoryless channel, where the number of repetitions is according to the previous random selection. In other words, this stage is the parallelization of multiple discrete channels, where each channel is randomly selected out of the family of multinomial channels.[1]

Several variants of this channel have been studied in previous works. Originally, a noiseless channel has been studied [Hec+17], where individual sequences are drawn uniformly and independently from input sequences, resulting in output sequences, whose original sequences are unknown to the receiver. The capacity has been derived for this case and it has been shown that a simple indexing and erasure correction scheme can achieve capacity. Further, the capacity for the case where each sequence is drawn exactly once and transmitted over a binary symmetric channel has been derived in [SH19]. Interestingly, also in this case, it has been proven that a coding scheme that indexes each sequence and protects the whole sequence with a capacity achieving code for the binary symmetric channel can achieve capacity. The results in [SH19] have been extended to the case of transmission over erasure channels [SHS20]. Recently, the capacity has been found for the case where each sequence is drawn according to a Bernoulli distribution and transmitted over a binary symmetric channel [SH21]. In that work it has further been shown that a concatenated code with an outer erasure code and an inner indexing and error correction code can achieve capacity. Another line of work on channels that permute several parallel input sequences is that on arbitrarily permuted parallel channels [Hof+13; WG08]. In their setup, a fixed number of parallel sequences is arbitrarily permuted and then transmitted over known constituent channels. In principle such a communication scenario is similar to ours, differing in the fact that the nature of the constituent channels is deterministic and the number of parallel channels is fixed, while in our work the number of parallel channels is growing with the sequence length and each channel is a random channel, chosen out of a family of possible discrete memoryless channels. A different

[1] The term multinomial channel is derived from [Mit06] and terms a channel that has a single input sequence and multiple output sequences, where each output sequence is the result of transmitting the input sequence over a q-ary symmetric memoryless channel.

type of permutation channels, where the symbols of a single sequence can be permuted have been discussed in [LSY17; Mak18].

In this chapter, we study the unordered parallel multinomial channel for a broad class of drawing distributions and for the case of transmission over independent memoryless symmetric channels. We start by defining the channel model in Section 5.1. We proceed with presenting the result about the capacity of this channel together with necessary definitions in Section 5.2. The presentation is enriched with a discussion of an intuitive interpretation of the capacity formula and with thoughts regarding practical code constructions over the channel. Sections 5.3 and 5.4 are devoted to proving the capacity result by showing that the capacity is an upper bound on achievable information rates and a proof of the existence of codes that have information rates arbitrarily close to capacity, while maintaining vanishing error rates.

Preliminary results to this work have been published in [Len+19a; Len+20c; Len+21f].

5.1 Channel Model

Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \Sigma_q^L$, $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,L})$ be M sequences, each of length L, comprising the input of the channel. Further, let $p_d(\mathtt{y}|\mathtt{x})$, $d \in \mathbb{N}_0$ be a family of probability matrices, i.e., discrete conditional distributions for memoryless channels, where Σ_q is the common input alphabet, i.e, $\mathtt{x} \in \Sigma_q$ and $\mathtt{y} \in \mathcal{Y}_d$ are the output symbols that reside in possibly different, discrete output alphabets $\mathcal{Y}_d$. We will concretize the conditional distributions and output alphabets according to multinomial channels later. The input sequences are shuffled with a random permutation $\boldsymbol{s} = (s_1, \dots, s_M) \in [M]^M$, drawn uniformly from the set of all possible permutations and independently from the channel input $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$. The resulting sequences are $\boldsymbol{z}_1, \dots, \boldsymbol{z}_M$, with

$$\boldsymbol{z}_i = \boldsymbol{x}_{s_i}.$$

Let $\boldsymbol{d} = (d_1, \dots, d_M)$ with $d_i \in \mathbb{N}_0$ be a random variable, called *drawing composition*, with joint distribution $\mathsf{Pr}(\boldsymbol{d} = \mathbf{d})$, where $\mathbf{d}$ denotes the realization of the random variable $\boldsymbol{d}$. We assume that $\boldsymbol{d}$ is independent of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ and the permutation $\boldsymbol{s}$. Each $\boldsymbol{z}_i$ is transmitted over the discrete channel d_i, i.e., according to the conditional probability matrix $p_{d_i}(\mathtt{y}|\mathtt{x})$, resulting in $\boldsymbol{y}_i = (y_{i,1}, \dots, y_{i,L})$, $y_{i,\ell} \in \mathcal{Y}_{d_i}$ such that

$$\mathsf{Pr}(\mathbf{y}_i|\mathbf{z}_i, \mathtt{d}_i) = \prod_{\ell=1}^{L} p_{\mathtt{d}_i}(y_{i,\ell}|z_{i,\ell}).$$

The sequences $\boldsymbol{y}_1, \dots, \boldsymbol{y}_M$ are the output of the channel. The overall input output relationship from input $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ to the output $\boldsymbol{y}_1, \dots, \boldsymbol{y}_M$ is thus

$$\begin{aligned}\mathsf{Pr}(\mathbf{y}_1, \dots, \mathbf{y}_M|\mathbf{x}_1, \dots, \mathbf{x}_M) &= \sum_{\mathbf{d},\mathbf{s}} \mathsf{Pr}(\mathbf{d}, \mathbf{s}) \, \mathsf{Pr}(\mathbf{y}_1, \dots, \mathbf{y}_M|\mathbf{x}_1, \dots, \mathbf{x}_M, \mathbf{d}, \mathbf{s}) \\ &= \sum_{\mathbf{d},\mathbf{s}} \mathsf{Pr}(\mathbf{d}) \, \mathsf{Pr}(\mathbf{s}) \prod_{i=1}^{M} p_{\mathtt{d}_i}(\mathbf{y}_i|\mathbf{x}_{\mathbf{s}_i}),\end{aligned}$$

where we abbreviate $p_{\mathtt{d}_i}(\mathbf{y}_i|\mathbf{x}_{s_i}) = \prod_{\ell=1}^{L} p_{\mathtt{d}_i}(\mathtt{y}_{i,\ell}|\mathbf{x}_{\mathbf{s}_i,\ell})$ due to the memoryless property of the individual channels. We will denote $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_M)$ as the vector of length ML containing all

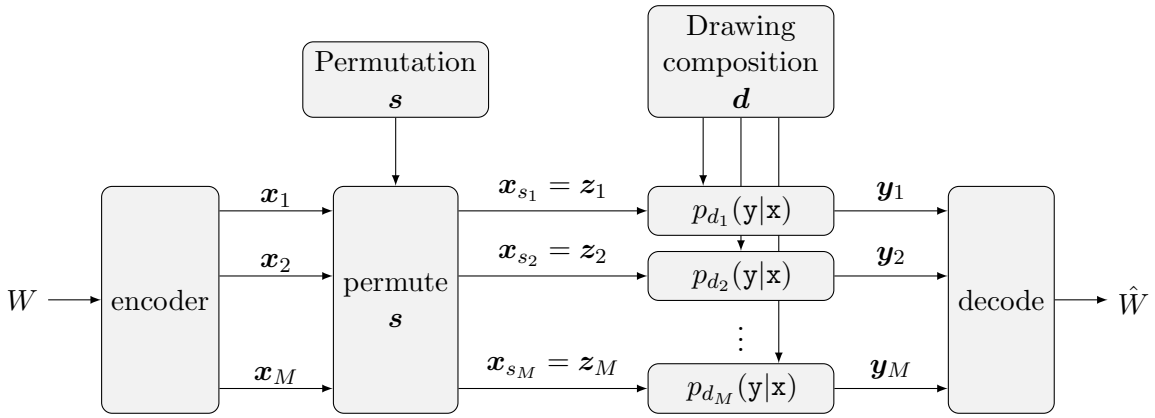


Figure 5.1: Visualization of the transmission scheme over the unordered parallel multinomial channel. A total of M transmit sequences $\boldsymbol{x}_i$ are arbitrarily permuted with each other, producing $\boldsymbol{z}_i = \boldsymbol{x}_{s_i}$. The resulting vectors $\boldsymbol{z}_i$ are transmitted over parallel discrete memoryless channels, where the channel probability matrix is chosen out of the family of channels $p_{d_i}(\mathtt{y}|\mathtt{x})$ according to the drawing composition $\boldsymbol{d}$.

input sequences, $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M)$ containing the shuffled input sequences and $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)$ as the permuted output vector of length ML. We will frequently refer to $\boldsymbol{Y}$ as *output clusters*. Figure 5.1 visualizes the transmission scheme over the unordered parallel multinomial channel.

Remark 5.1. *For drawing distributions that are permutation invariant, i.e., for which the probability* $\mathsf{Pr}\left(d_1 = \mathtt{d}_{\mathtt{s}_1}, \ldots, d_M = \mathtt{d}_{\mathtt{s}_M}\right)$ *is invariant over all permutations* $\mathtt{s}$*, the permutation and drawing stage can be swapped in order. That means, a channel, where the input sequences first traverse the* M *parallel discrete memoryless channels and the results are permuted afterwards has an equivalent input-output relation. However, for drawing distributions that are not permutation invariant, the channels are not necessarily equivalent.*

5.1.1 Multinomial Channel

In this work, we choose the constituent discrete memoryless channel $p_d(\mathtt{y}|\mathtt{x})$ to be the multinomial channel with $d \in \mathbb{N}_0$ draws. The multinomial channel has been proposed and discussed first by Mitzenmacher [Mit06] for binary inputs under the name of the *binomial channel*. Here we refer to the channel as the *multinomial channel* as we discuss the channel for larger input alphabet sizes which results in multinomial distributions, as we will see later. The channel is a discrete memoryless channel with input $x \in \Sigma_q$ and output[2] $y = (y_1, \ldots, y_d) \in \mathcal{Y}_d$ with $\mathcal{Y}_d = \Sigma_q^d$. To this end, we will regard Σ_q as an Abelian finite group over the integers $\{0, 1, \ldots, q-1\}$. For simplicity and explicitness, we will use the standard addition operation of integers modulo q. Each y_i is then obtained from x by transmission over an independent q-ary symmetric channel with error probability p such that

$$(y_1, \ldots, y_d) = (x + e_1, x + e_2, \ldots, x + e_d),$$

[2] Although, strictly speaking, y is a vector of length d over the alphabet Σ_q, we view y as a symbol of the output alphabet $\mathcal{Y}_d$ and thus do not highlight it in bold.

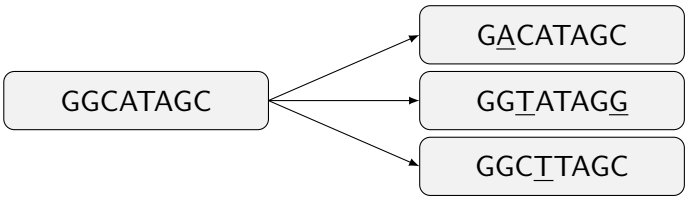


Figure 5.2: Visualization of the multinomial channel with $d = 3$ draws for a sequence $\boldsymbol{x}$ of length 8 over the DNA alphabet $\Sigma_4 = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$. Each received symbol is an element of Σ_4^3 and consists of 3 DNA symbols. Errors are underlined.

for identically distributed and independent $e_i \in \Sigma_q$ with distribution

$$\mathsf{Pr}\left(e_i = \mathsf{e}_i\right) = \begin{cases} 1-p, & \text{if } \mathsf{e}_i = 0 \\ \frac{p}{q-1}, & \text{if } \mathsf{e}_i \neq 0 \end{cases}.$$

Recall that the addition should be performed in the Abelian group of integers Σ_q, e.g., modulo q. The resulting channel probability matrix is thus

$$p_d(\mathsf{y}|\mathsf{x}) = \prod_{i=1}^{d} \begin{cases} 1-p, & \text{if } \mathsf{y}_i = \mathsf{x} \\ \frac{p}{q-1}, & \text{if } \mathsf{y}_i \neq \mathsf{x} \end{cases}.$$

Figure 5.2 displays an exemplary realization of the multinomial channel. Since the channel falls in the class of discrete memoryless channels, Shannon's coding theorem [Sha48] applies and we can find the capacity[3] of the multinomial channel by maximization of the symbol-wise mutual information. For a detailed derivation and discussion of Shannon's theorem, we refer the reader to [CT06, Ch. 7]. The following lemma is a generalization of the capacity formula derived in [Mit06] to non-binary alphabets.

Lemma 5.2. *The capacity of the q-ary multinomial channel with d draws and error probability p is given by*

$$C_{\mathsf{Mul}}(d,p,q) = \frac{1}{q} \sum_{\substack{t_0,\ldots,t_{q-1}: \\ t_0+\cdots+t_{q-1}=d}} \binom{d}{t_0,\ldots,t_{q-1}} \sum_{i=0}^{q-1} (1-p)^{t_i} \left(\frac{p}{q-1}\right)^{d-t_i} \log_q \left(\frac{(1-p)^{t_i} \left(\frac{p}{q-1}\right)^{-t_i}}{\frac{1}{q}\sum_{j=0}^{q-1}(1-p)^{t_j}\left(\frac{p}{q-1}\right)^{-t_j}} \right),$$

where $\binom{d}{t_0,\ldots,t_{q-1}} = \frac{d!}{t_0!\cdot t_1!\ldots t_{q-1}!}$ is the multinomial coefficient. The capacity achieving input distribution is the uniform distribution.

Proof. We begin by noticing that the multinomial channel is a discrete and memoryless channel. Therefore, the capacity can be found by maximizing the symbol-wise mutual information

$$C_{\mathsf{Mul}}(d,p) = \max_{\mathsf{Pr}(x)} I(x;y).$$

[3] For discrete memoryless channels, we refer to their *capacity* by the supremum of information rates for which there exists a code with vanishing error probability, see, e.g. [CT06, Sec. 7.5].

We start by finding the maximizing input distribution $\mathsf{Pr}(x)$ and then compute the mutual information for this distribution. To this end we first show that the channel exhibits symmetry, as defined in [Gal72, Ch. 4.5], which allows to use [Gal72, Thm. 4.5.2] to conclude that the uniform input distribution maximizes the mutual information. Using our notation, a channel is called symmetric[4] if there exists a partition $\Sigma_q^d(1), \ldots, \Sigma_q^d(P)$ of Σ_q^d such that for each part $j \in [P]$ it holds that the multiset $\{\{p_d(\mathtt{y}|\mathtt{x}) : \mathtt{x} \in \Sigma_q\}\}$ is invariant over all $\mathtt{y} \in \Sigma_q^d(j)$ and the multiset $\left\{\left\{p_d(\mathtt{y}|\mathtt{x}) : \mathtt{y} \in \Sigma_q^d(j)\right\}\right\}$ is invariant over all $\mathtt{x} \in \Sigma_q$. In our case, we will partition Σ_q^d into parts for which the set of all numbers of symbol occurrences is the same. More precisely, let $\mathsf{ct}_{\mathtt{x}}(\mathtt{y}) = |\{i \in [d] : \mathtt{y}_i = \mathtt{x}\}|$ be the number of occurrences of the symbol $\mathtt{x} \in \Sigma_q$ in $\mathtt{y}$ and we define the count spectrum

$$\mathsf{ctspec}(\mathtt{y}) = \{\{\mathsf{ct}_{\mathtt{x}}(\mathtt{y}) : \mathtt{x} \in \Sigma_q\}\}$$

as the multiset of the number of occurrences of each symbol in $\mathtt{y}$. We then partition Σ_q^d into $\Sigma_q^d(1), \ldots, \Sigma_q^d(P)$ according to $\mathsf{ctspec}(\mathtt{y})$, i.e., the partition is such that for all $\mathtt{y}_1 \in \Sigma_q^d(j_1)$, and $\mathtt{y}_2 \in \Sigma_q^d(j_2)$, $\mathsf{ctspec}(\mathtt{y}_1) = \mathsf{ctspec}(\mathtt{y}_2)$ holds if and only if $j_1 = j_2$. Therefore $\mathsf{ctspec}(\mathtt{y})$ is constant over all $\mathtt{y}$ in one part $\Sigma_q^d(j)$. Using the fact that

$$p_d(\mathtt{y}|\mathtt{x}) = (1-p)^t \left(\frac{p}{q-1}\right)^{d-t},$$

where $t = |\{i \in [d] : \mathtt{y}_i = \mathtt{x}\}|$ is the number of symbols in $\mathtt{y}$ that are equal to $\mathtt{x}$, we have

$$\{\{p_d(\mathtt{y}|\mathtt{x}) : \mathtt{x} \in \Sigma_q\}\} = \left\{\left\{(1-p)^t \left(\frac{p}{q-1}\right)^{d-t} : t \in \mathsf{ctspec}(\mathtt{y})\right\}\right\},$$

which is invariant over all $\mathtt{y} \in \Sigma_q^d(j)$ in one part. Further, for all $\mathtt{x} \in \Sigma_q$, the number

$$|\{\mathtt{y} \in \Sigma_q^d(j) : \mathsf{ct}_{\mathtt{x}}(\mathtt{y}) = t\}|$$

of words $\mathtt{y} \in \Sigma_q^d(j)$ with a given $\mathsf{ctspec}(\mathtt{y})$ that have exactly t symbols, which are equal to a given $\mathtt{x} \in \Sigma_q$ only depends on j and t and does not depend on $\mathtt{x}$. It follows that the set $\left\{\left\{p_d(\mathtt{y}|\mathtt{x}) : \mathtt{y} \in \Sigma_q^d(j)\right\}\right\}$ does not depend on $\mathtt{x}$ and thus the multinomial channel is symmetric.

Due to the symmetry, we know that the uniform input distribution $\mathsf{Pr}(\mathtt{x}) = \frac{1}{q}$ for all $\mathtt{x} \in \Sigma_q$ maximizes mutual information [Gal72, Thm. 4.5.2]. We thus proceed with computing the entropies $H(y)$ and $H(y|x)$ for uniform inputs. We obtain for the output distribution

$$\mathsf{Pr}(\mathtt{y}) = \sum_{\mathtt{x} \in \Sigma_q} p_d(\mathtt{y}|\mathtt{x})\mathsf{Pr}(\mathtt{x}) = \frac{1}{q} \sum_{a \in \Sigma_q} (1-p)^{\mathsf{ct}_a(\mathtt{y})} \left(\frac{p}{q-1}\right)^{d-\mathsf{ct}_a(\mathtt{y})},$$

where we used that the multiset $\{\{p_d(\mathtt{y}|\mathtt{x}) : \mathtt{x} \in \Sigma_q\}\}$ does not depend on $\mathtt{x}$ as shown above and we thus can express $\mathsf{Pr}(\mathtt{y})$ only as a function of the number of appearances of each symbol $a \in \Sigma_q$

[4] In other words, if we view $p_d(\mathtt{y}|\mathtt{x})$ as a matrix, whose rows are indexed by $\mathtt{x}$ and whose columns are indexed by $\mathtt{y}$, then a channel is symmetric if there exists a partition of the columns of the matrix $p_d(\mathtt{y}|\mathtt{x})$ such that each submatrix, obtained by restricting $p_d(\mathtt{y}|\mathtt{x})$ to the columns corresponding to a part, has the property that the rows are permutations of each other and the columns are permutation of each other.

in $\mathtt{y}$. In order to compute the output entropy, we now use the fact the number of $\mathtt{y} \in \Sigma_q^d$ with a given symbol composition $t_0, \ldots, t_{q-1} \in \mathbb{N}_0$, $t_0 + t_1 + \cdots + t_{q-1} = d$, is given by

$$|\{\mathtt{y} \in \Sigma_q^d : \mathsf{ct}_a(\mathtt{y}) = t_a \ \forall\ a \in \Sigma_q\}| = \binom{d}{t_0, \ldots, t_{q-1}} = \frac{d!}{\prod_{i=0}^{q-1} t_i!},$$

where t_i is the number of times the i-th symbol in Σ_q appears in $\mathtt{y}$ and $\binom{d}{t_1,\ldots,t_q}$ is the multinomial coefficient. Combining all words $\mathtt{y}$ with a given composition in the computation of the output entropy, we obtain

$$H(y) = -\frac{1}{q} \sum_{\substack{t_0,\ldots,t_{q-1}: \\ t_0+\cdots+t_{q-1}=d}} \binom{d}{t_0, \ldots, t_{q-1}} \sum_{i=0}^{q-1} (1-p)^{t_i} \left(\frac{p}{q-1}\right)^{d-t_i} \log_q \left(\frac{1}{q} \sum_{j=0}^{q-1} (1-p)^{t_j} \left(\frac{p}{q-1}\right)^{d-t_j} \right),$$

where the sum over $t_0, \ldots, t_{q-1}$ is over all possible compositions of a vector in Σ_q^d. Finally, we compute the conditional entropy to

$$\begin{aligned} H(y|x) &= -\sum_{\mathtt{y} \in \Sigma_q^d} \sum_{\mathtt{x} \in \Sigma_d} p_d(\mathtt{y}|\mathtt{x}) \mathsf{Pr}\,(\mathtt{x}) \log_q p_d(\mathtt{y}|\mathtt{x}) \\ &= -\frac{1}{q} \sum_{\mathtt{y} \in \Sigma_q^d} \sum_{a \in \Sigma_q} (1-p)^{\mathsf{ct}_a(\mathtt{y})} \left(\frac{p}{q-1}\right)^{d-\mathsf{ct}_a(\mathtt{y})} \log_q \left((1-p)^{\mathsf{ct}_a(\mathtt{y})} \left(\frac{p}{q-1}\right)^{d-\mathsf{ct}_a(\mathtt{y})} \right), \end{aligned}$$

where we used that $\{\{p_d(\mathtt{y}|\mathtt{x}) : \mathtt{x} \in \Sigma_q\}\}$ is independent of $\mathtt{x}$, as shown before and could thus replace the sum over $\mathtt{x}$ by a sum over the symbol count spectrum of $\mathtt{y}$. Combining those $\mathtt{y}$ with the same composition $t_0, \ldots, t_{q-1}$, we arrive at

$$H(y|x) = -\frac{1}{q} \sum_{\substack{t_,\ldots,t_{q-1}: \\ t_0+\cdots+t_{q-1}=d}} \binom{d}{t_0, \ldots, t_{q-1}} \sum_{i=0}^{q-1} (1-p)^{t_i} \left(\frac{p}{q-1}\right)^{d-t_i} \log_q \left((1-p)^{t_i} \left(\frac{p}{q-1}\right)^{d-t_i} \right).$$

Notice that the conditional entropy $H(y|x) = dH_q(p)$, where $H_q(p)$ is the q-ary entropy function, however here we prefer to express the entropy in the above form as this way it can compactly be combined with $H(y)$. The lemma then follows from the fact that $I(x;y) = H(y) - H(y|x)$ with uniformly distributed inputs x. □

The computation of the capacity in Lemma 5.2 can be quite computationally intensive, especially for large d and q. However, it is possible to further simplify the computation using that the outer sum over $t_0, \ldots, t_{q_1}$ can be reduced to a sum over possible multisets $\{\{t_0, \ldots, t_{q-1}\}\}$ as the corresponding summands have the same value. The capacity of the binary multinomial channel can be simplified to the expression

$$C_{\mathsf{Mul}}(d, p, 2) = \sum_{t=0}^{d} \binom{d}{t} (1-p)^{d-t} p^t \log \left(\frac{2}{1 + p^{d-2t}(1-p)^{2t-d}} \right),$$

which is known from [Mit06]. For $d = 1$, the capacity expression simplifies to

$$C_{\mathsf{Mul}}(1, p, q) = 1 - p \log_q(q-1) + p \log_q p + (1-p) \log_q (1-p) = 1 - H_q(p),$$

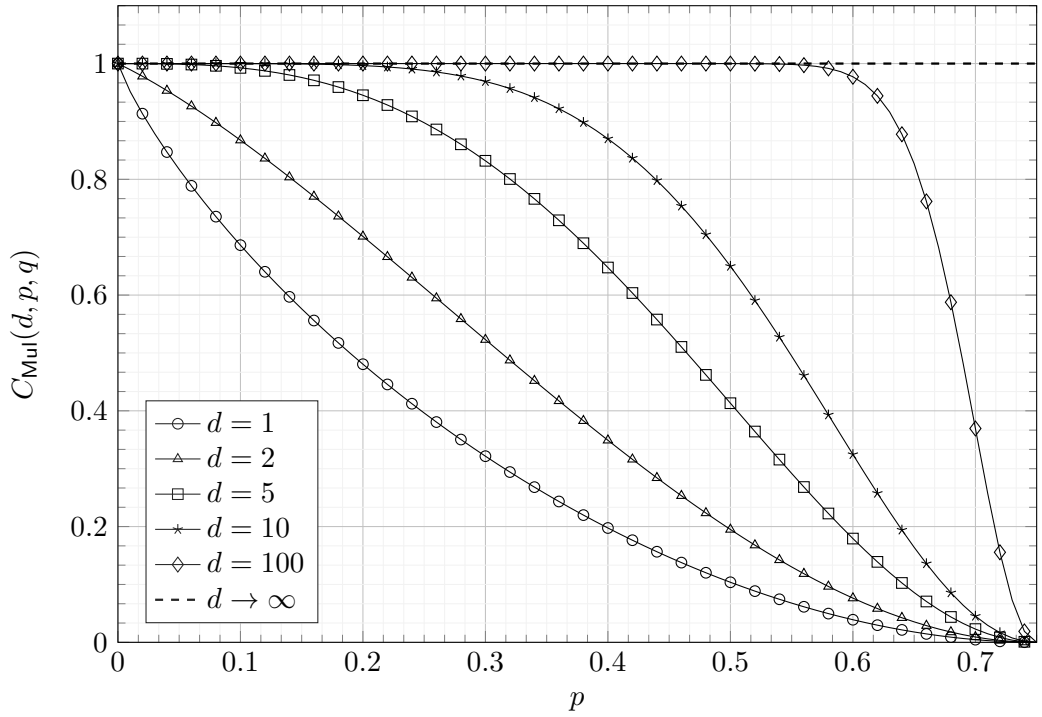


Figure 5.3: Capacity of the multinomial channel for $q = 4$ and different number of draws d over the channel error probability p.

which is precisely the capacity of the standard q-ary symmetric channel. Figure 5.3 shows the capacity of the multinomial channel for the DNA alphabet $\Sigma_4 = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ for different number of draws over the channel error probability.

As the multinomial channel is a discrete memoryless channel it is possible to define the well-known notion of *joint typicality* [CT06, Ch. 7.6] over the channel as follows.

Definition 5.3. *Consider the q-ary multinomial channel with error probability p, d draws and uniform input $x \in \Sigma_q$ with corresponding output $y \in \Sigma_q^{d\times L}$. We define the set of ϵ-jointly typical sequences $\mathbf{x} \in \Sigma_q^L$ and $\mathbf{y} \in \Sigma_q^{d\times L}$ by*

$$\mathcal{T}_{\mathsf{Mul}}^{L,\epsilon}(d,p,q) \triangleq \left\{ (\mathbf{x},\mathbf{y}) \in \Sigma_q^L \times \Sigma_q^{d\times L} : \left| -\frac{\log_q \mathsf{Pr}\left(\mathbf{y}\right)}{L} - H(y) \right| < \epsilon, \left| -\frac{\log_q \mathsf{Pr}\left(\mathbf{x},\mathbf{y}\right)}{L} - H(x,y) \right| < \epsilon \right\}.$$

Note that usually joint typicality includes also a condition on the input $\mathsf{Pr}\left(\mathbf{x}\right)$, however in our case this is trivially fulfilled for all input sequences since we consider uniformly distributed input sequences. The following result can be proven using standard methods for typical sequences and will be useful for the derivation of the converse bound.

Lemma 5.4. *Let the parameters d, p, q of the multinomial channel be arbitrary and fixed. Further, let $\boldsymbol{e}_i = (e_{i,1}, \ldots, e_{i,L}) \in \Sigma_q^L$, $1 \le i \le d$ be random error vectors with identically and independently distributed entries*

$$\mathsf{Pr}\left(e_{i,\ell} = \mathsf{e}_{i,\ell}\right) = \begin{cases} 1-p, & \text{if } \mathsf{e}_{i,\ell} = 0 \\ \frac{p}{q-1}, & \text{if } \mathsf{e}_{i,\ell} \neq 0 \end{cases},$$

for all $1 \leq i \leq d$ *and* $1 \leq \ell \leq L$. *For an arbitrary* $\epsilon > 0$, *let* $\mathcal{F}_\epsilon$ *be the event that the sequence* $\boldsymbol{e}' \triangleq (\boldsymbol{e}_2 - \boldsymbol{e}_1, \ldots, \boldsymbol{e}_d - \boldsymbol{e}_1) \in \Sigma_q^{(d-1)\times L}$ *is an* ϵ*-typical sequence,*[5] *Then, the number of* ϵ*-typical sequences* $\mathbf{e}'$ *is at most* $q^{L(C_{\mathsf{Mul}}(d,p,q)+dH_q(p)-1+\epsilon)}$ *and there exists an integer* $L_d(\epsilon)$ *that depends on* d, p, q *and* ϵ, *such that for all* $L \geq L_d(\epsilon)$, *it holds that* $\mathsf{Pr}\left(\mathcal{F}_\epsilon\right) \geq 1 - \epsilon$.

Proof. To start with, define the single letter variable $e' = (e_2 - e_1, \ldots, e_d - e_1)$, where $e_i \in \Sigma_q$, $1 \leq i \leq d$ are identically and independently distributed variables with

$$\mathsf{Pr}\left(e_i = \mathsf{e}_i\right) = \begin{cases} 1-p, & \text{if } \mathsf{e}_i = 0 \\ \frac{p}{q-1}, & \text{if } \mathsf{e}_i \neq 0 \end{cases}.$$

Notice that with this definition, $\boldsymbol{e}'$ is a sequence of vectors over Σ_q^{d-1}, where each vector is identically and independently distributed according to $\mathsf{Pr}\left(e' = \mathsf{e}'\right)$. We can thus define set of ϵ-typical sequences $\mathcal{T}_{\mathsf{QSC}}^{L,\epsilon}(d,p,q)$ as

$$\mathcal{T}_{\mathsf{QSC}}^{L,\epsilon}(d,p,q) \triangleq \left\{ \mathbf{e}' \in \Sigma_q^{(d-1)\times L} : \left| -\log_q \frac{\mathsf{Pr}\left(\mathbf{e}'\right)}{L} - H(e') \right| < \epsilon \right\}.$$

By the asymptotic equipartition property [CT06, Thm. 3.1.2], it follows that $\mathsf{Pr}\left(\mathcal{F}_\epsilon\right) \geq 1 - \epsilon$ for all $L \geq L_d(\epsilon)$, where $L_d(\epsilon)$ is a constant that depends only on d, p, q and ϵ. Further, the asymptotic equipartition property implies $|\mathcal{T}_{\mathsf{QSC}}^{L,\epsilon}(d,p,q)| \leq |2^{L(H(e')+\epsilon)}|$ for any L. It remains to compute the entropy $H(e')$. We will do so by relating the entropy $H(e')$ with the output entropy of the multinomial channel. To this end, let $x \in \Sigma_q$ be a uniformly distributed random variable that is independent of $e_1, \ldots, e_d$, which will be used as the input of the multinomial channel. Denote further by $y = (x + e_1, \ldots, x + e_d)$ the corresponding output. Then, as the uniform input distribution maximizes the mutual information between x and y, we know from Lemma 5.2 that

$$H(y) = I(x;y) + H(y|x) = C_{\mathsf{Mul}}(d,p,q) + dH_q(p).$$

We can also show that e' is independent of $e_1 + x$ using the following sequence of equations,

$$\begin{aligned} \mathsf{Pr}\left(e' = \mathsf{e}' | e_1 + x = \mathsf{y}_1\right) = \frac{\mathsf{Pr}\left(e' = \mathsf{e}', e_1 + x = \mathsf{y}_1\right)}{\mathsf{Pr}\left(e_1 + x = \mathsf{y}_1\right)} &\overset{(a)}{=} q\mathsf{Pr}\left(e' = \mathsf{e}', e_1 + x = \mathsf{y}_1\right) \\ &\overset{(b)}{=} \sum_{\mathsf{e}_1 \in \Sigma_q} q\mathsf{Pr}\left(e' = \mathsf{e}', e_1 = \mathsf{e}_1, x = \mathsf{y}_1 - \mathsf{e}_1\right) \\ &\overset{(c)}{=} \sum_{\mathsf{e}_1 \in \Sigma_q} \mathsf{Pr}\left(e' = \mathsf{e}', e_1 = \mathsf{e}_1\right) \end{aligned}$$

where we used in equality (a) that $\mathsf{Pr}\left(e_1 + x = \mathsf{y}_1\right) = 1/q$ for all y_1, as the sum with a uniform distribution absorbs the other distribution, i.e., the sum of x with any independent variable e_1 is again uniformly distributed over Σ_q. In equality (b), we demarginalized with respect to x and in equality (c), we used the independence of x from $e_1, \ldots, e_d$ and $\mathsf{Pr}\left(x = \mathsf{y}_1 - \mathsf{e}_1\right) = \frac{1}{q}$ due to the uniform distribution of x. This proves that e' is independent of $e_1 + x$. It follows that

$$H(e') = H(e'|e_1 + x) \overset{(d)}{=} H(e_2 + x, \ldots, e_d + x | e_1 + x) = H(y) - H(y_1) = C_{\mathsf{Mul}}(d,p,q) + dH_q(p) - 1,$$

where we used [CT06, Prob. 2.14] on the conditional entropy of a sum in inequality (d). □

[5] By *typical sequences*, we refer to sequences whose log-probability is ϵ-close to the negative entropy, as introduced by Shannon [Sha48]. For more details, see [CT06, Ch. 3].

5.1.2 Drawing Composition and Frequency

An important entity of to the unordered parallel multinomial channel is the drawing frequency, which counts how often a channel with d draws has been chosen. It is derived from the drawing composition $\boldsymbol{d}$ and is defined as follows.

Definition 5.5. *We define the* drawing frequency $\boldsymbol{n} = (n_0, n_1, \dots)$*, where* $n_d \in \mathbb{N}_0$*,* $d \geq 0$*, as the numbers of occurrences that a sequence has been drawn* d *times, i.e.,*

$$n_d = |\{i \in [M] : d_i = d\}|.$$

The drawing frequency counts the number of times the sequence $\boldsymbol{z}_i$ is transmitted over the q-ary symmetric channel. Since the drawing composition $\boldsymbol{d}$ is a random variable, so is the drawing frequency $\boldsymbol{n}$. The probability mass function of $\boldsymbol{n}$ can thus directly be derived from that of $\boldsymbol{d}$. We will make three restrictions on the distribution of $\boldsymbol{d}$ for our capacity result that will both simplify the derivation of the bounds and ensure that the involved quantities are well-defined. The restrictions are as follows.

Definition 5.6. *Let* $\mathsf{Pr}(\boldsymbol{d} = \mathbf{d})$ *be a given family of permutation-invariant probability mass functions*[6] *for the drawing composition and* $\boldsymbol{n} = (n_0, n_1, \dots)$*,* $n_d = |\{i \in [M] : d_i = d\}|$ *be the derived drawing frequency. We say that the distribution* $\mathsf{Pr}(\boldsymbol{d} = \mathbf{d})$ *is* regular *if it fulfills the following conditions.*

1. Frequency convergence: *The distribution converges to* $\boldsymbol{\nu} = (\nu_0, \nu_1, \dots)$*,* $\nu_d \in \mathbb{R}$*,* $d \geq 0$ *in frequency, i.e., for every* $\epsilon > 0$*,*

$$\lim_{M \to \infty} \mathsf{Pr}\left(\sum_{d \geq 0} \left| \frac{n_d}{M} - \nu_d \right| > \epsilon \right) = 0.$$

2. Bounded draws: *There exists some constant* $c \in \mathbb{R}$ *such that for all* $M \in \mathbb{N}$

$$\mathsf{Pr}\left(\sum_{i=1}^{M} d_i \leq cM \right) = 1.$$

3. Balanced draws: *For every* $\epsilon > 0$*, there exists* $D_\epsilon \in \mathbb{N}$ *such that for all* $M \in \mathbb{N}$

$$\sum_{d \geq D_\epsilon} \mathsf{E}[n_d]\, d \leq \epsilon M$$

First, we restricted to convergent drawing frequencies, meaning that the relative number of times, a channel with d draws is selected converges to a deterministic value. In other words, we call a distribution convergent in frequency, if the relative frequencies $\frac{n_d}{M}$ jointly converge to $\boldsymbol{\nu}$ in probability, when M goes to infinity. One example of a distribution that converges in frequency is that of identical and independent draws as illustrated in the following example.

[6] Through the term *family*, we highlight that the drawing composition has a probability mass for each $M \in \mathbb{N}$.

Example 5.7. *Consider a drawing composition $\boldsymbol{d}$, where the draws are identically and independently distributed with probability mass function $\mathsf{Pr}(d_i = \mathtt{d}) = \nu_{\mathtt{d}}$. Consequently, the joint distribution is given by $\mathsf{Pr}(\boldsymbol{d} = \mathbf{d}) = \prod_{i=1}^{M} \nu_{\mathtt{d}_i}$. The fact that the associated drawing frequency converges to $\boldsymbol{\nu} = (\nu_0, \nu_1, \dots)$ basically follows from an application of the weak law of large numbers on the individual distributions n_d, which are binomial distributed with M trials and success probability ν_d. However, as we need to prove the joint convergence of an infinite number of random variables $n_d, d \geq 0$, we require slightly more elaborate arguments as we show in the following.*

Fix an arbitrarily small $\epsilon > 0$. Due to the fact that ν_d, $d \geq 0$ define a valid probability mass, we can find a $D_0(\epsilon) \in \mathbb{N}_0$ such that $\sum_{d \geq D_0(\epsilon)} \nu_d < \epsilon/4$. We now bound the total deviation of the drawing frequency $\frac{n_d}{M}$ from the distribution ν_d by

$$\sum_{d \geq 0} \left| \frac{n_d}{M} - \nu_d \right| = \sum_{d=0}^{D_0(\epsilon)-1} \left| \frac{n_d}{M} - \nu_d \right| + \sum_{d \geq D_0(\epsilon)} \left| \frac{n_d}{M} - \nu_d \right| \leq \sum_{d=0}^{D_0(\epsilon)-1} \left| \frac{n_d}{M} - \nu_d \right| + \sum_{d \geq D_0(\epsilon)} \left(\frac{n_d}{M} + \nu_d \right).$$

Notice that each n_d is binomial distributed with M trials and success probability ν_d. As $D_0(\epsilon)$ is fixed and finite, we have, using a union bound argument,

$$\mathsf{Pr}\left(\sum_{d=0}^{D_0(\epsilon)-1} \left| \frac{n_d}{M} - \nu_d \right| > \frac{\epsilon}{2} \right) \leq \sum_{d=0}^{D_0(\epsilon)-1} \mathsf{Pr}\left(\left| \frac{n_d}{M} - \nu_d \right| > \frac{\epsilon}{2D_0(\epsilon)} \right) \overset{M \to \infty}{\longrightarrow} 0,$$

which goes to 0 as $M \to \infty$ due to the fact that each individual summand goes to zero by the weak law of large numbers and there are a finite number of summands. On the other hand, we can use that $\sum_{d \geq D_0(\epsilon)} n_d$ is binomial distributed with M trials and success probability $\sum_{d \geq D_0(\epsilon)} \nu_d$. Applying Lemma A.4 on its binomial tail, we obtain

$$\mathsf{Pr}\left(\sum_{d \geq D_0(\epsilon)} \left(\frac{n_d}{M} + \nu_d \right) > \frac{\epsilon}{2} \right) \leq \mathsf{Pr}\left(\sum_{d \geq D_0(\epsilon)} \frac{n_d}{M} > \frac{\epsilon}{4} \right) \leq \mathrm{e}^{-2M\left(\frac{\epsilon}{4} - \sum_{d \geq D_0(\epsilon)} \nu_d\right)^2} \overset{M \to \infty}{\longrightarrow} 0,$$

which also goes to 0, as $M \to \infty$, since $\epsilon/4 > \sum_{d \geq D_0(\epsilon)} \nu_d$ by the choice of $D_0(\epsilon)$ and thus the exponent is negative and tends to $-\infty$. Putting everything together, we obtain that

$$\mathsf{Pr}\left(\sum_{d \geq 0} \left| \frac{n_d}{M} - \nu_d \right| > \epsilon \right) \leq \mathsf{Pr}\left(\sum_{d \geq 0}^{D_0(\epsilon)-1} \left| \frac{n_d}{M} - \nu_d \right| > \frac{\epsilon}{2} \right) + \mathsf{Pr}\left(\sum_{d \geq D_0(\epsilon)} \left(\frac{n_d}{M} + \nu_d \right) > \frac{\epsilon}{2} \right) \overset{M \to \infty}{\longrightarrow} 0,$$

which shows that the drawing composition $\boldsymbol{d}$ converges in frequency to $\nu_0, \nu_1, \dots$ as desired.

The second property states that the total number of draws is at most cM for a constant $c \in \mathbb{R}$. Note that this property is quite common as it is directly fulfilled for the case of drawing distributions that result from drawing cM times from the input sequences. This property entails several useful properties through the fact that the total number of channel outcomes is limited by q^{cML}, simplifying the analysis at several instances.

The following example illustrates that this property is fulfilled for drawing composition with marginal distributions that have only a finite support.

Example 5.8. *Consider a drawing composition $\boldsymbol{d}$, where the individual draws have a marginal distribution with $\mathsf{Pr}(d_i > d) = 0$ for some fixed $d \in \mathbb{N}$ for all $1 \leq i \leq M$. Then,*

$$\mathsf{Pr}\left(\sum_{i=1}^{M} d_i > dM\right) \leq \sum_{i=1}^{M} \mathsf{Pr}(d_i > d) = 0,$$

and the total number of draws is thus at most dM.

Finally, the third property is a technical restriction that facilitates the derivation of the converse bound, as we will see in Section 5.3.

5.2 Capacity of the Unordered Parallel Multinomial Channel

A famous quantity that gives a fundamental limit on the maximal information rate such that reliable communication over a channel is still possible is the capacity. As introduced by Shannon [Sha48], the capacity of a probabilistic channel is the supremum of code rates for which we can asymptotically transmit with vanishing error probability. In the following, we first specify the notion of code rates and error probabilities for the unordered parallel multinomial channel and then proceed with stating our main result about its capacity.

5.2.1 Error-Correcting Codes

Before we state the capacity of the unordered parallel multinomial channel, we introduce the notion of error-correcting codes and code rates over the unordered parallel multinomial channel. The input of the channel is the sequences $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$, each of length L. Thus, a code is a set $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ such that each codeword consists of M sequences, each of length L over the alphabet Σ_q. Consequently, the *rate* of a code $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ is given by

$$R = \frac{\log_q |\mathcal{C}|}{ML}.$$

Note that this definition of rate is slightly different from that in Part I, where we related the size of the code with respect to $\log_q \binom{q^L}{M}$. However, both notions directly translate into each other with a straightforward conversion.

Each code $\mathcal{C}$ is equipped with an encoder

$$\mathsf{enc}_{\mathcal{C}} : [q^{MLR}] \mapsto \mathcal{C}$$

that maps a message $W \in [q^{MLR}]$ to a codeword and a decoder

$$\mathsf{dec}_{\mathcal{C}} : \left(\bigcup_{d \geq 0} \mathcal{Z}_d^L\right)^M \mapsto [q^{MLR}]$$

that outputs an estimate $\widehat{W}$ of the original message W given the received sequences $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M$. Note that the input space of the decoder is the set of all possible output clusters that could

potentially be received, taking into account the fact that the output alphabets of the constituent channels might differ. The error probability of a code $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ and a decoder $\mathsf{dec}_{\mathcal{C}}$ is given by

$$\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}\right) = \frac{1}{q^{MLR}} \sum_{\mathtt{W}=1}^{q^{MLR}} \mathsf{Pr}\left(\mathsf{dec}_{\mathcal{C}}(\boldsymbol{y}_1, \dots \boldsymbol{y}_M) \neq \mathtt{W} | W = \mathtt{W}\right),$$

where $\boldsymbol{y}_1, \dots \boldsymbol{y}_M$ is the random result of transmitting $\mathsf{enc}_{\mathcal{C}}(W) = (\boldsymbol{x}_1, \dots \boldsymbol{x}_M)$ over the unordered parallel multinomial channel. Here we assumed that the messages are chosen uniformly from the set of all messages $W \in [q^{MLR}]$, i.e., $\mathsf{Pr}\left(W = \mathtt{W}\right) = \frac{1}{q^{MLR}}$.

5.2.2 Coding Theorem

As the capacity of a channel is an asymptotic bound on achievable rates, we need to specify how the channel parameters grow to infinity. We consider the regime, where $M \to \infty$ and $M = q^{\beta L}$ for some fixed $0 < \beta < 1$. This choice is motivated by the following two facts. First, the case where M is exponential in L is the interesting case, as for $M \geq q^{\beta L}$ it has been shown in [Hec+17] through counting arguments that no positive rate is achievable (even in the error-free case). For the case where M is subexponential in L, the total number of symbols required for indexing $M \log_q M$ all sequences is thus sub-linear in ML and thus a simple scheme that indexes every sequence and protects each sequence with a strong error-correcting code is rate-optimal. Second, this parameter regime is practically relevant for the case, where one wishes to transmit many relatively short sequences, as is the case in DNA-based archival storage. We use the standard notion of achievable rates and channel capacity as follows.

Definition 5.9. *Fix $0 < \beta < 1, 0 < p < 1$, $q \in \mathbb{N}$ and let $\mathsf{Pr}\left(\boldsymbol{d}\right)$ be a a regular drawing distribution that converges in frequency to $\boldsymbol{\nu}$. Then, a code rate R is* achievable*, if there exists a family of codes $\mathcal{C}(M, L) \subseteq \Sigma_q^{M \times L}$ with $|\mathcal{C}(M, L)| = q^{RML}$ together with a decoder that has vanishing error probability $\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}(M, L)\right) \to 0$ as $M \to \infty$, where $M = q^{\beta L}$.*

The Shannon capacity $C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q)$ is the supremum of achievable rates.

With this definition, for any code rate $R < C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q)$ there exists a family codes with rate R that has vanishing error probability as $M \to \infty$. Conversely, every family of codes with code rate $R > C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q)$ has an non-vanishing error rate. With these prerequisites we are in the position to formulate the main theorem on the capacity of the unordered parallel multinomial channel. Recall to this end the definition of regularity for the probability mass function of the drawing composition $\boldsymbol{d}$ from Definition 5.6, which implies convergence in distribution, a bounded number of draws, and balanced draws.

Theorem 5.10. *Fix the parameters $0 < \beta < 1, q \in \mathbb{N}$, $0 < p < \frac{q-1}{2q}$ with $2\beta < 1 - H_q(2p)$ and let the distribution $\mathsf{Pr}\left(\boldsymbol{d}\right)$ be a given regular distribution that converges in frequency to $\boldsymbol{\nu}$. Then, the capacity of the unordered parallel multinomial channel is given by*

$$C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q) = \sum_{d \geq 0} \nu_d C_{\mathsf{Mul}}(d, p, q) - \beta(1 - \nu_0).$$

5.2.3 Interpretation and Discussion

Conceptually, the unordered parallel multinomial channel is composed of two sub-channels, as it has been illustrated in Figure 5.1. In the first sub-channel, the input sequences $\boldsymbol{x}_i$, $i = 1, \dots, M$ are randomly shuffled according to a permutation, which is chosen uniformly from the set of all permutations of $[M]$. For the second sub-channel, each resulting sequence is transmitted over one of M parallel multinomial channels. The i-th sequence is drawn d_i times, where the draws are chosen according to the realization of the random variable $\boldsymbol{d} = (d_1, \dots, d_M)$.

Capacity of the Ordered Parallel Multinomial Channel

We start by discussing the capacity of the second sub-channel and will refer to this sub-channel, i.e., the channel from $\boldsymbol{z}_1, \dots, \boldsymbol{z}_M$ to $\boldsymbol{y}_1, \dots, \boldsymbol{y}_M$ as the *ordered parallel multinomial* channel. Lemma A.7, which is derived in Appendix A.3, states that the capacity of this sub-channel is

$$C_{\mathsf{OPM}}(\boldsymbol{\nu}, p, q) = \sum_{d \geq 0} \nu_d C_{\mathsf{Mul}}(d, p, q).$$

The intuition of the result is as follows. Recall that the drawing frequency converges to $\boldsymbol{\nu}$. This means that the relative number of times a multinomial channel with d draws occurs within the M channels converges to $\frac{n_d}{M} \to \nu_d$ as $M \to \infty$. Since the multinomial channel with d draws has capacity $C_{\mathsf{Mul}}(d, p, q)$ and its relative frequency of occurrence converges to ν_d, the channel has a capacity of $C_{\mathsf{OPM}}(\boldsymbol{\nu}, \beta, p, q)$. The interested reader finds the rigorous proof of this result in Lemma A.7 in Appendix A.3. Noteworthily, this capacity result requires that the capacity achieving input distribution is the same over all constituent channels, which holds for the case of the multinomial channel, since the uniform distribution is the capacity-achieving input distribution for any number of draws. If the input distributions were to differ among the constituent channels, the problem of finding the capacity would require a deeper analysis using methods similar to those for coding for compound channels [BBT59; Wol59] or random state channels [Ahl86; CS99; GP80]. For an overview of related results, see, e.g., [CK11].

Influence of the Permutation

We now turn to discuss the influence of the first sub-channel, which permutes the sequences. There are in total only $M - n_0$ sequences i which have been drawn at least once, i.e., with $d_i > 0$. As the channel randomly permutes the sequences, the receiver has an uncertainty of $\frac{M!}{n_0!}$ options to associate the $M - n_0$ output clusters with M input sequences. Note that the receiver does not have to associate those outputs with $d_i = 0$ with input sequences as those do not carry any useful information. A random coding argument then suggests that the rate loss induced by this uncertainty is roughly

$$\frac{\log \frac{M!}{n_0!}}{ML} = \frac{M \log M - n_0 \log n_0 + O(M)}{ML} \to \beta - \beta\nu_0,$$

in probability as $M \to \infty$. Here we used that $n_0 = M\nu_0 + o(M)$ with high probability due to the assumed convergence of the drawing frequency. Note that the rigorous derivation of the capacity is more involved since a precise analysis of the effect of the permutation operation on the capacity is non-trivial. We will provide a rigorous proof of Theorem 5.10 in Sections 5.3 and 5.4.

Practical Aspects for Code Design with Rates Approaching Capacity

Interestingly, in contrast to the information-theoretic results, it is still an open problem to find efficiently encodable and decodable schemes that achieve capacity on the noisy drawing channel. This is mainly due to the apriori incertitude how often each input sequence is drawn combined with the loss of ordering of sequences. We will break down these two aspects and existing solutions for each of the aspects in the following.

Channel uncertainty: The amount of information that a receiver may deduce about an input sequence increases with the number of times the input sequence is observed at the output. Since this number is random in the noisy drawing channel, the encoder cannot choose appropriate code rates for each sequence in advance. Thus, in order to operate close to capacity, the input strands must be coded with appropriate cross-correlation such that input sequences with more draws may help in the decoding of those with less (or no) draws. For the case, where the ordering of the output sequences is known to the receiver, rate-matching codes [Hof+13; WG08] provide a solution to construct this correlation. Explicit constructions of rate-matching codes exist [WG08], for example based on erasure codes.

Loss of ordering: Through the random drawing of sequences, the receiver has no immediate information about how the output sequences may be associated with input sequences. This loss of ordering can be combat with indexing, i.e., each input sequence $\boldsymbol{X}_i$ is prepended with a field that designates its index i. However, due to channel noise, also these indices require appropriate protection from errors. As explained in the previous paragraph, indices of sequences with more draws are easier to decode, which implies that also the efficient decoding of the indices requires rate-matching techniques.

The crux of the noisy drawing channel is that the combination of these two techniques in an efficient manner is non-trivial. On the one hand, the rate-matching techniques require a correct ordering of sequences, on the other hand an efficient decoding of the indices requires rate-matching.

For the case of Bernoulli drawing compositions this code design issue could be elegantly solved [SH19] using a scheme, which equips each sequence with an index and a capacity-achieving code on the q-ary symmetric channel, together with an outer erasure code. Here, the erasure code takes the roll of the rate-matching code and a rate-matching decoding of the indices is not necessary, as sequences may be drawn at most once. For drawing distributions with more than one draw per sequence, it remains however an open problem to design codes that protect against channel uncertainty and loss of ordering. One possible solution could be the usage of rate-matching techniques that do not require knowledge of the sequence ordering.

Application to Different Drawing Distributions

The general result of Theorem 5.10 implies a capacity result for a variety of drawing distributions of interest. In particular, we show that Theorem 5.10 recovers the results of [SH21] for Bernoulli drawing compositions.

Example 5.11. *Consider a drawing composition, where the draws are identically and independently distributed Bernoulli variables with success probability* $1-r$, $0 \leq r \leq 1$. *That is,* $d_i \sim \mathsf{Ber}(1-r)$. *It is straight-forward to verify that this distribution is regular according to Definition 5.6, as it converges in frequency to* $\boldsymbol{\nu}$ *where* $\nu_0 = 1-\nu_1 = r$ *and* $\nu_d = 0$ *for all* $d \geq 2$, *by the weak law of large numbers, as shown in Example 5.7. Further, the total number of draws is limited to* M *and we see that the distribution is balanced in the sense of Definition 5.6 by choosing, e.g.,* $D_\epsilon = 2$.

We conclude that the capacity of this channel is thus

$$C_{\mathsf{BerUPM}}(r,p,q) = (1-r)(1-H_q(p)-\beta),$$

for all parameters satisfying $p < \frac{q-1}{2q}$ *and* $2\beta < 1 - H_q(2p)$*, which is precisely the result in [SH21].*

5.3 Converse Bound

We begin with a concise overview of the ideas employed when proving a converse bound, i.e., an upper bound on achievable rates, for the unordered parallel multinomial channel. The starting point is Fano's inequality [FH61], which allows to derive an upper bound on achievable code rates by means of the mutual information between the channel input and output. Hence, an upper bound on the mutual information implies an upper bound on all achievable rates. The main difficulty when deriving upper bounds on the mutual information $I(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X})$ is that both the output entropy $H(\boldsymbol{Y})$ and the conditional entropy $H(\boldsymbol{Y}|\boldsymbol{X})$ depend on the input distribution $\mathsf{Pr}(\boldsymbol{X})$ in a non-trivial way. This is because the effect of the permutation $\boldsymbol{s}$ on the entropy $H(\boldsymbol{Y}|\boldsymbol{X})$ depends on the similarity of the sequences $\boldsymbol{x}_1,\dots,\boldsymbol{x}_M$. In particular, for input distributions that favor similar input sequences, the output clusters $\boldsymbol{y}_1,\dots,\boldsymbol{y}_M$ are also similar to each other and thus a permutation has a smaller effect on the entropy as compared to the case, where the input distribution favors dissimilar input sequences. On the other hand, input distributions that favor similar input sequences $\boldsymbol{x}_1,\dots,\boldsymbol{x}_M$ entail output distributions with smaller entropy $H(\boldsymbol{Y})$. Thus, $\mathsf{Pr}(\boldsymbol{X})$ affects both $H(\boldsymbol{Y})$ and $H(\boldsymbol{Y}|\boldsymbol{X})$ and it is not immediately clear, which distribution maximizes the mutual information. Figure 5.4 illustrates the two cases of similar and dissimilar input and output sequences. To overcome this difficulty, [SH19] showed that introducing a statistic that measures the similarity of the output sequences by means of their Hamming distances helps to find the maximal mutual information.[7] A precise definition of this statistic follows in Definition 5.13. We formalize and adopt this approach for the unordered parallel multinomial channel. The converse bound is formulated in the following lemma.

Lemma 5.12. *Fix* $0 < \beta < 1, q \in \mathbb{N}$, $0 < p < \frac{q-1}{2q}$ *with* $2\beta < 1 - H_q(2p)$ *and let the distribution* $\mathsf{Pr}(\boldsymbol{d})$ *be a regular distribution that converges in frequency to* $\boldsymbol{\nu}$*. Then, any achievable rate* R *over the unordered parallel multinomial channel satisfies*

$$R \leq C_{\mathsf{UPM}}(\boldsymbol{\nu},\beta,p,q).$$

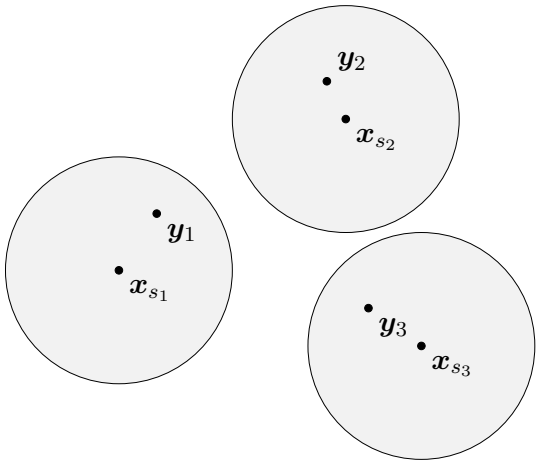
Proof. Let $\mathcal{C} \subseteq \Sigma_q^{M\times L}$ be a code of rate $R = \frac{\log_q|\mathcal{C}|}{ML}$. The code $\mathcal{C}$ has an encoder $\mathsf{enc}_{\mathcal{C}} : [q^{MLR}] \mapsto \mathcal{C}$ and decoder $\mathsf{dec}_{\mathcal{C}}$. Denote by $W \in [q^{MLR}]$ a uniformly random message to be transmitted over the channel and $\widehat{W} = \mathsf{dec}_{\mathcal{C}}(\boldsymbol{Y})$ the output of the decoder, where $\boldsymbol{Y}$ is the result of transmitting $\boldsymbol{X} = \mathsf{enc}_{\mathcal{C}}(W)$ over the unordered parallel multinomial channel. The error probability of this scheme is $\mathsf{Pr}(\mathsf{Err}|\mathcal{C}) = \mathsf{Pr}(W \neq \widehat{W})$ and Fano's inequality implies that

$$R \leq \mathsf{Pr}(\mathsf{Err}|\mathcal{C})\, R + \frac{1 + I(\boldsymbol{X};\boldsymbol{Y})}{ML}.$$

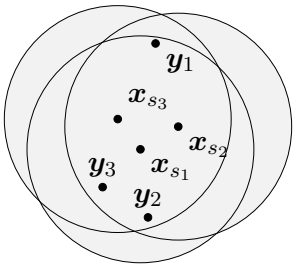
Here we use Lemma 5.14, derived in the sequel, which gives an upper bound on $I(\boldsymbol{X};\boldsymbol{Y})$ to obtain

$$R \leq C_{\mathsf{UPM}}(\boldsymbol{\nu},\beta,p,q) + \mathsf{Pr}(\mathsf{Err}|\mathcal{C})\, R + o(1),$$

[7] An analogous quantity, defined on the channel input, has been employed in [Len+18] to evaluate the number of possible words obtained from a given channel input in the combinatorial setting.



(a) Example of an input $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ for which each permutation of the sequences $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ results in a distinct channel outcome.



(b) Example of an input $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ for which permutations of the sequences $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ result in a similar or the same channel outcomes.

Figure 5.4: Illustration of the effect of the similarity of input sequences on the entropy after permutation. Sequences that are close to each other have a small Hamming distance and, similarly, sequences that are far apart have a large distance.

as $M \to \infty$. By the definition of achievable rates, $\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}\right) \to 0$, as $M \to \infty$, and thus

$$R \leq C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q).$$

□

We proceed with bounding the mutual information $I(\boldsymbol{X}; \boldsymbol{Y})$ from above in a step-by-step fashion. The following statistic of the output sequences is the key ingredient for deriving an analytically tractable upper bound on the entropy terms of the mutual information.

Definition 5.13. *Consider the output $\boldsymbol{Y}$ of the unordered parallel multinomial channel and recall that the number of draws of the i-th output cluster $\boldsymbol{y}_i$ is given by d_i. Write each $\boldsymbol{y}_i$ as*

$$\boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{y}_i^{(1)} \\ \vdots \\ \boldsymbol{y}_i^{(d_i)} \end{pmatrix}.$$

such that each $\boldsymbol{y}_i^{(j)} \in \Sigma_q^L$ corresponds to one sequence of the multinomial channel. For some $\alpha > 0$, we define $\mathcal{U} \subseteq [M]$ to be the largest subset of $[M]$ such that

1. *For all $i \in \mathcal{U}$: $d_i > 0$.*
2. *For all $i, j \in \mathcal{U}$ with $i \neq j$: $d_{\mathsf{H}}\left(\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_j^{(1)}\right) > \alpha L$.*

If the largest subset is not unique, we choose the first according to some (arbitrary) ordering of subsets. We further denote the conditional expectation of $|\mathcal{U}|$ given $\boldsymbol{d} = \mathbf{d}$ by $U_{\mathbf{d}} \triangleq \mathsf{E}\left[|\mathcal{U}| \;|\boldsymbol{d} = \mathbf{d}\right]$.

Note that only the size of $\mathcal{U}$ will be of importance later, and we can choose $\mathcal{U}$ arbitrarily (but deterministic) in the case of ties between several subsets. We further remark that $\mathcal{U}$ is defined only based on the first sequence of each cluster. This is because, given that the first sequence in a cluster is close to that of another cluster, this automatically also restricts the remaining sequences of the cluster, as all sequence stem from the same original sequence. We start with a short explanation of how $\mathcal{U}$ can be used to bound the output entropy $H(\boldsymbol{Y})$. The main idea is the following. Conceptually, we will split the output into *free* clusters, which are contained in $\mathcal{U}$ and into those which are not contained in $\mathcal{U}$. Then, with some careful analysis, the entropy of the free clusters $\mathcal{U}$ is bounded simply by the sum of maximum output entropies of the corresponding multinomial channels. On the other hand, the entropy of those clusters, which are not in $\mathcal{U}$ can be bounded more severely, as their first sequence has to be close to at least one of the sequences in the clusters in $\mathcal{U}$, resulting in a smaller entropy as compared to the free clusters. This means that, if the input distribution is chosen such that it favors sequences that are close in Hamming distance, which corresponds to the case of small $\mathcal{U}$, also the bound on the output entropy $H(\boldsymbol{Y})$ will be smaller. Note that there are a couple of subtleties that need to be overcome when rigorously applying such an argument. One important difference with respect to the derivation of [SH21] is that, the entropy of the non-free clusters is not trivially bounded by L and we thus apply careful combinatorial arguments using Lemma 5.4.

Lemma 5.14. *Fix the parameters $0 < \beta < 1, q \in \mathbb{N}$, $0 < p < \frac{q-1}{2q}$ with $2\beta < 1 - H_q(2p)$ and let the distribution $\mathsf{Pr}(\boldsymbol{d})$ be a given regular distribution that converges in frequency to $\boldsymbol{\nu}$. Then, the mutual information over the unordered parallel multinomial channel satisfies*

$$I(\boldsymbol{X};\boldsymbol{Y}) \leq MLC_{\mathsf{UPM}}(\boldsymbol{\nu},\beta,p,q) + o(ML).$$

Proof. We start by incorporating the permutation $\boldsymbol{s}$ into the mutual information $I(\boldsymbol{X};\boldsymbol{Y})$ as follows. To start with, we have

$$I(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}).$$

The drawing composition $\boldsymbol{d}$ is a function of $\boldsymbol{Y}$, as we can directly infer it from the size of the clusters and thus we can compute the mutual information by

$$\begin{aligned} I(\boldsymbol{X};\boldsymbol{Y}) &= H(\boldsymbol{Y},\boldsymbol{d}) - H(\boldsymbol{Y},\boldsymbol{d}|\boldsymbol{X}) = H(\boldsymbol{Y}|\boldsymbol{d}) + H(\boldsymbol{d}) - H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{d}) - H(\boldsymbol{d}|\boldsymbol{X}) \\ &= H(\boldsymbol{Y}|\boldsymbol{d}) - H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{d}) = I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}). \end{aligned}$$

This means that the condition on $\boldsymbol{d}$ does not change the mutual information. On, the other hand we can express the conditional mutual information as

$$\begin{aligned} I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}) &\overset{(a)}{=} H(\boldsymbol{Y}|\boldsymbol{d}) - H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}) - H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{d}) + H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{d}) \\ &\overset{(b)}{=} H(\boldsymbol{Y}|\boldsymbol{d}) - H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}) + H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{d}) - M\log_q M + O(M), \end{aligned}$$

where we applied the chain rule of entropy twice in equality (a). In equality (b), we used that $H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{d}) = H(\boldsymbol{s})$ due to the permutation invariance of the drawing distribution and that a random uniform permutation of M elements has an entropy of $H(\boldsymbol{s}) = \log_q(M!) = M\log_q M + O(M)$. Expanding the condition on $\boldsymbol{d}$, we obtain

$$I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}) = \sum_{\mathbf{d}} \mathsf{Pr}(\mathbf{d}) \left(H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) - H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}=\mathbf{d}) + H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{d}=\mathbf{d})\right) - \beta ML + O(M).$$

Recall from Definition 5.13, the notation $U_{\mathbf{d}} \triangleq \mathsf{E}\left[|\mathcal{U}| \ |\boldsymbol{d}=\mathbf{d}\right]$ for conditional expectation of the size of the random variable $\mathcal{U}$. Plugging in the bounds on the conditional entropy terms $H(\boldsymbol{Y})$ and $H(\boldsymbol{Y}|\boldsymbol{X})$ from Lemma 5.15, Lemma 5.16 and 5.17, we obtain

$$H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) - H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}=\mathbf{d}) + H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{d}=\mathbf{d}) - \beta ML \leq L\sum_{d\geq 0}\mathtt{n}_d(C_{\mathsf{Mul}}(d,p,q))$$
$$+ (M-\mathtt{n}_0-U_{\mathbf{d}})(\log_q U_{\mathbf{d}} + L(H_q(\alpha)-1)) - U_{\mathbf{d}}\log_q U_{\mathbf{d}} + L\sum_{d\geq D} d\mathtt{n}_d + o(ML) \tag{5.1}$$

for any $p < 2\delta < \alpha < \frac{q-1}{q}$, $D \in \mathbb{N}$ and large enough M. To start with, we show that the last sum over d is asymptotically negligible due to the following. For an arbitrary $\epsilon > 0$ Choose $D = D_\epsilon$, where D_ϵ is the constant guaranteed from Definition 5.6 such that $\sum_{d\geq D_\epsilon}\mathsf{E}\left[n_d\right] d \leq \epsilon M$ for all M. Then, averaging over the drawing composition $\boldsymbol{d}$, one obtains

$$L\sum_{\mathbf{d}}\mathsf{Pr}\left(\boldsymbol{d}=\mathbf{d}\right)\sum_{d\geq D_\epsilon} d\mathtt{n}_d = L\sum_{d\geq D_\epsilon} d\mathsf{E}\left[n_d\right] \leq \epsilon ML.$$

We are now in the position to maximize the mutual information in terms of a maximization over the variable $U_{\mathbf{d}}$ as follows. Denote by $f(U_{\mathbf{d}})$ the terms in the mutual information expression (5.1) that do not vanish and depend on $U_{\mathbf{d}}$, i.e.,

$$f(U_{\mathbf{d}}) = (M-\mathtt{n}_0-U_{\mathbf{d}})(\log_q U_{\mathbf{d}} + L(H_q(\alpha)-1)) - U_{\mathbf{d}}\log_q U_{\mathbf{d}}.$$

Taking the derivative with respect to $U_{\mathbf{d}}$, we see that

$$\begin{aligned} f'(U_{\mathbf{d}}) &= -(\log_q U_{\mathbf{d}} + L(H_q(\alpha)-1)) + \log_q(\mathrm{e})\frac{M-\mathtt{n}_0-U_{\mathbf{d}}}{U_{\mathbf{d}}} - \log_q U_{\mathbf{d}} - \log_q(\mathrm{e}) \\ &> L(1-H_q(\alpha))) - 2\log_q U_{\mathbf{d}} - 2\log_q(\mathrm{e}). \end{aligned}$$

Therefore, $f'(U_{\mathbf{d}}) > 0$ if

$$U_{\mathbf{d}} < \mathrm{e}^{-1}q^{L/2(1-H_q(\alpha)))} = \mathrm{e}^{-1}M^{\frac{1-H_q(\alpha)}{2\beta}}.$$

Hence, if $2\beta < 1-H_q(\alpha)$, the exponent of M is larger than 1 and $f'(U_{\mathbf{d}}) > 0$ for all $0 \leq U_{\mathbf{d}} \leq M$, provided that M is large enough. This means that $f(U_{\mathbf{d}})$ is strictly increasing and using further $U_{\mathbf{d}} \leq M-\mathtt{n}_0$, as $\mathcal{U}$ consists of sequences, which have been drawn at least once, we obtain for $2\beta < 1-H_q(\alpha)$ and large enough M,

$$f(U_{\mathbf{d}}) \leq f(M-\mathtt{n}_0) = -(M-\mathtt{n}_0)\log_q(M-\mathtt{n}_0).$$

We proceed with introducing the event $\mathcal{N}_\epsilon$ for an arbitrary $\epsilon > 0$ as the event on the random variable $\boldsymbol{d}$ that $\sum_{d\geq 0}\left|\frac{\mathtt{n}_d}{M} - \nu_d\right| \leq \epsilon/4$. Splitting the sum over $\mathbf{d}$ in the computation of $I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d})$ according to this event, we obtain

$$I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}) = \sum_{\mathbf{d}}\mathsf{Pr}\left(\mathbf{d}\right) I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) = \sum_{\mathbf{d}\notin\mathcal{N}_\epsilon}\mathsf{Pr}\left(\mathbf{d}\right) I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) + \sum_{\mathbf{d}\in\mathcal{N}_\epsilon}\mathsf{Pr}\left(\mathbf{d}\right) I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d})$$
$$\overset{(c)}{\leq} (\mathsf{Pr}\left(\boldsymbol{d}\notin\mathcal{N}_\epsilon\right)+\epsilon)ML + L\sum_{\mathbf{d}\in\mathcal{N}_\epsilon}\mathsf{Pr}\left(\mathbf{d}\right)\sum_{d\geq 0}\mathtt{n}_d C_{\mathsf{Mul}}(d,p,q) - (M-\mathtt{n}_0)\log_q(M-\mathtt{n}_0) + o(ML),$$

where we used that $I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) \leq H(\boldsymbol{X}|\boldsymbol{d}=\mathbf{d}) \leq ML$ to bound the mutual information in the first term in inequality (c). Analyzing the term inside the sum, we find that for all $\mathbf{d} \in \mathcal{N}_\epsilon$, it holds that

$$L\sum_{d\geq 0} \mathtt{n}_d C_{\mathsf{Mul}}(d,p,q) \leq ML\sum_{d\geq 0} \nu_d C_{\mathsf{Mul}}(d,p,q) + ML\epsilon/4.$$

On the other hand, we can bound

$$\begin{aligned}-(M-\mathtt{n}_0)\log_q(M-\mathtt{n}_0) &\leq -M(1-\nu_0-\epsilon/4)\log(M(1-\nu_0-\epsilon/4))\\ &\leq -\beta ML(1-\nu_0)+\beta ML\epsilon/4+O(M)\end{aligned}$$

for all $\mathbf{d}\in\mathcal{N}_\epsilon$. Using that $\mathsf{Pr}\left(\boldsymbol{d}\notin\mathcal{N}_\epsilon\right)\to 0$ as $M\to\infty$ by the definition of frequency convergence from Definition 5.6, we obtain

$$I(\boldsymbol{X};\boldsymbol{Y}|\boldsymbol{d}) \leq MLC_{\mathsf{UPM}}(\boldsymbol{\nu},\beta,p,q)+o(ML),$$

as we can choose ϵ as small as desired. As we can choose any α with $\alpha > 2p$ and $\alpha < \frac{q-1}{q}$, this yields the statement of the lemma. □

We proceed with a rigorous derivation of the bound on the output entropy $H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d})$ and will bound the entropy terms $H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}=\mathbf{d})$ and $H(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{d}=\mathbf{d})$ afterwards.

5.3.1 Output Entropy Bound

We start with deriving an upper bound on the output entropy, which is given in the following lemma. Recall from the discussion in the beginning of the section, the main proof idea is to split the output clusters according to clusters that are either contained in $\mathcal{U}$ or not and then bound the entropy of both individual parts.

Lemma 5.15. *Fix $0<\beta<1, q\in\mathbb{N}$, $0<p<1$. For any constant $D\in\mathbb{N}$ and $0<\alpha<\frac{q-1}{q}$, the output entropy satisfies*

$$\begin{aligned}H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) \leq\ & L\sum_{d\geq 0}\mathtt{n}_d(C_{\mathsf{Mul}}(d,p,q)+dH_q(p)) + (M-\mathtt{n}_0-U_{\mathbf{d}})(\log_q U_{\mathbf{d}} + L(H_q(\alpha)-1))\\ &+ L\sum_{d\geq D} d\mathtt{n}_d + o(ML).\end{aligned}$$

Proof. We start by investigating the distribution of $\boldsymbol{Y}$ given $\boldsymbol{d}$ and we show that it is equal to the output distribution of an ordered parallel multinomial channel with $\boldsymbol{d}$ draws, whose input distribution is the shuffled input distribution.

$$\mathsf{Pr}\left(\mathbf{Y}|\mathbf{d}\right) = \sum_{\mathbf{Z}} \mathsf{Pr}\left(\mathbf{Z}|\mathbf{d}\right)\mathsf{Pr}\left(\mathbf{Y}|\mathbf{Z},\mathbf{d}\right) \overset{(a)}{=} \sum_{\mathbf{Z}} \mathsf{Pr}\left(\mathbf{Z}\right)\prod_{i=1}^{M} p_{\mathbf{d}_i}(\mathbf{y}_i|\mathbf{z}_i),$$

where in equality (a), we expanded the conditional output probability according to the channel model and used that $\boldsymbol{Z}$ is independent of $\boldsymbol{d}$. Here, $\boldsymbol{Z}$ has the probability distribution

$$\mathsf{Pr}\left(\boldsymbol{Z}=\mathbf{Z}\right) = \sum_{\mathbf{s}} \mathsf{Pr}\left(\mathbf{s}\right)\mathsf{Pr}\left(\boldsymbol{x}_{\mathbf{s}_1}=\mathbf{z}_1,\ldots,\boldsymbol{x}_{\mathbf{s}_M}=\mathbf{z}_M|\boldsymbol{s}=\mathbf{s}\right).$$

This means that $\boldsymbol{Y}$ given $\boldsymbol{d}$ is distributed as M parallel multinomial channels with $\boldsymbol{d}$ draws and a shuffled input distribution. We will continue with splitting the computation of the output entropy into two parts. One, in output clusters $\boldsymbol{y}_i$ with $i \in \mathcal{U}$ and in those which $i \notin \mathcal{U}$. We again use the fact that marginalization reduces entropy, and obtain

$$H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) \leq H(\boldsymbol{Y},\mathcal{U}|\boldsymbol{d}=\mathbf{d}) = H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d},\mathcal{U}) + H(\mathcal{U}|\boldsymbol{d}=\mathbf{d}).$$

Since $\mathcal{U}$ is a subset of $[M]$, it has at most 2^M different possible outcomes and, henceforth, the second entropy term is at most $H(\mathcal{U}|\boldsymbol{d}=\mathbf{d}) \leq \log_q(2)M$. We thus have

$$H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d}) \leq H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d},\mathcal{U}) + O(M) = \sum_{\mathtt{u}\subseteq[M]} \mathsf{Pr}\left(\mathcal{U}=\mathtt{u}|\boldsymbol{d}=\mathbf{d}\right) H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}) + O(M).$$

We are now in the position to use the chain rule of entropy to perform the above mentioned splitting. To this end, for an arbitrary subset $\mathcal{A} \subseteq [M]$, we introduce the notation $\boldsymbol{Y}_{\mathcal{A}} = (\boldsymbol{y}_i : i \in \mathcal{A})$ as the vector containing all output clusters $\boldsymbol{y}_i$ with $i \in \mathcal{A}$. The clusters are ordered according to ascending indices such that the vector is well-defined. We now split the clusters according to the partition $\mathtt{u}$ and $[M] \setminus \mathtt{u}$. We obtain by the chain rule of entropy

$$H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}) = H(\boldsymbol{Y}_{\mathtt{u}}|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}) + H(\boldsymbol{Y}_{[M]\setminus\mathtt{u}}|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u},\boldsymbol{Y}_{\mathtt{u}}). \tag{5.2}$$

We proceed with bounding the first term in (5.2) using the fact that the joint entropy is bounded by the sum of marginal entropies

$$H(\boldsymbol{Y}_{\mathtt{u}}|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}) \leq \sum_{i\in\mathtt{u}} H(\boldsymbol{y}_i|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}). \tag{5.3}$$

Now, fix an arbitrary $\epsilon > 0$ and, to simplify the subsequent analysis, we introduce the random binary indicator variable F_i, $i \in [M]$ that is equal to 0, if the error vectors of the i-th clusters are ϵ-typical as defined in Lemma 5.4 and 1, otherwise. We arrive at

$$\begin{aligned}
H(\boldsymbol{y}_i|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}) &\leq H(\boldsymbol{y}_i,F_i|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}) \\
&\overset{(b)}{\leq} 1 + \sum_{\mathtt{f}_i\in\{0,1\}} \mathsf{Pr}\left(F_i=\mathtt{f}_i|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}\right) H(\boldsymbol{y}_i|\boldsymbol{d}=\mathbf{d},F_i=\mathtt{f}_i,\mathcal{U}=\mathtt{u}) \\
&\overset{(c)}{\leq} 1 + H(\boldsymbol{y}_i|\boldsymbol{d}=\mathbf{d},F_i=0,\mathcal{U}=\mathtt{u}) + \mathsf{Pr}\left(F_i=1|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}\right)\mathtt{d}_i L,
\end{aligned} \tag{5.4}$$

where we used that the entropy of a Bernoulli random variable is at most 1 in (b). Inequality (c) follows from splitting the sum over $\mathtt{f}_i$ into two terms and bounding $\mathsf{Pr}\left(F_i=0|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u}\right) \leq 1$ as well as $H(\boldsymbol{y}_i|\boldsymbol{d}=\mathbf{d},F_i=1,\mathcal{U}=\mathtt{u}) \leq \mathtt{d}_i L$. The latter bound is due to the fact that the cluster $\boldsymbol{y}_i$ consists of $\mathtt{d}_i L$ symbols over Σ_q and thus its entropy is directly bounded by $\mathtt{d}_i L$. Denoting the $\mathtt{d}_i$ sequences of the i-th cluster by $\boldsymbol{y}_i^{(1)},\ldots,\boldsymbol{y}_i^{(\mathtt{d}_i)} \in \Sigma_q^L$ as in Definition 5.13, we can rewrite the above entropy as

$$H(\boldsymbol{y}_i|\boldsymbol{d}=\mathbf{d},F_i=0,\mathcal{U}=\mathtt{u}) = H\left(\boldsymbol{y}_i\middle|\boldsymbol{d}=\mathbf{d},F_i=0,\mathcal{U}=\mathtt{u},\boldsymbol{y}_i^{(1)}\right) + H\left(\boldsymbol{y}_i^{(1)}\middle|\boldsymbol{d}=\mathbf{d},F_i=0,\mathcal{U}=\mathtt{u}\right).$$

The first summand can be bounded using Lemma 5.4, as follows. To start with, for the sake of the argument, consider the distribution of $\boldsymbol{y}_i$ given $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{d}$ without the condition on F_i

and $\mathcal{U}$ first. To this end, denote by $\boldsymbol{e}_i^{(j)} \triangleq \boldsymbol{y}_i^{(j)} - \boldsymbol{x}_i$ the error vectors of the i-th cluster. As we have seen in the beginning of the proof, without the condition on F_i and $\mathcal{U}$ those are distributed according to the multinomial channel model, independent from the input. Now, we can express the conditional distribution of $\boldsymbol{y}_i$ as

$$\begin{aligned}&\mathsf{Pr}\left(\boldsymbol{y}_i = \mathbf{y}_i \middle| \boldsymbol{d} = \mathbf{d}, \boldsymbol{y}_i^{(1)} = \mathbf{y}_i^{(1)}\right)\\ &= \mathsf{Pr}\left(\boldsymbol{e}_i^{(2)} - \boldsymbol{e}_i^{(1)} = \mathbf{y}_i^{(2)} - \mathbf{y}_i^{(1)}, \ldots, \boldsymbol{e}_i^{(\mathtt{d}_i)} - \boldsymbol{e}_i^{(1)} = \mathbf{y}_i^{(\mathtt{d}_i)} - \mathbf{y}_i^{(1)} \middle| d_i = \mathtt{d}_i, \boldsymbol{y}_i^{(1)} = \mathbf{y}_i^{(1)}\right)\end{aligned}$$

as that of the error vectors $\boldsymbol{e}_i^{(j)} - \boldsymbol{e}_i^{(1)}$. By Lemma 5.4, given that $F_i = 0$, i.e., the error vectors are ϵ-typical sequences, the number of possible options for the error vectors is at most $q^{L(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)+\mathtt{d}_i H_q(p)-1+\epsilon)}$. Since further conditioning can only decrease the number of possible options, we have $H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{y}_i^{(1)}) \leq L(C_{\mathsf{Mul}}(\mathtt{d}_i, p, q) + \mathtt{d}_i H_q(p) - 1 + \epsilon)$ and together with the trivial bound $H\left(\boldsymbol{y}_i^{(1)} \middle| \boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}\right) \leq L$, we obtain for all $i \in \mathtt{u}$

$$H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}) \leq L(C_{\mathsf{Mul}}(\mathtt{d}_i, p, q) + \mathtt{d}_i H_q(p) + \epsilon). \tag{5.5}$$

We now bound the second entropy term in (5.2) using again the fact that the joint entropy is at most the sum of the individual entropies and obtain

$$H(\boldsymbol{Y}_{[M]\setminus\mathtt{u}}|\boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u}) \leq \sum_{i \in [M]\setminus\mathtt{u} : \mathtt{d}_i > 0} H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u}), \tag{5.6}$$

where we used that $H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u}) = 0$ for all $i \in [M] \setminus \mathtt{u}$ with $\mathtt{d}_i = 0$. Performing the analogous steps as above to introduce the conditioning on the random variable F_i, we obtain for all $i \in [M] \setminus \mathtt{u}$

$$H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u}) \leq 1 + H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u}) + \mathsf{Pr}\left(F_i = 1 | \boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}\right) \mathtt{d}_i L. \tag{5.7}$$

Using the same notation as in the derivation of the first term, we obtain for all $i \in [M] \setminus \mathtt{u}$

$$\begin{aligned}&H(\boldsymbol{y}_i|\boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u})\\ &\quad= H\left(\boldsymbol{y}_i \middle| \boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{y}_i^{(1)}, \boldsymbol{Y}_\mathtt{u}\right) + H\left(\boldsymbol{y}_i^{(1)} \middle| \boldsymbol{d} = \mathbf{d}, F_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{Y}_\mathtt{u}\right)\\ &\quad\overset{(d)}{\leq} L(C_{\mathsf{Mul}}(\mathtt{d}_i, p, q) + \mathtt{d}_i H_q(p) - 1 + \epsilon) + \log_q |\mathtt{u}| + L H_q(\alpha),\end{aligned} \tag{5.8}$$

where the first summand in inequality (d) has been bounded using the same arguments as above. The second summand has been bounded using the fact that, given $\mathcal{U} = \mathtt{u}$ and $\boldsymbol{Y}_\mathtt{u}$, there are only $|\mathtt{u}| q^{L H_q(\alpha)}$ options for $\boldsymbol{y}_i^{(1)}$, as $\boldsymbol{y}_i^{(1)}$ has to have distance at most αL to one of the sequences in $\boldsymbol{Y}_\mathtt{u}$. Note that this entropy bound on the size of the Hamming ball is only valid if $\alpha < \frac{q-1}{q}$. Plugging

Table 5.1: Illustration of the bounds on the entropy used used for different types of output clusters in the proof of Lemma 5.15

Notation	Meaning	Entropy Bound
$F_i = 1$	The cluster's error vectors are not typical sequences.	$\mathtt{d}_i L$
$i \in \mathtt{u} \wedge F_i = 0$	The cluster's error vectors are typical sequences and the cluster is free.	$(C_{\mathsf{Mul}}(\mathtt{d}_i, p, q) + \mathtt{d}_i H_q(p))L$
$i \notin \mathtt{u} \wedge F_i = 0$	The cluster's error vectors are typical sequences and the first sequence in the cluster is not free.	$(C_{\mathsf{Mul}}(\mathtt{d}_i, p, q) + \mathtt{d}_i H_q(p))L - (1 - H_q(\alpha))L + \log_q \lvert\mathtt{u}\rvert$

(5.6), (5.7), (5.8) and (5.3), (5.4), (5.5) into (5.2), we conclude that

$$
\begin{aligned}
H(\boldsymbol{Y}|\boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}) &\leq \sum_{i=1}^{M} L(C_{\mathsf{Mul}}(\mathtt{d}_i, p, q) + \mathtt{d}_i H_q(p) + \epsilon) + \sum_{i \in [M] \setminus \mathtt{u} : \mathtt{d}_i > 0} \log_q |\mathtt{u}| + L(H_q(\alpha) - 1) \\
&\quad + \sum_{i=1}^{M} \mathsf{Pr}\left(F_i = 1 | \boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}\right) \mathtt{d}_i L + M \\
&\overset{(e)}{=} \sum_{d \geq 0} L\mathtt{n}_d (C_{\mathsf{Mul}}(d, p, q) + d H_q(p) + \epsilon) + (M - \mathtt{n}_0 - |\mathtt{u}|)(\log_q |\mathtt{u}| + L(H_q(\alpha) - 1)) \\
&\quad + \sum_{i=1}^{M} \mathsf{Pr}\left(F_i = 1 | \boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}\right) \mathtt{d}_i L + M,
\end{aligned}
$$

where in equality (e) we replaced the sum over i by a sum over d. We further used that the number of terms in the sum over $i \in [M] \setminus \mathtt{u}$ with $\mathtt{d}_i > 0$ is precisely $M - \mathtt{n}_0 - |\mathtt{u}|$. The different cases according to which we computed the entropy of the output clusters have been summarized in Table 5.1. As a reminder we note that the above inequality holds for all $0 < \epsilon < 1$, where F_i is the random variable that depends on ϵ through the ϵ-typical sequences from Lemma 5.4. We continue with bounding the last summand from above.

$$
L \sum_{\mathtt{u}} \mathsf{Pr}\left(\mathcal{U} = \mathtt{u} | \boldsymbol{d} = \mathbf{d}\right) \sum_{i=1}^{M} \mathsf{Pr}\left(F_i = 1 | \boldsymbol{d} = \mathbf{d}, \mathcal{U} = \mathtt{u}\right) \mathtt{d}_i = L \sum_{i=1}^{M} \mathsf{Pr}\left(F_i = 1 | \boldsymbol{d} = \mathbf{d}\right) \mathtt{d}_i.
$$

Note that by Lemma 5.4, $\mathsf{Pr}\left(F_i = 1 | \boldsymbol{d} = \mathbf{d}\right) < \epsilon$ for all $L \geq L_{\mathtt{d}_i}(\epsilon)$. Since we cannot guarantee that $L \geq L_{\mathtt{d}_i}$ for all $i \in [M]$, we restrict on a large but constant number of draws. To this end, denote by $D \in \mathbb{N}$ the arbitrary constant from the Lemma statement. We split the sum over i

according to terms with more or less than D draws and obtain

$$L\sum_{i=1}^{M}\mathsf{Pr}\left(F_i=1|\boldsymbol{d}=\mathbf{d}\right)\mathtt{d}_i = L\sum_{i:\mathtt{d}_i<D}\mathsf{Pr}\left(F_i=1|\boldsymbol{d}=\mathbf{d}\right)\mathtt{d}_i + L\sum_{i:\mathtt{d}_i\geq D}\mathsf{Pr}\left(F_i=1|\boldsymbol{d}=\mathbf{d}\right)\mathtt{d}_i$$
$$\overset{(f)}{\leq} \epsilon L\sum_{i:\mathtt{d}_i<D}\mathtt{d}_i + L\sum_{i:\mathtt{d}_i\geq D}\mathsf{Pr}\left(F_i=1|\boldsymbol{d}=\mathbf{d}\right)\mathtt{d}_i \overset{(g)}{\leq} \epsilon cML + L\sum_{i:\mathtt{d}_i\geq D}\mathtt{d}_i,$$

where we used that $\mathsf{Pr}\left(F_i=1|\boldsymbol{d}=\mathbf{d}\right)<\epsilon$ for all $L\geq L_{\mathtt{d}_i}(\epsilon)$ and thus inequality (f) holds for all $L\geq\max_{0\leq d<D}L_d(\epsilon)$. We further used in inequality (g) that the total number of draws is bounded by cM, according to Definition 5.6. We are now in the position to compute the overall entropy

$$\begin{aligned}
H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d},\mathcal{U}) &= \sum_{\mathtt{u}\subseteq[M]}\mathsf{Pr}\left(\mathcal{U}=\mathtt{u}|\boldsymbol{d}=\mathbf{d}\right)H(\boldsymbol{Y}|\boldsymbol{d}=\mathbf{d},\mathcal{U}=\mathtt{u})\\
&\leq L\sum_{d\geq 0}\mathtt{n}_d(C_{\mathsf{Mul}}(d,p,q)+dH_q(p)) + \mathsf{E}[(M-\mathtt{n}_0-|\mathcal{U}|)(\log_q|\mathcal{U}|+L(H_q(\alpha)-1))|\boldsymbol{d}=\mathbf{d}]\\
&\quad + L\sum_{i:\mathtt{d}_i\geq D}\mathtt{d}_i + \epsilon cML + O(M)\\
&\overset{(h)}{\leq} L\sum_{d\geq 0}\mathtt{n}_d(C_{\mathsf{Mul}}(d,p,q)+dH_q(p)) + (M-\mathtt{n}_0-U_{\mathbf{d}})(\log_q U_{\mathbf{d}}+L(H_q(\alpha)-1))\\
&\quad + L\sum_{i:\mathtt{d}_i\geq D}\mathtt{d}_i + \epsilon cML + O(M),
\end{aligned}$$

where inequality (h) is due to Jensen inequality and the fact that $-|\mathcal{U}|\log_q|\mathcal{U}|$ is a concave function in $|\mathcal{U}|$. Note that this inequality holds for any constant D and large enough $L\geq\max_{0\leq d<D}L_d(\epsilon)$. The claim of the lemma follows as we can choose ϵ arbitrarily small.

□

5.3.2 Ordered Conditional Entropy Bound

Next, we compute the conditional output entropy, conditioned on the permutation $\boldsymbol{s}$.

Lemma 5.16. *Fix $0<\beta<1, q\in\mathbb{N}$, $0<p<1$. Then,*

$$H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}=\mathbf{d}) = \sum_{d\geq 0}\mathtt{n}_d dH_q(p).$$

Proof. We can use the fact that $\boldsymbol{Z}$ is a function of $\boldsymbol{X}$ and $\boldsymbol{s}$, since $\boldsymbol{z}_i=\boldsymbol{x}_{s_i}$ to obtain that the conditional output entropy is given by

$$H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{d}=\mathbf{d}) = H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{s},\boldsymbol{Z},\boldsymbol{d}=\mathbf{d}) \overset{(a)}{=} H(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{d}=\mathbf{d}),$$

where in step (a) we used that $\boldsymbol{Y}$ is independent of $\boldsymbol{X}$ and $\boldsymbol{s}$ given $\boldsymbol{Z}$. This allows to compute the entropy by

$$H(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{d}=\mathbf{d}) \overset{(b)}{=} \sum_{i=1}^{M}H(\boldsymbol{y}_i|\boldsymbol{Z},\boldsymbol{d}=\mathbf{d}) \overset{(c)}{=} \sum_{i=1}^{M}H(\boldsymbol{y}_i|\boldsymbol{z}_i,d_{s_i}=\mathtt{d}_i) \overset{(d)}{=} \sum_{i=1}^{M}\mathtt{d}_i H_q(p) = \sum_{d\geq 0}\mathtt{n}_d dH_q(p),$$

where equality (b) follows from the independence of the variables $\boldsymbol{y}_i$ given the input $\boldsymbol{Z}$ and drawing distribution $\boldsymbol{d}$. In equation (c) we used that, given $\boldsymbol{z}_i$ and d_i, the variable $\boldsymbol{y}_i$ is independent of all $\boldsymbol{z}_j$ and d_j with $j \neq i$ due to the fact that $\boldsymbol{y}_i$ can be expressed as the sum of the d_i-fold repetition of $\boldsymbol{z}_i$ and an error vector that is chosen independently, as presented in Section 5.1.1. Note that $H(\boldsymbol{y}_i|\boldsymbol{x}_i, d_i = \mathtt{d}_i) = \mathtt{d}_i H_q(p)$ is precisely the channel entropy of the multinomial channel, which has been shown in Lemma 5.2 to be independent of the input distribution of $\boldsymbol{z}_i$ and thus is only dependent on $\mathtt{d}_i$, which we used in equality (d). □

5.3.3 Permutation Entropy Bound

The last ingredient that is missing to prove Lemma 5.14 is to bound the entropy of the permutation given the input and output sequences. Note that our proof is motivated by the idea of [SH21], where a similar statement has been proven for the case, where the drawing composition is a Bernoulli random variable.

Lemma 5.17. *Fix the parameters $0 < \beta < 1, q \in \mathbb{N}$, $0 < p < 1$. Then for any α with $2p < \alpha < 1$,*

$$H(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{d} = \mathbf{d}) \leq M \log_q M - U_{\mathtt{d}} \log_q U_{\mathtt{d}} + o(ML).$$

Proof. To start with, we observe that

$$H(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{d} = \mathbf{d}) = H(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U}, \boldsymbol{d} = \mathbf{d}), \tag{5.9}$$

as $\mathcal{U}$ is a function of $\boldsymbol{Y}$ and we thus can introduce the condition without changing the entropy. We can further expand the entropy to

$$H(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U}, \boldsymbol{d} = \mathbf{d}) \leq \sum_{\mathtt{u}} \mathsf{Pr}\left(\mathcal{U} = \mathtt{u}|\boldsymbol{d} = \mathbf{d}\right) \sum_{i=1}^{M} H(s_i|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}). \tag{5.10}$$

On the one hand, for each $i \in [M]$ with $\mathtt{d}_i = 0$, we can trivially bound the entropy of the permutation by $H(s_i|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) \leq \log_q M$, as there are at most M options for s_i. On the other hand, we fix an arbitrary $\delta > p$ and introduce for each $i \in [M]$ with $\mathtt{d}_i > 0$, the Bernoulli variable E_i, which is equal to one, if $d_{\mathsf{H}}\left(\boldsymbol{x}_{s_i}, \boldsymbol{y}_i^{(1)}\right) \geq \delta L$ and 0, otherwise. Here $\boldsymbol{y}_i^{(1)} \in \Sigma_q^L$ is the first sequence in the cluster according to the nomenclature of Definition 5.13. As the Hamming distance between $\boldsymbol{x}_{s_i}$ and $\boldsymbol{y}_i^{(1)}$ is binomial distributed with success probability p and L trials, we know from Lemma A.4 that

$$\mathsf{Pr}\left(E_i = 1|\mathbf{d}\right) \leq \mathrm{e}^{-2L(\delta - p)^2} \tag{5.11}$$

This allows to derive the following upper bound on the individual entropy terms.

$$\begin{aligned}
H(s_i|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) &\leq H(s_i, E_i|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) \\
&\leq H(E_i|\boldsymbol{X}, \boldsymbol{Y}, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) + H(s_i|\boldsymbol{X}, \boldsymbol{Y}, E_i, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) \\
&\overset{(a)}{\leq} 1 + \sum_{\mathtt{e}_i \in \{0,1\}} \mathsf{Pr}\left(E_i = \mathtt{e}_i|\mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}\right) H(s_i|\boldsymbol{X}, \boldsymbol{Y}, E_i = \mathtt{e}_i, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) \\
&\overset{(b)}{\leq} 1 + \mathsf{Pr}\left(E_i = 1|\mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}\right) \log_q M + H(s_i|\boldsymbol{X}, \boldsymbol{Y}, E_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}),
\end{aligned} \tag{5.12}$$

where we used in inequality (a) that the entropy of a Bernoulli random variable is at most 1. Inequality (b) follows from the fact that $\mathsf{Pr}\left(E_i = 0 | \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}\right) \leq 1$ and the fact that we can again trivially bound the entropy $H(s_i | \boldsymbol{X}, \boldsymbol{Y}, E_i = 1, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d})$ by $\log_q M$. It remains to bound $H(s_i | \boldsymbol{X}, \boldsymbol{Y}, E_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d})$ from above. To this end, we will set $\delta = \alpha/2$ and for all $i \in \mathtt{u}$, we introduce the sets

$$\mathcal{A}_i = \left\{ j \in [M] : d_{\mathsf{H}}\left(\boldsymbol{x}_j, \boldsymbol{y}_i^{(1)}\right) < \delta L \right\}$$

of input sequences that have distance less than δL to the first sequence in the i-th output cluster. This set contains all input sequences that could potentially have produced $\boldsymbol{y}_i^{(1)}$, given that $E_i = 0$. Note that by definition of E_i and $\mathcal{A}_i$, we directly have $s_i \in \mathcal{A}_i$, given $E_i = 0$. Further, the sets $\mathcal{A}_i$ are disjoint, as for any $i, k \in \mathtt{u}$ and any sequence $j \in \mathcal{A}_i$ it holds by the triangle inequality,

$$d_{\mathsf{H}}\left(\boldsymbol{x}_j, \boldsymbol{y}_k^{(1)}\right) \geq d_{\mathsf{H}}\left(\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_k^{(1)}\right) - d_{\mathsf{H}}\left(\boldsymbol{x}_j, \boldsymbol{y}_i^{(1)}\right) > (\alpha - \delta)L = \delta L,$$

implying that each $j \in [M]$ can be contained in at most one set $\mathcal{A}_i$. For all $i \in \mathtt{u}$ with $E_i = 0$, $s_i \in \mathcal{A}_i$, and s_i can thus assume at most $|\mathcal{A}_i|$ values, limiting its entropy to at most $\log_q |\mathcal{A}_i|$. Bounding the entropy for all other terms $i \notin \mathtt{u}$ by $\log_q M$, we obtain

$$\begin{aligned}
\sum_{i:\mathtt{d}_i > 0} H(s_i | \boldsymbol{X}, \boldsymbol{Y}, E_i = 0, \mathcal{U} = \mathtt{u}, \boldsymbol{d} = \mathbf{d}) &\leq \sum_{i \notin \mathtt{u}:\mathtt{d}_i > 0} \log_q M + \sum_{i \in \mathtt{u}} \log_q |\mathcal{A}_i| \\
&= (M - \mathtt{n}_0 - |\mathtt{u}|) \log M + \sum_{i \in \mathtt{u}} \log_q |\mathcal{A}_i| \overset{(c)}{\leq} (M - \mathtt{n}_0 - |\mathtt{u}|) \log M + |\mathtt{u}| \log_q (M/|\mathtt{u}|) \\
&= (M - \mathtt{n}_0) \log M - |\mathtt{u}| \log |\mathtt{u}|, \qquad (5.13)
\end{aligned}$$

where inequality (c) is due to $\sum_{i \in \mathtt{u}} |\mathcal{A}_i| \leq M$ due to the disjointedness of the sets $\mathcal{A}_i$. Thus, the sum is bounded from above by setting $|\mathcal{A}_i| = M/|\mathtt{u}|$ by Jensen's inequality and the concavity of the logarithm. Plugging (5.13) and (5.12) into (5.10) and taking also those i with $\mathtt{d}_i = 0$ into account, we obtain

$$\begin{aligned}
H(\boldsymbol{s} | \boldsymbol{X}, \boldsymbol{Y}, \mathcal{U}, \boldsymbol{d} = \mathbf{d}) &\leq M \log_q M + \sum_{\mathtt{u}} \mathsf{Pr}\left(\mathtt{u} | \mathbf{d}\right) \left(-|\mathtt{u}| \log_q |\mathtt{u}| + \sum_{i:\mathtt{d}_i > 0} \left(1 + \mathsf{Pr}\left(E_i = 1 | \mathbf{d}, \mathtt{u}\right) \log_q M\right) \right) \\
&\leq M \log_q M - \mathsf{E}\left[|\mathcal{U}| \log_q |\mathcal{U}| \;\middle|\; \boldsymbol{d} = \mathbf{d} \right] + \log_q M \sum_{i:\mathtt{d}_i > 0}^{M} \mathsf{Pr}\left(E_i = 1 | \mathbf{d}\right) + O(M) \\
&\overset{(d)}{=} M \log_q M - U_{\mathtt{d}} \log_q U_{\mathtt{d}} + o(ML), \qquad (5.14)
\end{aligned}$$

where we used Jensen's inequality in (d) together with the bound (5.11) on the probability $\mathsf{Pr}\left(E_i = 1 | \mathbf{d}\right)$, which proves the claim of the lemma with (5.9). □

5.4 Achievable Rates

We proceed by deriving achievable rates for the unordered parallel multinomial channel. We derive these results using standard random coding techniques [Sha48]. For a thorough introduction into

this area, we recommend the reader to familiarize with [CT06]. The basic idea is as follows. Choose a random codebook $\mathcal{C}$ of rate R with identically and independent distributed codewords, which are drawn from some given input distribution $\mathsf{Pr}(\boldsymbol{X} = \mathbf{X})$. Then, define some generic, easy-to-analyze decoder and proceed with computing the average error probability over all codebooks. If one can prove that for a given rate R, the average error probability tends to zero, we can deduce that there exists at least one codebook of rate R whose error probability also tends to zero. This is because the infimum of an ensemble is always bounded from above by its average.

In our case we will use a decoder that is based on jointly typical sequences [CT06; Mac15] over the multinomial channel. It is known that for any discrete memoryless channel, and thus also for the multinomial channel, the input and output sequences are jointly typical with respect to the input distribution with high probability. Conversely, another input sequence, chosen independently from the same input distribution is jointly typical with the previous output sequence with very small probability. We will use this fact and define a new kind of typicality over the unordered multinomial channel as follows. Roughly speaking, we will say that the M input sequences $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ are jointly typical with the M output clusters $\boldsymbol{y}_1, \dots, \boldsymbol{y}_M$, if there exists a matching between the input sequences and output clusters, whose size is large, i.e., close to M. We hereby match an input sequence $\boldsymbol{x}_i$ and an output cluster $\boldsymbol{y}_j$, if they are jointly typical with respect to the multinomial channel. Note that the output clusters, which are empty can be matched to any input sequence, as they are jointly typical with respect to any input sequence. Given a channel output $\boldsymbol{y}_1, \dots, \boldsymbol{y}_M$, the decoder then decides for a codeword if it is jointly typical in the above sense with a unique codeword and fails in any other case. Analyzing this decoder will show that for any rate R below the capacity of the unordered parallel multinomial channel, the probability that the correct transmitted codeword is jointly typical with the received word with high probability and any other codeword is jointly typical with small probability. We now turn to a rigorous derivation of achievable rates. We will devote the rest of this section to prove the following result about achievable rates in a step-by-step fashion. We will proceed by presenting the final results first and wrap up the necessary ingredients towards the end of the section.

Lemma 5.18. *Fix $\beta > 0, q \in \mathbb{N}, 0 < p < \frac{q-1}{2q}$, and let the distribution $\mathsf{Pr}(\boldsymbol{d})$ be a regular distribution that converges in frequency to $\boldsymbol{\nu}$. Then, any rate R with*

$$R < C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q).$$

is achievable over the unordered parallel multinomial channel.

We start with setting up the necessary definitions required for the following expositions and assume that the conditions of Lemma 5.18 are fulfilled throughout the remainder of this section. We will prove the results using the conventional random coding argument. To this end recall the communication setup presented in Section 5.2.1. Let now $\mathcal{C} = \{\boldsymbol{X}(1), \dots, \boldsymbol{X}(q^{MLR})\} \subseteq \Sigma_q^{M \times L}$ be a randomly chosen codebook of code rate R, where each codeword $\boldsymbol{X}(i) \in \Sigma_q^{M \times L}$ is selected independently and uniform over all possible words in $\Sigma_q^{M \times L}$, i.e., each symbol in $\boldsymbol{X}(i)$ is chosen independently and uniformly over Σ_q. We will write $\boldsymbol{X}(i) = (\boldsymbol{x}_1(i), \dots, \boldsymbol{x}_M(i))$. In order to define the decoder, we fix an $0 < \epsilon < 1$ and will use the notion of jointly typical sequences as described in the following definition. To this end, recall Definition 5.3 of jointly typical sequences over the multinomial channel.

Definition 5.19. *We define the largest typicality matching $T^{\epsilon}_{\mathsf{UPM}}(\mathbf{X}, \mathbf{Y})$ between the input $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$ and output $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$ as the largest integer T such that there exist two*

sequences of integers $i_1, \dots, i_T$ *and* $j_1, \dots, j_T$*, with* $i_t, j_t \in [M]$ *for all* $1 \le t \le T$*, each sequence composed of distinct integers, such that* $(\boldsymbol{x}_{i_t}, \boldsymbol{y}_{j_t}) \in \mathcal{T}^{L,\epsilon}_{\mathsf{Mul}}(d_{j_t}, p, q)$ *for all* $1 \le t \le T$*, where* d_{j_t} *is the size of the cluster* $\boldsymbol{y}_{j_t}$*. We further define the set of jointly typical sequences over the unordered parallel multinomial channel as*

$$\mathcal{T}^{M,L,\epsilon}_{\mathsf{UPM}}(p,q) = \left\{ \begin{array}{l} ((\mathbf{x}_1, \dots, \mathbf{x}_M), (\mathbf{y}_1, \dots, \mathbf{y}_M)) \in \Sigma_q^{M \times L} \times \left(\Sigma_q^{d_1 \times L} \times \cdots \times \Sigma_q^{d_M \times L}\right) : \\ \quad d_1, \dots, d_M \in \mathbb{N}_0 \wedge T^{\epsilon}_{\mathsf{UPM}}((\mathbf{x}_1, \dots, \mathbf{x}_M), (\mathbf{y}_1, \dots, \mathbf{y}_M)) \ge (1-\epsilon)M \end{array} \right\}.$$

In other words, an input $\boldsymbol{X}$ and output $\boldsymbol{Y}$ is jointly typical over the unordered parallel multinomial channel, if there exist many (distinct) pairs of input sequences and output clusters $(\boldsymbol{x}_i, \boldsymbol{y}_j)$ that are jointly typical with respect to the multinomial channel. The decoder $\mathsf{dec}_{\mathcal{C}}(\boldsymbol{Y})$ decodes to $\widehat{W}$, if $\boldsymbol{X}(\widehat{W})$ is the unique codeword that is jointly typical with $\boldsymbol{Y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_M)$ with respect to the unordered multinomial channel as in Definition 5.19, i.e., $(\boldsymbol{X}(\widehat{W}), \boldsymbol{Y}) \in \mathcal{T}^{M,L,\epsilon}_{\mathsf{UPM}}(p,q)$. If there is none or more than two codewords that are jointly typical with $\boldsymbol{Y}$, than the decoder outputs a failure, resulting in a decoding error. The following event will be helpful throughout this section. Denote by $\mathcal{J}_i$ the event that the i-th codeword $\boldsymbol{X}(i)$ is jointly typical with $\boldsymbol{Y}$, i.e., $(\boldsymbol{X}(i), \boldsymbol{Y}) \in \mathcal{T}^{M,L,\epsilon}_{\mathsf{UPM}}(p,q)$ and by $\mathcal{J}_i^{\mathsf{c}}$ the complement event.

Proof of Lemma 5.18. The average probability of a decoding error, averaged over all codebooks, is given by

$$\mathsf{Pr}\left(\mathsf{Err}\right) = \sum_{\mathcal{C}} \mathsf{Pr}\left(\mathcal{C}\right) \mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}\right) = \mathsf{Pr}\left(\mathsf{Err}|W=1\right),$$

where the last equality is due to the symmetry of the choice of random codebooks, see, e.g., [CT06, Ch. 7.7]. The two possible error events are that either $\boldsymbol{X}(1)$ is not jointly typical with $\boldsymbol{Y}$ or that one of the other codewords is jointly typical with respect to $\boldsymbol{Y}$. By the union bound we obtain

$$\mathsf{Pr}\left(\mathsf{Err}|W=1\right) \le \mathsf{Pr}\left(\mathcal{J}_1^{\mathsf{c}} \cup \bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right) \le \mathsf{Pr}\left(\mathcal{J}_1^{\mathsf{c}}|W=1\right) + \mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right).$$

By Lemmas 5.20 and 5.21, for any $0 < \epsilon < 1$ and $R < C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q) - 5\epsilon$ the probability of both the error events above converges to zero as $M \to \infty$. Since we can choose ϵ arbitrarily small, it follows that for each $R < C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q)$, the error probability $\mathsf{Pr}\left(\mathsf{Err}|W=1\right) \to 0$ vanishes as $M \to \infty$. Since the average error probability over all codebooks vanishes, for all code rates $R < C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q)$, there exists at least one codebook of rate R that has vanishing error rate and thus R is an achievable rate. □

We proceed with bounding the error probability of both error events.

5.4.1 Decoding Probability for the Correct Codeword

We start with bounding the probability that the correct codeword is not within the decoding radius. The following lemma proves that the transmitted codeword is jointly typical to its corresponding output with high probability.

Lemma 5.20. *Fix the parameters* $0 < \beta < 1, q \in \mathbb{N}$*,* $0 < p < 1$ *and let the distribution* $\mathsf{Pr}\left(\boldsymbol{d}\right)$ *be a regular distribution. Then, for any* $0 < \epsilon < 1$*, the probability* $\mathsf{Pr}\left(\mathcal{J}_1|W=1\right)$ *that the transmitted codeword is jointly typical with the channel output satisfies* $\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) \to 1$*, as* $M \to \infty$*.*

Proof. We bound $\mathsf{Pr}\left(\mathcal{J}_1|W=1\right)$ from below. The proof follows a similar outline as the second part of the proof of Lemma A.7. For an arbitrary $\epsilon > 0$ denote by $\mathcal{N}_\epsilon$ the event on the random variable $\boldsymbol{n}$ that $\sum_{d\geq 0}\left|\frac{\mathtt{n}_d}{M}-\nu_d\right| \leq \epsilon/4$. We demarginalize with respect to the drawing composition $\boldsymbol{d}$ and obtain

$$\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) = \sum_{\mathbf{d}} \mathsf{Pr}\left(\mathbf{d}\right)\mathsf{Pr}\left(\mathcal{J}_1|W=1,\boldsymbol{d}=\mathbf{d}\right) \geq \sum_{\boldsymbol{d}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{d}\right)\mathsf{Pr}\left(\mathcal{J}_1|W=1,\boldsymbol{d}=\mathbf{d}\right),$$

where we used that the drawing composition $\boldsymbol{d}$ is independent of the message W. Note that the event $\mathcal{N}_\epsilon$ is defined as an event on the drawing frequency $\boldsymbol{n}$, however since $\boldsymbol{n}$ is a function of $\boldsymbol{d}$, one can also view it as an event on the drawing composition $\boldsymbol{d}$. Denote by $\boldsymbol{Z}(1)=(\boldsymbol{z}_1(1),\ldots,\boldsymbol{z}_M(1))$ the permuted input sequences. We will analyze the number

$$T^{\epsilon}_{\mathsf{OPM}}(\boldsymbol{Z}(1),\boldsymbol{Y}) \triangleq \left|\left\{i\in[M]:(\boldsymbol{z}_i(1),\boldsymbol{y}_i)\in\mathcal{T}^{L,\epsilon}_{\mathsf{Mul}}(\mathtt{d}_i,p,q)\right\}\right|$$

of jointly typical pairs over the multinomial channel. This is because the unordered typicality is at least as large as the ordered typicality, i.e., $T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}(1),\boldsymbol{Y}) \geq T^{\epsilon}_{\mathsf{OPM}}(\boldsymbol{Z}(1),\boldsymbol{Y})$ due to the fact that $\boldsymbol{Z}(1)$ is simply a permutation of $\boldsymbol{X}$ and we thus can match $\boldsymbol{y}_i$ to $\boldsymbol{x}_{s_i}(1)$ for all pairs of sequences that contribute to $T^{\epsilon}_{\mathsf{OPM}}(\boldsymbol{Z}(1),\boldsymbol{Y})$. Note that this matching could also be larger, however this bound is sufficient for our analysis as this implies that

$$\begin{aligned}\mathsf{Pr}\left(\mathcal{J}_1|W=1,\boldsymbol{d}=\mathbf{d}\right) &= \mathsf{Pr}\left(T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}(1),\boldsymbol{Y})\geq(1-\epsilon)M|W=1,\boldsymbol{d}=\mathbf{d}\right)\\ &\geq \mathsf{Pr}\left(T^{\epsilon}_{\mathsf{OPM}}(\boldsymbol{Z}(1),\boldsymbol{Y})\geq(1-\epsilon)M|W=1,\boldsymbol{d}=\mathbf{d}\right).\end{aligned}$$

For a given number of draws $\boldsymbol{d}=\mathbf{d}$ the size of the largest typical matching $T^{\epsilon}_{\mathsf{OPM}}(\boldsymbol{Z}(1),\boldsymbol{Y})$ is the sum of M independent random Bernoulli random variables with success probabilities $\pi_i \triangleq \mathsf{Pr}((\boldsymbol{z}_i(1),\boldsymbol{y}_i)\in\mathcal{T}^{L,\epsilon}_{\mathsf{Mul}}(\mathtt{d}_i,p,q)|W=1,d_i=\mathtt{d}_i)$. From the results about jointly typical sequences [CT06, Thm. 7.6.1] we know that for all $\epsilon>0$ and $i\in[M]$, it holds that $\pi_i > 1-\epsilon/2$ for all $L\geq L_{\mathtt{d}_i}$, as $\boldsymbol{y}_i$ is the result of transmitting $\boldsymbol{z}_i(1)$ over the multinomial channel. As $\max_{i\in[M]} L_{\mathtt{d}_i}$ might increase with M, we focus our attention to a subset of multinomial channels whose number of draws is bounded from above by a large, but finite quantity. To this end, let D_ϵ be the smallest integer such that $\sum_{d\geq D_\epsilon}\nu_d < \epsilon/4$. We have that for all $\mathbf{d}\in\mathcal{N}_\epsilon$, the number of positions $i\in[M]$ with $\mathtt{d}_i < D_\epsilon$ is at least

$$\sum_{d=0}^{D_\epsilon-1}\mathtt{n}_d \geq M\sum_{d=0}^{D_\epsilon-1}\nu_d - \frac{M\epsilon}{4} > M\left(1-\frac{\epsilon}{2}\right).$$

Thus, at least $M(1-\epsilon/2)$ Bernoulli variables have success probability at least $\pi_i > 1-\epsilon/2$ for all $L\geq\max_{0\leq d<D_\epsilon}L_d$ (which is finite) and we obtain

$$\begin{aligned}&\mathsf{Pr}\left(\mathcal{J}_1|W=1,\boldsymbol{d}=\mathbf{d}\right)\\ &\geq \sum_{i=M-M\epsilon}^{M-\frac{M\epsilon}{2}}\binom{M-\frac{M\epsilon}{2}}{i}\left(1-\frac{\epsilon}{2}\right)^i\left(\frac{\epsilon}{2}\right)^{M-\frac{M\epsilon}{2}-i} = \sum_{i=0}^{\frac{M\epsilon}{2}}\binom{M-\frac{M\epsilon}{2}}{i}\left(1-\frac{\epsilon}{2}\right)^{M-\frac{M\epsilon}{2}-i}\left(\frac{\epsilon}{2}\right)^i\\ &= 1-\sum_{i=\frac{M\epsilon}{2}+1}^{M-\frac{M\epsilon}{2}}\binom{M-\frac{M\epsilon}{2}}{i}\left(1-\frac{\epsilon}{2}\right)^{M-\frac{M\epsilon}{2}-i}\left(\frac{\epsilon}{2}\right)^i \overset{(a)}{\geq} 1-\mathrm{e}^{-2\left(M-\frac{M\epsilon}{2}\right)\left(\frac{\epsilon^2}{4-2\epsilon}\right)^2},\end{aligned}$$

for all $0 < \epsilon < 1$ and large enough L. Here we used Lemma A.4 to bound the binomial tail in inequality (a). Thus, finally, for any $\epsilon > 0$ and large enough L,

$$\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) \geq \left(1 - \mathrm{e}^{-2\left(M-\frac{M\epsilon}{2}\right)\left(\frac{\epsilon^2}{4-2\epsilon}\right)^2}\right) \mathsf{Pr}\left(\boldsymbol{d} \in \mathcal{N}_\epsilon\right),$$

where the first term approach 1 as $M \to \infty$ for any $1 < \epsilon < 0$ and the second term approaches 1 as well by assumption of convergence of the drawing frequency. It follows that $\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) \to 1$. □

5.4.2 Decoding Probability for the Wrong Codewords

The next lemma proves that the probability that any other codeword is jointly typical with respect to the output that stems from the correct codeword is small.

Lemma 5.21. *Fix the parameters $0 < \beta < 1, q \in \mathbb{N}$, $0 < p < 1$ and let the distribution $\mathsf{Pr}\left(\boldsymbol{d}\right)$ be a given regular distribution that converges in frequency to $\boldsymbol{\nu}$. Then, for any $0 < \epsilon < 1$, and any rate $R < C_{\mathsf{UPM}}(\boldsymbol{\nu}, \beta, p, q) - 5\epsilon$, the probability that at least one other codeword $\boldsymbol{X}(i)$, $2 \leq i \leq q^{MLR}$ is jointly typical with the channel output of the transmission of $\boldsymbol{X}(1)$ over the unordered parallel multinomial channel satisfies*

$$\mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right) \to 0,$$

as $M \to \infty$.

Proof. We again denote by $\mathcal{N}_\epsilon$ the event for $\boldsymbol{n}$ that $\sum_{d\geq 0}\left|\frac{\mathtt{n}_d}{M} - \nu_d\right| \leq \epsilon/4$. Similar as in the proof of Lemma 5.20 we demarginalize with respect to the drawing composition $\boldsymbol{d}$ and obtain

$$\begin{aligned}\mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right) &\leq \mathsf{Pr}\left(\boldsymbol{d} \notin \mathcal{N}_\epsilon\right) + \sum_{\mathbf{d}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1, \boldsymbol{d}=\mathbf{d}\right) \\ &\overset{(a)}{\leq} \mathsf{Pr}\left(\boldsymbol{d} \notin \mathcal{N}_\epsilon\right) + q^{MLR} \sum_{\mathbf{d}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\mathcal{J}_2|W=1, \boldsymbol{d}=\mathbf{d}\right), \end{aligned} \tag{5.15}$$

where in inequality (a) we used the union bound together with the fact that $\mathsf{Pr}\left(\mathcal{J}_i|W=1,\boldsymbol{d}=\mathbf{d}\right)$ is invariant over all $2 \leq i \leq q^{MLR}$ due to the identical and independent choice of codewords. To start with, denote $\boldsymbol{Z}(2) = (\boldsymbol{z}_1(2), \ldots, \boldsymbol{z}_M(2))$ as the random variable, which is the permutation of the codeword $\boldsymbol{X}(2)$. We observe that $T^\epsilon_{\mathsf{UPM}}(\mathbf{X}(2), \mathbf{Y}) = T^\epsilon_{\mathsf{UPM}}(\mathbf{Z}(2), \mathbf{Y})$ since the size of the largest matching is invariant to permutations of the sequences and thus

$$\mathsf{Pr}\left(\mathcal{J}_2|W=1,\boldsymbol{d}=\mathbf{d}\right) = \mathsf{Pr}\left(T^\epsilon_{\mathsf{UPM}}(\boldsymbol{Z}(2), \boldsymbol{Y}) \geq M(1-\epsilon)|W=1,\boldsymbol{d}=\mathbf{d}\right).$$

For an arbitrary $n \in [M]$ denote by $\mathcal{P}(M,n) = \{\boldsymbol{m} = (m_1, \ldots, m_n) \in [M]^n : m_i \neq m_j \ \forall\ i \neq j\}$ the set of length-n partial permutations of the set $[M]$. Denote further by $T_{m,j}$ a Bernoulli random variable, which is equal to 1, if $(\boldsymbol{z}_m(2), \boldsymbol{y}_j) \in \mathcal{T}^{L,\epsilon}_{\mathsf{Mul}}(\mathtt{d}_j, p, q)$ and 0, otherwise. This allows to rewrite

the above probability as

$$\begin{aligned}\mathsf{Pr}\left(\mathcal{J}_2|W=1,\boldsymbol{d}=\mathbf{d}\right) &= \mathsf{Pr}\left(\exists \boldsymbol{m}\in\mathcal{P}(M,M):\sum_{j=1}^{M} T_{m_j,j}\geq M(1-\epsilon)\middle|W=1,\boldsymbol{d}=\mathbf{d}\right)\\ &\overset{(b)}{=} \mathsf{Pr}\left(\exists \boldsymbol{m}\in\mathcal{P}(M,M):\sum_{j:\mathtt{d}_j>0} T_{m_j,j}\geq M(1-\epsilon)-\mathtt{n}_0\middle|W=1,\boldsymbol{d}=\mathbf{d}\right),\end{aligned}$$

where (b) is due to the fact that $T_{m,j}=1$ with probability 1 for all empty clusters, i.e., for all $j\in[M]$ with $\mathtt{d}_j=0$. Denote by $j_1,\ldots,j_{M-\mathtt{n}_0}$ precisely those indices with $\mathtt{d}_{j_t}>0$ for all $1\leq t\leq M-\mathtt{n}_0$. Then, we can simplify the above expression by

$$\begin{aligned}\mathsf{Pr}\left(\mathcal{J}_2|W=1,\boldsymbol{d}=\mathbf{d}\right) &= \mathsf{Pr}\left(\exists \boldsymbol{m}'\in\mathcal{P}(M,M-\mathtt{n}_0):\sum_{t=1}^{M-\mathtt{n}_0} T_{m'_t,j_t}\geq M(1-\epsilon)-\mathtt{n}_0\middle|W=1,\boldsymbol{d}=\mathbf{d}\right)\\ &\overset{(c)}{\leq} \sum_{\boldsymbol{m}'\in\mathcal{P}(M,M-\mathtt{n}_0)} \mathsf{Pr}\left(\sum_{t=1}^{M-\mathtt{n}_0} T_{m'_t,j_t}\geq M(1-\epsilon)-\mathtt{n}_0\middle|W=1,\boldsymbol{d}=\mathbf{d}\right),\end{aligned}$$

where inequality (c) is due to an application of the union bound. We will bound the above probability as follows. To start with, since $\boldsymbol{X}(2)$ is chosen independently from $\boldsymbol{Y}$, also $\boldsymbol{Z}(2)$ is independent of $\boldsymbol{Y}$ and it follows that, given $\boldsymbol{d}=\mathbf{d}$, for all $i,j\in[M]$, $\pi_j\triangleq\mathsf{Pr}\left(T_{i,j}=1|W=1,\boldsymbol{d}=\mathbf{d}\right)$, it holds that $\pi_j<q^{-L(C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)-\epsilon)}$ for $L\geq L_{\mathtt{d}_j}$ [CT06, Thm. 7.6.1], where $L_{\mathtt{d}_j}$ are integers that depend on ϵ and the channel $\mathtt{d}_j$. Since at least $M(1-\epsilon)-\mathtt{n}_0$ of the Bernoulli variables $T_{m'_t,j_t}$ must be equal to 1, we can use these definitions to bound the above probability

$$\begin{aligned}&\mathsf{Pr}\left(\sum_{t=1}^{M-\mathtt{n}_0} T_{m'_t,j_t}\geq M(1-\epsilon)-\mathtt{n}_0\middle|W=1,\boldsymbol{d}=\mathbf{d}\right)\leq \sum_{\mathcal{I}\subseteq[M-\mathtt{n}_0]:|\mathcal{I}|=M(1-\epsilon)-\mathtt{n}_0}\prod_{t\in\mathcal{I}}\pi_{j_t}\\ &\overset{(d)}{\leq}\sum_{\mathcal{I}\subseteq[M]:|\mathcal{I}|=M(1-\epsilon)}\prod_{j\in\mathcal{I}}\pi_j\leq\binom{M}{M(1-\epsilon)}\max_{\mathcal{I}\subseteq[M]:|\mathcal{I}|=M(1-\epsilon)}\prod_{j\in\mathcal{I}}\pi_j.\end{aligned}$$

Note that in inequality (d) we factored those j into account with $\mathtt{d}_j=0$ into the product, for which $\pi_j=1$. This will simplify the subsequent notation and analysis. Mathematically, inequality (d) holds, as each set $\mathcal{I}_1$ in the first sum is contained in some set $\mathcal{I}_2$ of the second sum such that the positions $j\in\mathcal{I}_2\setminus\mathcal{I}_1$ are exactly those positions with $\mathtt{d}_j=0$ and $\pi_j=1$ and thus each term in the first sum is accounted for by at least one term in the second sum. In order to use the above bound on π_j, we restrict our attention to those channels j with at most a finite but large number of draws, such that the maximum over $L_{\mathtt{d}_j}$ is guaranteed to be constant in M. To this end, let D_ϵ be the smallest integer such that $\sum_{d\geq D_\epsilon}\nu_d<\epsilon/4$ and abbreviate $\mathcal{D}(\epsilon)=\{j\in[M]:\mathtt{d}_j<D_\epsilon\}$. We can then bound the product over π_j to

$$\prod_{j\in\mathcal{I}}\pi_j\leq\prod_{j\in\mathcal{I}\cap\mathcal{D}(\epsilon)}\pi_j<\prod_{j\in\mathcal{I}\cap\mathcal{D}(\epsilon)}q^{-L(C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)-\epsilon)}=q^{-L\sum_{j\in\mathcal{I}\cap\mathcal{D}(\epsilon)}(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon)}$$

for all $L\geq\max_{0\leq d<D_\epsilon}L_d$, which is constant and not a function of M or L, as desired. Analyzing

the exponent of the error probability expression above, we find that

$$\begin{aligned}\sum_{j\in\mathcal{I}\cap\mathcal{D}(\epsilon)}(C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)-\epsilon) &= \left(\sum_{j=1}^{M}(C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)-\epsilon)-\sum_{j\notin\mathcal{I}\cap\mathcal{D}(\epsilon)}(C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)-\epsilon)\right)\\ &\overset{(e)}{\geq}\left(\sum_{j=1}^{M}(C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)-\epsilon)-\frac{3M\epsilon}{2}\right)=\sum_{d\geq 0}\mathtt{n}_d C_{\mathsf{Mul}}(d,p,q)-\frac{5M\epsilon}{2}\\ &\geq M\sum_{d\geq 0}\nu_d C_{\mathsf{Mul}}(d,p,q)-\frac{7M\epsilon}{2},\end{aligned}$$

where in inequality (e) we bounded the second sum using $C_{\mathsf{Mul}}(\mathtt{d}_j,p,q)\leq 1$ together with the fact that by definition of D_ϵ and for all $\mathbf{d}\in\mathcal{N}_\epsilon$,

$$|\mathcal{D}(\epsilon)|=\sum_{d=0}^{D_\epsilon-1}\mathtt{n}_d\geq M\sum_{d=0}^{D_\epsilon-1}\nu_d-\frac{M\epsilon}{4}>M\left(1-\frac{\epsilon}{2}\right),$$

and thus

$$|\{j\in[M]:j\notin\mathcal{I}\cap\mathcal{D}(\epsilon)\}|\leq M-|\mathcal{I}|+M-|\mathcal{D}(\epsilon)|<\frac{3M\epsilon}{2},$$

where we used that $|\mathcal{I}|=M(1-\epsilon)$ for the considered sets. For any $0<\epsilon<1$ and large enough M, and any $\mathbf{d}\in\mathcal{N}_\epsilon$, the resulting upper bound $\mathsf{Pr}\left(\mathcal{J}_2|W=1,\boldsymbol{d}=\mathbf{d}\right)$ is henceforth

$$\begin{aligned}\mathsf{Pr}\left(\mathcal{J}_2|W=1,\boldsymbol{d}=\mathbf{d}\right)&\leq|\mathcal{P}(M,M-\mathtt{n}_0)|\binom{M}{M(1-\epsilon)}q^{-ML(\sum_{d\geq 0}\nu_d C_{\mathsf{Mul}}(d,p,q)-7\epsilon/2)}\\ &\overset{(f)}{\leq}2^M q^{-ML(\sum_{d\geq 0}\nu_d C_{\mathsf{Mul}}(d,p,q)-\beta(1-\nu_0)-9\epsilon/2)},\end{aligned}\tag{5.16}$$

where we used $\binom{M}{M(1-\epsilon)}\leq 2^M$ and $|\mathcal{P}(M,M-\mathtt{n}_0)|=\frac{M!}{\mathtt{n}_0!}\leq M^{M-\mathtt{n}_0}\leq q^{\beta LM(1-\nu_0+\epsilon)}$ for all $\mathbf{d}\in\mathcal{N}_\epsilon$ in inequality (f). Plugging (5.16) into the average code error probability (5.15), we obtain

$$\mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}}\mathcal{J}_i\middle|W=1\right)\leq\mathsf{Pr}\left(\boldsymbol{d}\notin\mathcal{N}_\epsilon\right)+q^{-ML(-R+C_{\mathsf{UPM}}(\boldsymbol{\nu},p,q)-9\epsilon/2-\log_q(2)/L)}.$$

In the above expression, it holds that $\mathsf{Pr}\left(\boldsymbol{d}\notin\mathcal{N}_\epsilon\right)\to 0$ as $M\to\infty$ by assumption of a drawing distribution that converges in frequency. Further, $\log_q(2)/L\to 0$ as $M\to\infty$ and thus, fur any $R<C_{\mathsf{UPM}}(\boldsymbol{\nu},p,q)-9\epsilon/2$, the exponent of q will be negative for large enough M and the sought-after error probability converges to 0 as $M\to\infty$, which proves the claim of the lemma. □

5.5 Conclusion

The main topic of this chapter was a probabilistic channel whose input comprises many parallel sequences and the output is obtained by shuffling these sequences and transmitting each individual sequence over a multinomial channel with a random number of draws. We have presented conditions on the drawing distribution and the channel parameters that allowed to derive an explicit closed form expression for the channel capacity. We have presented the capacity formula

and given an intuitive interpretation of the involved terms together with a rigorous proof of a converse and an achievablity bound.

The converse bound for the unordered parallel multinomial channel has, as in other works on permutation channels [Len+20d; SH19; SH21], been derived using an auxiliary statistic that characterizes the similarity of the sequences. As the presented statistic based on mutual Hamming distances only enables us to derive the capacity for a limited number of parameters, one intriguing question is, whether other measures of similarity allow to derive converse bounds for a larger range of parameters. Further, while this technique allows an elegant, tractable analysis of the channel statistics, a natural open question is also whether other techniques can be used to derive converse bounds. On the other hand, achievable rates have been derived using standard random coding techniques together with a novel measure of typicality over parallel channels. It appears that the technique presented here naturally generalizes to other constituent channels, which is however out of scope of this dissertation.

There are other possible generalizations of this work, which are interesting and non-trivial. One is the generalization to asymmetric channels instead of the symmetric channel discussed here. This would likely require a sharper, or possible soft metric, as the Hamming distance might not be the appropriate measure due to the asymmetry of the channel. Another interesting direction is the incorporation of insertions and deletions into the channel model. Notice however that this direction is particularly challenging due to the fact that even for standalone insertion and deletion channels, an explicit capacity formula remains unknown to date.

Chapter 6

Probabilistic DNA Storage Channel

Consider a DNA-based data storage system, where information is stored on many short DNA sequences, which are transferred into a liquid storage medium. One of the most striking properties of such systems is the unordered nature in which DNA strands reside in the storage medium. Together with the fact that usually each strand is contained several times in the medium, these systems are the basis for a unique communication model in which the received sequences are erroneous samples of the original sequences and the origin of the sequences is unknown to the receiver. The first information-theoretic studies of such a communication system have been conducted in [Hec+17; MSG15]. Based on a sequence sampling model, where the output sequences are obtained through independent and uniform draws from the input sequences, the capacity was derived for the case of no errors within the sequences. Other works [SH19; SH21] have derived the capacity for noisy sequences, however with a different sampling model that either assumes that each sequence is drawn exactly once [SH19] or according to a Bernoulli distribution [SH21]. Building on these works, we generalize the results of [Hec+17; MSG15] for the independent and uniform drawing model to the case, where errors can occur in the sequences.

This chapter is organized as follows. We start with a precise definition of the probabilistic DNA storage channel in Section 6.1. Next are the definitions and our result on the channel capacity in Section 6.2. For the derivation of the capacity, we show in Section 6.3 that the probabilistic DNA storage channel is equivalent to a degraded unordered parallel multinomial channel. We use this relationship to derive an upper bound on all achievable information rates in Section 6.4. This bound is complemented with a result that proves the attainability of all information rates below the converse bound in Section 6.5. We conclude with numerical evaluations of the presented capacity formulae and a discussion on cost-efficient system design in Section 6.6.

The probabilistic DNA storage channel is closely related to the unordered parallel multinomial channel, that has been presented in Chapter 5. While this chapter is intended to be self-contained, we will use results from Chapter 5 and adopt notation in some places. We therefore advise the interested reader to familiarize with the contents of Chapter 5 for an in-depth understanding of the presented materials in the following.

Parts of the results within this chapter have been published in [Len+19a; Len+20c; Len+21f].

6.1 Channel Model

The input of the DNA storage channel is M sequences $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ where each $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,L})$, $\boldsymbol{x}_i \in \Sigma_q^L$, $i \in [M]$, is a vector of length L over the q-ary alphabet Σ_q. From these input sequences, a total of N sequences are drawn with replacement, each uniformly and independent of other draws, and received with errors. Denote by $\boldsymbol{I} = (I_1, \dots, I_N), I_j \in [M]$ the *drawing indices*, i.e. in the j-th draw, the input sequence $\boldsymbol{x}_{I_j}$ has been drawn. The realization of $\boldsymbol{I}$ and I_j are denoted by $\mathbf{I}$ and $\mathtt{I}_j$, respectively. We assume that the draws I_j are i.i.d. uniform random variables with $\mathsf{Pr}(I_j = i) = \frac{1}{M}$ for all $j \in [N]$ and $i \in [M]$ that are independent from the input $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$. The output of the channel are N sequences $\boldsymbol{r}_j = (r_{j,1}, \dots, r_{j,L}) \in \Sigma_q^L$, $j \in [N]$, each of length L. Each sequence $\boldsymbol{r}_j$ is obtained by drawing a random input sequence $\boldsymbol{x}_{I_j}$ and transmitting it over a q-ary symmetric channel with error probability p. That is, the output sequences are given by

$$\boldsymbol{r}_j = \boldsymbol{x}_{I_j} + \boldsymbol{e}_j,$$

for all $j \in [N]$, where the sum is performed over the finite Abelian group of integers, i.e., modulo q. Hereby, $\boldsymbol{e}_j = (e_{j,1}, \dots, e_{j,L})$ are random error vectors with i.i.d. entries

$$\mathsf{Pr}(e_{j,\ell} = \mathsf{e}_{j,\ell}) = \begin{cases} 1-p, & \text{if } \mathsf{e}_{j,\ell} = 0 \\ \frac{p}{q-1}, & \text{if } \mathsf{e}_{j,\ell} \neq 0 \end{cases}.$$

for all $j \in [N]$ and $\ell \in [L]$ that are independent of the input $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$. Hence, the overall input-output relationship can be summarized as

$$\begin{aligned} \mathsf{Pr}(\mathbf{r}_1, \dots, \mathbf{r}_N | \mathbf{x}_1, \dots, \mathbf{x}_M) &= \sum_{\mathbf{I}} \mathsf{Pr}(\mathbf{I}) \, \mathsf{Pr}(\mathbf{r}_1, \dots, \mathbf{r}_N | \mathbf{x}_1, \dots, \mathbf{x}_M, \mathbf{I}) \\ &= \sum_{\mathbf{I}} \mathsf{Pr}(\mathbf{I}) \prod_{j=1}^{N} \mathsf{Pr}\left(\mathbf{r}_j, | \mathbf{x}_{\mathtt{I}_j}\right), \end{aligned}$$

where $\mathsf{Pr}\left(\mathbf{r}_j, | \mathbf{x}_{\mathtt{I}_j}\right)$ is according to the q-ary symmetric channel described above. For notational convenience, we stack all input and output sequences to matrices $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_M) \in \Sigma_q^{M \times L}$ and $\boldsymbol{R} = (\boldsymbol{r}_1, \dots, \boldsymbol{r}_N) \in \Sigma_q^{N \times L}$, such that each sequence is a row of the corresponding matrix. Here we choose to define the input and output sequences as matrices of sizes $M \times L$, and $N \times L$ respectively, for notional convenience. However, it can directly be verified that by defining the input and output as multi-sets of M, respectively N vectors, each of length L, one obtains an equivalent channel. Figure 6.1 illustrates an exemplary realization of this channel.

Throughout the section, we will adopt the notation from Chapter 5 and use the following random variables. The *drawing composition* $\boldsymbol{d} = (d_1, \dots, d_M)$ with $d_i = |\{j \in [N] : I_j = i\}|$, $i \in [M]$, which counts the number of times the i-th input sequence has been drawn and the *drawing frequency* $\boldsymbol{n} = (n_0, n_1, \dots)$ with $n_d = |\{i \in [M] : d_i = d\}|$, $d = 0, \dots, N$, which denotes the number of input sequences that have been drawn a total of d times.

6.2 Capacity of the Probabilistic DNA Storage Channel

The term *capacity* goes back to the seminal work of Shannon [Sha48] and states a limit on the maximal information rate at which it is possible to reliably communicate over a communication channel. We rigorously define the notions of information rate and reliability and state the main theorem on the channel capacity, which will be proven in the subsequent sections.

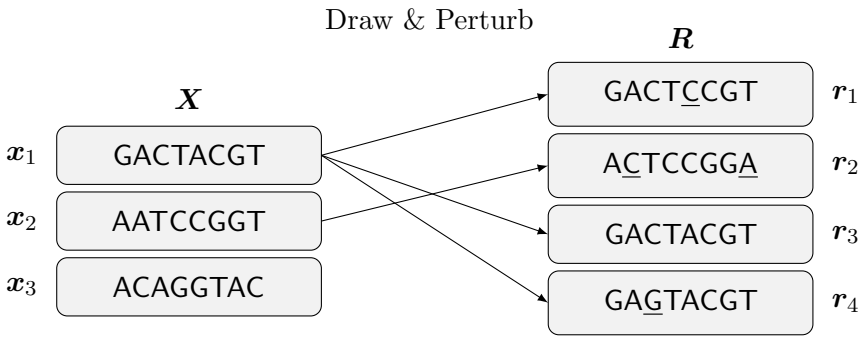


Figure 6.1: Exemplary realization of the DNA storage channel with $M = 3$ and $N = 4$. Each sequence $\boldsymbol{r}_j$ is obtained by drawing a random input sequence and sending it through a q-ary symmetric channel with crossover probability p. The arrows indicate the origin of each output sequence. Here, $(I_1, I_2, I_3, I_4) = (1, 2, 1, 1)$. Errors are underlined.

6.2.1 Codes for the DNA Storage Channel

The definition of codes over the probabilistic DNA storage channel is similar to that for codes over the unordered parallel multinomial channel in Section 5.2.1. We concisely highlight the main points and differences with respect to those codes. As for the case of the unordered parallel multinomial channel, a code is a set $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ such that each codeword is M sequences, each of length L over the alphabet Σ_q and its *rate* is given by

$$R = \frac{\log_q |\mathcal{C}|}{ML}.$$

Each code $\mathcal{C}$ is equipped with an encoder $\mathsf{enc}_{\mathcal{C}} : [q^{MLR}] \mapsto \mathcal{C}$ that maps a message $W \in [q^{MLR}]$ to a codeword and a decoder

$$\mathsf{dec}_{\mathcal{C}} : \Sigma_q^{N \times L} \mapsto [q^{MLR}]$$

that estimates the original message $\widehat{W}$ of the original message W given the received sequences $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N$. The error probability of a code $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ and a decoder $\mathsf{dec}_{\mathcal{C}}$ is given by

$$\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}\right) = \frac{1}{|\mathcal{C}|} \sum_{\mathtt{W}=1}^{q^{MLR}} \mathsf{Pr}\left(\mathsf{dec}_{\mathcal{C}}(\boldsymbol{r}_1, \ldots \boldsymbol{r}_N) \neq \mathtt{W} | W = \mathtt{W}\right),$$

where $\boldsymbol{r}_1, \ldots \boldsymbol{r}_N$ is the result of transmitting $\mathsf{enc}_{\mathcal{C}}(W) = (\boldsymbol{x}_1, \ldots \boldsymbol{x}_M)$ over the probabilistic DNA storage channel, and the message is chosen uniformly from the set of all messages $W \in [q^{MLR}]$.

6.2.2 Main Result

To begin with, the notion of achievable rates and channel capacity is an asymptotic result for large code lengths. Since the DNA storage channel consists of several code dimensions M, L and N we specify the precise asymptotic regime for which we derive the capacity. We consider the case, where $M \to \infty$ and $M = q^{\beta L}$ for some fixed $0 < \beta < 1$. Next, we set $N = cM$ for some fixed parameter $c > 0$. Note that this choice is motivated by the following two facts. First, $M = q^{\beta L}$ is the interesting case from an analytical perspective and from a practical perspective for systems,

where many short sequences are transmitted, as pointed out in Section 5.2.2. Further, c is a parameter controlling the *recovery rate* of the DNA storage systems, i.e., the average number of times each stored nucleotide is sequenced. From an efficiency perspective, it is advisable to choose this parameter such that the number of read nucleotides scales linearly with the number of stored nucleotides. We use the following standard notion of achievable rates and channel capacity.

Definition 6.1. *Fix $0 < \beta < 1, 0 < p < 1$, $c > 0$, $q \in \mathbb{N}$. A code rate R is* achievable, *if there exists a family of codes $\mathcal{C}(M, L, N) \subseteq \Sigma_q^{M \times L}$ with $|\mathcal{C}(M, L, N)| = q^{RML}$ and a decoder that has vanishing error probability $\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}(M, L, N)\right) \to 0$ as $M \to \infty$, where $M = q^{\beta L}$ and $N = cM$.*

The Shannon capacity $C_{\mathsf{DNA}}(c, \beta, p, q)$ is the supremum of achievable rates.

By this definition, it is possible to communicate reliably at information rates below the capacity. Conversely, transmitting at rates above the capacity is not possible with vanishing error probability. We are now in the position to formulate the main theorem on the capacity of the probabilistic DNA storage channel.

Theorem 6.2. *Fix the parameters $0 < \beta < 1, 0 < p < \frac{q-1}{2q}$, $c > 0$ $q \in \mathbb{N}$ with $2\beta < 1 - H_q(2p)$. Then, the capacity of the probabilistic DNA storage channel is given by*

$$C_{\mathsf{DNA}}(c, \beta, p, q) = \sum_{d \geq 0} \mathsf{Poi}_c(d) C_{\mathsf{Mul}}(d, p, q) - \beta(1 - \nu_0),$$

where $\mathsf{Poi}_c(d) = \frac{\mathrm{e}^{-c} c^d}{d!}$ is the probability mass function of the Poisson distribution and $C_{\mathsf{Mul}}(d, p, q)$ is the capacity of the multinomial channel from Lemma 5.2.

6.3 Probabilistic DNA Storage Channel as Degraded Unordered Parallel Multinomial Channel

We proceed with proving Theorem 6.2. A core part of the proof of Theorem 6.2 is that of identifying the probabilistic DNA storage channel as a degraded unordered parallel multinomial channel. That is, the output of the probabilistic DNA storage channel can be interpreted as a permutation of the sequences in the output clusters from the unordered parallel multinomial channel. Therefore, the receiver looses the information about the clusters, i.e., the groups of sequences, which originate from the same input sequence. We use this connection to prove a converse bound and achievable rates over the DNA storage channel. The converse bound is based on the fact that a degraded channel cannot have a capacity that exceeds the original channel. The achievable rates are derived via the analysis of a random code with a decoder that clusters the output sequences and then decides on a codeword based on a typicality measure between the clusters and codeword sequences similar to that used in Lemma 5.18.

Recall the notion of the unordered parallel multinomial channel from Chapter 5, which has input $\boldsymbol{X}$ and output $\boldsymbol{Y}$ and let its drawing composition $\boldsymbol{d}$ be according to the drawing model presented in Section 6.1. This section is devoted to showing that if we define an auxiliary channel that is comprised of an unordered parallel multinomial channel followed by a permutation channel, we obtain a channel that has the same input-output relationship as the DNA storage channel. More precisely, the auxiliary channel has input $\boldsymbol{X}$ and output $\boldsymbol{R}'$, where $\boldsymbol{R}'$ is obtained as follows. First, $\boldsymbol{X}$ is transmitted over an unordered parallel multinomial channel with drawing composition $\boldsymbol{d}$ that is derived from the independent and uniform draws $\boldsymbol{I}$, resulting in $\boldsymbol{Y}$. Now,

the $N = d_1 + \cdots + d_M$ sequences of all clusters are permuted according to a uniform random permutation $\boldsymbol{s}'$. The resulting permuted sequences comprise the channel output $\boldsymbol{R}'$. The auxiliary channel is visualized in Figure 6.2. Note that intuitively this equivalence is immediate to the fact that the random drawing of the DNA storage channel effectively results in a permutation of the output sequences and the multinomial channels of the unordered parallel multinomial channel represent the repeated usage of q-ary symmetric channels due to the drawing. However, we will provide more rigorous arguments of this equivalence and will show in the following that the overall input-output relationship of auxiliary channel is equivalent to that of the probabilistic DNA storage channel. By definition of the probabilistic DNA storage channel, we have

$$\mathsf{Pr}\left(\mathbf{R}'|\mathbf{X}\right) = \sum_{\mathbf{s}'} \mathsf{Pr}\left(\mathbf{R}', \mathbf{s}'|\mathbf{X}\right) = \sum_{\mathbf{s}'} \mathsf{Pr}\left(\mathbf{s}'\right) \mathsf{Pr}\left(\mathbf{R}'|\mathbf{X}, \mathbf{s}'\right)$$

The effect of the permutation $\boldsymbol{s}'$ can be illustrated as follows. Let the permutation $\boldsymbol{s}'$ be defined such that $s'_{i,j} = k$ means that the j-th output of the i-th cluster $\boldsymbol{y}_i$ is permuted to $\boldsymbol{r}'_k$. With this notation, $\boldsymbol{r}'_{s'_{i,1}}, \ldots, \boldsymbol{r}'_{s'_{i,d_i}}$ are the output sequences that correspond to the i-th output cluster $\boldsymbol{y}_i$. We can then rewrite the conditional output probability as

$$\begin{aligned}
\mathsf{Pr}\left(\mathbf{R}'|\mathbf{X}\right) &= \sum_{\mathbf{s}'} \mathsf{Pr}\left(\mathbf{s}'\right) \mathsf{Pr}\left(\boldsymbol{y}_1 = \left(\mathbf{r}'_{\mathbf{s}'_{1,1}}, \ldots, \boldsymbol{r}'_{\mathbf{s}'_{1,d_1}}\right), \ldots, \boldsymbol{y}_M = \left(\mathbf{r}'_{\mathbf{s}'_{M,1}}, \ldots, \boldsymbol{r}'_{\mathbf{s}'_{M,d_M}}\right) \middle| \mathbf{X}, \mathbf{s}'\right) \\
&= \sum_{\mathbf{d},\mathbf{s},\mathbf{s}'} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\mathbf{s}\right) \mathsf{Pr}\left(\mathbf{s}'\right) \prod_{i=1}^{M} p_{\mathtt{d}_i} \left(\left(\mathbf{r}'_{\mathbf{s}'_{i,1}}, \ldots, \mathbf{r}'_{\mathbf{s}'_{i,d_i}}\right) \middle| \mathbf{x}_{\mathbf{s}_i}\right) \\
&= \sum_{\mathbf{d},\mathbf{s},\mathbf{s}'} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\mathbf{s}\right) \mathsf{Pr}\left(\mathbf{s}'\right) \prod_{i=1}^{M} \prod_{j=1}^{\mathtt{d}_i} p_1 \left(\mathbf{r}'_{\mathbf{s}'_{i,j}} \middle| \mathbf{x}_{\mathbf{s}_i}\right) \overset{(a)}{=} \sum_{\mathbf{I}} \mathsf{Pr}\left(\mathbf{I}\right) \prod_{j=1}^{N} p_1(\mathbf{r}'_j|\mathbf{x}_{\mathbf{I}_j}),
\end{aligned}$$

where $p_1(\mathtt{y}|\mathtt{x})$ is the transition matrix for the q-ary symmetric channel with a single output sequence. In equality (a) we introduced the drawing variable $\boldsymbol{I}$. Due to the uniformity of the permutations, an output sequence $\boldsymbol{r}'_j$ is the result of an input sequence $\boldsymbol{x}_{I_j}$ with uniform probability and thus the variables I_j are also uniformly distributed. This establishes the equivalence of the auxiliary channel and the probabilistic DNA storage channel as desired.

6.4 Converse Bound

We rigorously prove the converse result in the following using the auxiliary channel model presented in Section 6.3.

Lemma 6.3. *Fix the parameters $0 < \beta < 1, 0 < p < \frac{q-1}{2q}$, $c > 0$ $q \in \mathbb{N}$ with $2\beta < 1 - H_q(2p)$. Then, any achievable rate R over the probabilistic DNA storage channel satisfies*

$$R \leq C_{\mathsf{DNA}}(c, \beta, p, q).$$

Proof. In Section 6.3 it has been illustrated that there exists an auxiliary channel, which has the same input-output relationship as the DNA storage channel. As two channels with the same input-output relationship have the same capacity, the auxiliary channel has the same capacity as the DNA storage channel. This auxiliary channel has been chosen as a degraded unordered parallel

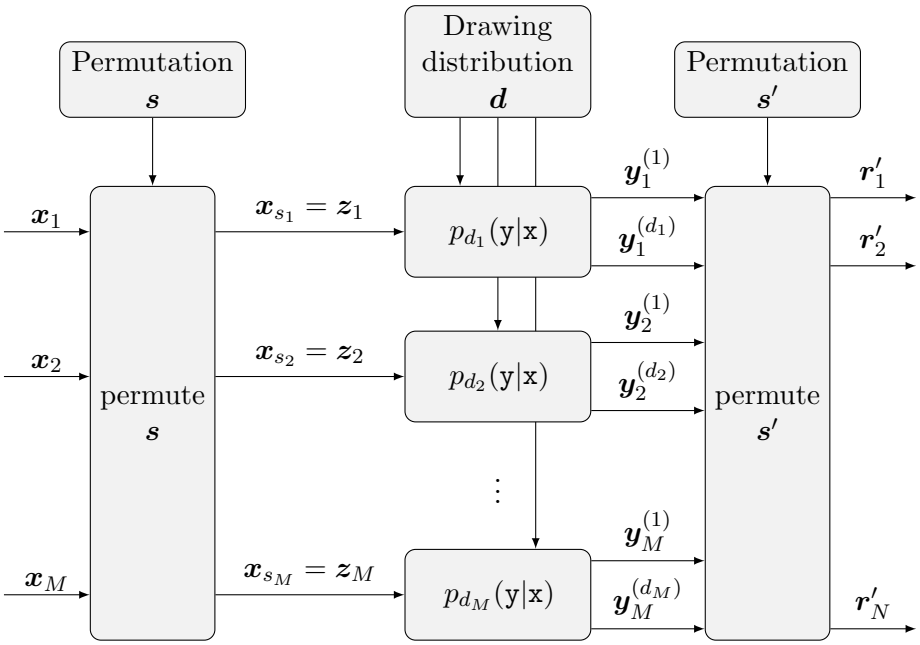


Figure 6.2: The probabilistic DNA storage channel as a degraded unordered parallel multinomial channel. Each cluster $\boldsymbol{y}_i$ consists of the d_i individual sequences $\boldsymbol{y}_i^{(1)}, \ldots, \boldsymbol{y}_i^{(d_i)}$. The $N = d_1 + \cdots + d_M$ individual sequences of all clusters are permuted, resulting in the output sequences $\boldsymbol{r}'_1, \ldots, \boldsymbol{r}'_N$.

multinomial channel with input $\boldsymbol{X}$, intermediate result $\boldsymbol{Y}$ and output $\boldsymbol{R}'$, where $\boldsymbol{X} - \boldsymbol{Y} - \boldsymbol{R}'$ form a Markov chain by construction. Thus, by the data processing inequality, the capacity of the auxiliary channel is at most the capacity of the unordered parallel multinomial channel, where the latter has a drawing composition $\boldsymbol{d}$ that is derived from the draws $\boldsymbol{I}$. It remains to prove that we can apply Theorem 5.10 for the capacity of the unordered parallel multinomial channel. To this end, we show that the drawing distribution is regular as defined in Definition 5.6. By Lemma 6.4, the distribution is regular and thus the converse result of the lemma holds by the data processing inequality. □

Lemma 6.4. *The distribution* $\mathsf{Pr}(\boldsymbol{d} = \mathbf{d})$ *that is induced by* $d_i = |\{j \in [N] : I_j = i\}|$, $i \in [M]$ *with i.i.d. uniform random variables* I_j *with* $\mathsf{Pr}(I_j = i) = \frac{1}{M}$ *for all* $j \in [N]$ *and* $i \in [M]$ *is a regular distribution in the sense of Definition 5.6 that converges to* $\nu_d = \mathsf{Poi}_c(d)$ *in frequency.*

Proof. We start by showing that the drawing frequency $n_d = |\{i \in [M] : d_i = d\}|$ converges to $\nu_d = \mathsf{Poi}_c(d)$ in frequency. To this end, for an arbitrary $\epsilon > 0$ denote by $\mathcal{N}_\epsilon$ the event that $\sum_{d \geq 0} |n_d - M\nu_d| \leq \epsilon M$. We will use the effect of *Poissonization* [Mit96] of the drawing distribution. Denote by $\widetilde{d}_i$, $i \in [M]$ independent and identically distributed random variables with mean c, i.e., $\mathsf{Pr}\left(\widetilde{d}_i = d\right) = \nu_d$. It has been shown [Mit96, Corollary 2.12] that any event on the exact distribution has probability at most $\sqrt{2\pi \mathrm{e} N}$ times the probability of the event for the case of i.i.d. Poisson variables. It follows that

$$\mathsf{Pr}(\boldsymbol{d} \notin \mathcal{N}_\epsilon) \leq \sqrt{2\pi \mathrm{e} N} \mathsf{Pr}\left(\widetilde{\boldsymbol{d}} \notin \widetilde{\mathcal{N}}_\epsilon\right), \tag{6.1}$$

where $\widetilde{\mathcal{N}}_\epsilon$ is the event that $\sum_{d\geq 0}|\widetilde{n}_d - M\nu_d| \leq \epsilon M$, where $\widetilde{n}_d = |\{i \in [M] : \widetilde{d}_i = d\}|$ is the drawing frequency derived from the i.i.d. Poisson variables. We split the probability in sequences with many draws and few draws,

$$\mathsf{Pr}\left(\widetilde{\boldsymbol{d}} \notin \mathcal{N}_\epsilon\right) \leq \mathsf{Pr}\left(\sum_{d=0}^{M^{1/3}-1} |\widetilde{n}_d - M\nu_d| \geq \frac{\epsilon M}{2}\right) + \mathsf{Pr}\left(\sum_{d\geq M^{1/3}} |\widetilde{n}_d - M\nu_d| \geq \frac{\epsilon M}{2}\right). \tag{6.2}$$

We now use that $\widetilde{n}_d$ is binomial distributed with M trials and success probability ν_d and thus

$$\mathsf{Pr}\left(|\widetilde{n}_d - M\nu_d| \geq \frac{\epsilon M^{2/3}}{2}\right) \overset{(a)}{\leq} 2\mathrm{e}^{-\frac{\epsilon^2}{2}M^{1/3}},$$

where we used Lemma A.4 on the two-sided binomial tail distribution in inequality (a). Therefore, by the union bound, the first summand in (6.2) is at most

$$\mathsf{Pr}\left(\sum_{d=0}^{M^{1/3}-1} |\widetilde{n}_d - M\nu_d| \geq \frac{\epsilon M}{2}\right) \leq \sum_{d=0}^{M^{1/3}-1} \mathsf{Pr}\left(|\widetilde{n}_d - M\nu_d| \geq \frac{\epsilon M^{2/3}}{2}\right) \leq 2M^{1/3}\mathrm{e}^{-\frac{\epsilon^2}{2}M^{1/3}}. \tag{6.3}$$

Next, we bound the second summand in (6.2). By the triangle inequality, $|\widetilde{n}_d - M\nu_d| \leq \widetilde{n}_d + M\nu_d$ and we thus bound the second summand by

$$\mathsf{Pr}\left(\sum_{d\geq M^{1/3}} |\widetilde{n}_d - M\nu_d| \geq \frac{\epsilon M}{2}\right) \leq \mathsf{Pr}\left(\sum_{d\geq M^{1/3}} \widetilde{n}_d \geq M\left(\frac{\epsilon}{2} - \sum_{d\geq M^{1/3}} \nu_d\right)\right).$$

To begin with, we see that $\sum_{d\geq M^{1/3}} \widetilde{n}_d$ is a binomial distribution with M trials and success probability $\sum_{d\geq M^{1/3}} \nu_d$. The tail of the Poisson distribution has exponential decay, see, e.g., [JLR00, Thm. 2.1], or, more precisely,

$$\sum_{d\geq M^{1/3}} \nu_d \leq \mathrm{e}^{-M^{1/3}}$$

for all $M \geq (7c)^3$ by the result of [JLR00, Eq. 2.11]. It follows that we can derive the following upper bound on the outage probability

$$\mathsf{Pr}\left(\sum_{d\geq M^{1/3}} \widetilde{n}_d \geq M\left(\frac{\epsilon}{2} - \sum_{d\geq M^{1/3}} \nu_d\right)\right) \leq \mathrm{e}^{-M\left(\epsilon/2 - 2\mathrm{e}^{-M^{1/3}}\right)^2}, \tag{6.4}$$

where we used the bound on the binomial tail from Lemma A.4. Note that this inequality only holds for large enough M, as we require $M \geq (7c)^3$ and also $\epsilon/2 > 2\mathrm{e}^{-M^{1/3}}$ in order for Lemma A.4 to apply. Plugging (6.4), (6.3), and (6.2) into (6.1), we obtain

$$\mathsf{Pr}\left(\boldsymbol{d} \notin \mathcal{N}_\epsilon\right) \leq \sqrt{2\pi\mathrm{e}cM}\left(2M^{1/3}\mathrm{e}^{-\frac{\epsilon^2}{2}M^{1/3}} + \mathrm{e}^{-M\left(\epsilon/2 - 2\mathrm{e}^{-M^{1/3}}\right)^2}\right) \to 0,$$

as $M \to \infty$, as the first exponent scales at least as $-\epsilon^2 M^{1/3}$ and the second exponent scales as $-\epsilon^2 M$. Thus, the polynomial scaling factors, which become logarithmic summands in the exponent, are asymptotically negligible.

Next, the total number of draws is precisely $N = cM$ by the definition of the drawing composition and thus also the condition on the maximum number of draws from Definition 5.6 applies.

We finally verify that the drawing frequency is balanced, as defined in Definition 5.6. The expected value of the drawing frequency is given by [KSC78; SC64]

$$\mathsf{E}[n_d] = M\frac{\binom{N}{d}}{M^d}\left(1-\frac{1}{M}\right)^{N-d} \overset{(b)}{\leq} M\frac{c^d}{d!},$$

where in inequality (b) we used that $\binom{n}{m} \leq \frac{n^m}{m!}$ for any integer $n, m \in \mathbb{N}$ and also that the exponential is less than 1. Denote by D_ϵ an integer to be chosen later and $\epsilon > 0$ an arbitrary positive constant. We can bound the sum over the weighted expected values by

$$\sum_{d\geq D_\epsilon} \mathsf{E}[n_d]\, d \leq M \sum_{d\geq D_\epsilon} \frac{c^d}{(d-1)!} = cM\mathrm{e}^c \sum_{d\geq D_\epsilon - 1} \nu_d \overset{(c)}{\leq} cM\mathrm{e}^c\mathrm{e}^{-D_\epsilon},$$

where we used [JLR00, Eq. 2.11] in inequality (c), which holds for all $D_\epsilon \geq 7c+1$. Consequently, if we strive to have a sum of at most ϵM, we can choose $D_\epsilon = \max\{7c+1, \ln\left(\frac{c\mathrm{e}^c}{\epsilon}\right)\}$, and the sum over the expected values is at most ϵM for all M as desired. This proves that the distribution is regular in the sense of Definition 5.6.

□

6.5 Achievable Rates

We proceed with proving that all rates $R < C_{\mathsf{DNA}}(c, \beta, p, q)$ over the probabilistic DNA storage channel are achievable. The main idea for the proof is that, given that the number of sequences is not too large, it is relatively easy to construct a deterministic clustering algorithm that groups all sequences, that stem from the same input sequence, to a cluster. Clearly, it is not possible to achieve perfect clustering, as some sequences might have had too many errors such that it is not possible to identify their cluster. However, we can show that a greedy clustering algorithm clusters almost all sequences correctly. We can then define a measure of typicality between the estimated clusters and the codewords, similar to this used for the achievable rate results in Lemma 5.18. That is, we count the largest matching between input sequences and clusters such that each pair in the matching is jointly typical with respect to the multinomial channel. Since most clusters are estimated correctly, the number of typical pairs between estimated clusters and a codeword will be close to that of the unordered parallel multinomial channel, which we have shown in Lemma 5.18 to be a decoding metric that results in vanishing error probabilities for all rates below capacity.

We proceed with setting up the necessary definitions required for the definition of the decoder. Our results will be proven with a random coding argument. Note that we use again the auxiliary equivalent channel model presented in Section 6.3. Let now $\mathcal{C} = \{\boldsymbol{X}(1), \ldots, \boldsymbol{X}(q^{MLR})\} \subseteq \Sigma_q^{M\times L}$ be a randomly chosen codebook of code rate R, where each codeword $\boldsymbol{X}(i) \in \Sigma_q^{M\times L}$ is selected independently and uniform over all possible words in $\Sigma_q^{M\times L}$. We will write $\boldsymbol{X}(i) = (\boldsymbol{x}_1(i), \ldots, \boldsymbol{x}_M(i))$. Further, $\boldsymbol{R}'$ is the output of the auxiliary channel. In order to define the decoder, we fix an $\alpha > 2p$ and $0 < \epsilon < 1$. The decoder consists of two parts. The first part is a clustering algorithm and the second is a stage that matches jointly typical between codewords and estimated clusters. Next is a description of the decoding algorithm and the clustering algorithm.

Algorithm 1 Clustering algorithm

Input: N received sequences $\boldsymbol{r}'_1, \dots, \boldsymbol{r}'_N$; cluster radius αL
Output: M Clusters $\widehat{\boldsymbol{y}}_1, \dots, \widehat{\boldsymbol{y}}_M$
$i \leftarrow 0$
$\mathcal{R} \leftarrow \{\{\boldsymbol{r}'_1, \dots, \boldsymbol{r}'_N\}\}$
while $\mathcal{R} \neq \emptyset$ **do**
    $i \leftarrow i + 1$
    for $\boldsymbol{r}' \in \mathcal{R}$ **do**
        if ($\widehat{\boldsymbol{y}}_i$ is empty) or $\left(d_{\mathsf{H}}\left(\boldsymbol{r}', \widehat{\boldsymbol{y}}_i^{(1)}\right) < \alpha L\right)$ **then**
            append $\boldsymbol{r}'$ to $\widehat{\boldsymbol{y}}_i$
            $\mathcal{R} \leftarrow \mathcal{R} \setminus \{\boldsymbol{r}'\}$
if $i > M$ **then** discard $\widehat{\boldsymbol{y}}_{M+1}, \dots, \widehat{\boldsymbol{y}}_i$
if $i < M$ **then** add empty clusters $\widehat{\boldsymbol{y}}_{i+1}, \dots, \widehat{\boldsymbol{y}}_M$

6.5.1 Clustering Algorithm

Let in the following $\alpha > 2p$ be the clustering radius. Consider Algorithm 1, which greedily picks an output sequence and adds other output sequences, such that their distance with respect to the first pick is less than αL. These sequences are combined to a cluster $\widehat{\boldsymbol{y}}_1$ and all elements in $\widehat{\boldsymbol{y}}_1$ are removed as candidates for succeeding clusters. The procedure successively continues to form clusters $\widehat{\boldsymbol{y}}_2, \dots, \widehat{\boldsymbol{y}}_{\widehat{M}}$ on the remaining sequences with the same procedure until no more sequences are present. Afterwards, the algorithm adds empty clusters or removes excess clusters, such that the total number of estimated clusters is M. It is evident that this clustering algorithm is neither efficient in computational complexity or accuracy, however it is easy to analyze and will suffice for our purposes.[1] Interestingly, under some mild conditions, this naive clustering algorithm produces many correct clusters. More precisely, we have the following. Consider the bipartite graph G_{cluster} with vertices $\boldsymbol{y}_j$, $j \in [M]$ on the left and $\widehat{\boldsymbol{y}}_i$, $i \in [M]$ on the right. We draw an edge from $\boldsymbol{y}_j$ to $\widehat{\boldsymbol{y}}_i$, if the multiset of sequences in $\boldsymbol{y}_j$ is equal to the multiset of sequences in $\widehat{\boldsymbol{y}}_i$. We define the number G of *correct* clusters as the size of the largest matching[2] in G_{cluster}.

Lemma 6.5. *Fix $0 < \beta < 1$ and α with $2p < \alpha < \frac{q-1}{q}$ and $\beta < 1 - H_q(\alpha)$ and an arbitrary $\epsilon > 0$. Then, the probability of having at least $M(1-\epsilon)$ correct clusters satisfies*

$$\lim_{M \to \infty} \mathsf{Pr}\left(G \geq M(1-\epsilon)\right) = 1.$$

Proof. Denote by G_i, $i \in [M]$ a binary indicator variable that is equal to 1, if $d_i > 0$ and $\boldsymbol{y}_i$ has been clustered correctly and 0, otherwise. Further, let $\widehat{M}$ be the number of non-empty clusters produced by Algorithm 1, before removing clusters or adding empty clusters. To start with, it holds that $G \geq \sum_{i=1}^{M} G_i + \min\{n_0, M - \widehat{M}\}$, since we can construct a matching, where we arbitrarily match the $M - \widehat{M}$ empty clusters and we match each cluster $\boldsymbol{y}_i$ with $G_i = 1$ to the correct cluster produced by the algorithm. Note that the edges of this matching share no common vertices, since the matching of empty clusters is arbitrary and the non-empty clusters, produced by Algorithm 1, are disjoint by construction of the algorithm. The bound further covers the case,

[1] A more elaborate clustering algorithm, designed for DNA-based data storage, can be found in, e.g., [Ras+17].
[2] A matching in a bipartite graph is a set of edges such that no two edges share common end points.

where $\widehat{M} > M$ where we possibly remove some of the correct clusters. Thus, by the union bound, the probability on the number of correct clusters is at least

$$\mathsf{Pr}\left(G \geq M(1-\epsilon)\right) \geq 1 - \mathsf{Pr}\left(\sum_{i=1}^{M} G_i \leq M - n_0 - M\epsilon/2\right) - \mathsf{Pr}\left(M - \widehat{M} \leq n_0 - M\epsilon/2\right). \quad (6.5)$$

We proceed with showing that the sum over the variables G_i in (6.5) is close to $M - n_0$ with high probability and $M - \widehat{M}$ is close to n_0 with high probability.

We start with the second term. To this end, denote by F_j, $j \in [\mathbb{N}]$ the binary indicator, which is equal to 1, if $d_{\mathsf{H}}\left(\boldsymbol{x}_{I_j}, \boldsymbol{r}'_j\right) \geq \alpha L/2$ and 0, otherwise, where I_j is the original input sequence that corresponds to $\boldsymbol{r}'_j$. With this definition, we observe that $\widehat{M} \leq M - n_0 + \sum_{j=1}^{N} F_j$. This is because whenever the clustering algorithm selects a sequence $\boldsymbol{r}'_j$ with $F_j = 0$, the remaining sequences j' from this cluster with $F_{j'} = 0$, that have not been clustered yet, will be contained in the estimated cluster. Thus, each sequence $\boldsymbol{r}'_j$ with $F_j = 1$ can produce at most one extra cluster. Since F_j, $j \in [N]$ are identical and independent Bernoulli random variables with success probability at most $\mathrm{e}^{-2L(\alpha/2-p)^2}$ (see Lemma A.4), it holds that

$$\mathsf{Pr}\left(M - \widehat{M} \leq n_0 - \epsilon M/2\right) \leq \mathsf{Pr}\left(\sum_{j=1}^{N} F_j \geq M\epsilon/2\right) \leq \mathrm{e}^{-M\left(\epsilon/(2c) - \mathrm{e}^{-2L(\alpha/2-p)^2}\right)^2} = o(1),$$

for all $\epsilon > 0$, as $L \to \infty$.

We turn towards the first summand in (6.5). Recall that according to Definition 5.13, we denote by $\boldsymbol{y}_i^{(1)}, \ldots, \boldsymbol{y}_i^{(d_i)}$ the sequences of a cluster $\boldsymbol{y}_i$. We can estimate the probability $\mathsf{Pr}\left(G_i = 1\right)$ for all i with $d_i > 0$ as follows. A cluster $\boldsymbol{y}_i$ is guaranteed to be estimated correctly, if $d_{\mathsf{H}}\left(\boldsymbol{x}_{s_i}, \boldsymbol{y}_i^{(j)}\right) \leq \alpha L/2$ for all sequences $j \in [d_i]$ and also if there exists no other output sequence $\boldsymbol{y}_{i'}^{(j')}$ from another cluster $i' \neq i$ that has distance less than αL to one of the sequences in the cluster $\boldsymbol{y}_i$. Demarginalizing with respect to the drawing composition, we obtain

$$\begin{aligned}\mathsf{Pr}\left(G_i = 1\right) &= \sum_{\mathsf{d}_i} \mathsf{Pr}\left(\mathsf{d}_i\right) \mathsf{Pr}\left(G_i = 1 | \mathsf{d}_i\right) \overset{(a)}{\geq} \sum_{\mathsf{d}_i \geq 1} \mathsf{Pr}\left(\mathsf{d}_i\right) \left(1 - \mathsf{d}_i \mathrm{e}^{-2L(\alpha/2-p)^2} - N\mathsf{d}_i q^{-L(1-H_q(\alpha))}\right) \\ &\overset{(b)}{\geq} \mathsf{Pr}\left(d_i \geq 1\right) - c\mathrm{e}^{-2L(\alpha/2-p)^2} - c^2 q^{-L(1-H_q(\alpha)-\beta)}\end{aligned}$$

where in inequality (a) we used the union bound and Lemma A.4 on the binomial tails together with the fact that there are at most N other output sequences, where each of these sequences has a marginal distribution that is uniformly random over all sequences of length L. In inequality (b), we used that $\sum_{\mathsf{d}_i \geq 1} \mathsf{Pr}\left(\mathsf{d}_i\right) \mathsf{d}_i \leq \mathsf{E}\left[d_i\right] = c$ and $M = q^{\beta L}$. Next, we compute

$$\mathsf{E}\left[M - n_0 - \sum_{i=1}^{M} G_i\right] \leq M\left(c\mathrm{e}^{-2L(\alpha/2-p)^2} + c^2 q^{-L(1-H_q(\alpha)-\beta)}\right),$$

where we used that $\mathsf{E}\left[n_0\right] = \sum_{i=1}^{M} \mathsf{Pr}\left(d_i = 0\right)$. Using Markov's inequality, we conclude that the probability of the first summand in (6.5) is at most

$$\mathsf{Pr}\left(M - n_0 - \sum_{i=1}^{M} G_i \geq M\epsilon/2\right) \leq \frac{2c\mathrm{e}^{-2L(\alpha/2-p)^2} + 2c^2 q^{-L(1-H_q(\alpha)-\beta)}}{\epsilon},$$

which approaches 0 as $M \to \infty$ for any $\beta < 1 - H_q(\alpha)$ and $\alpha > 2p$ and the claim follows. □

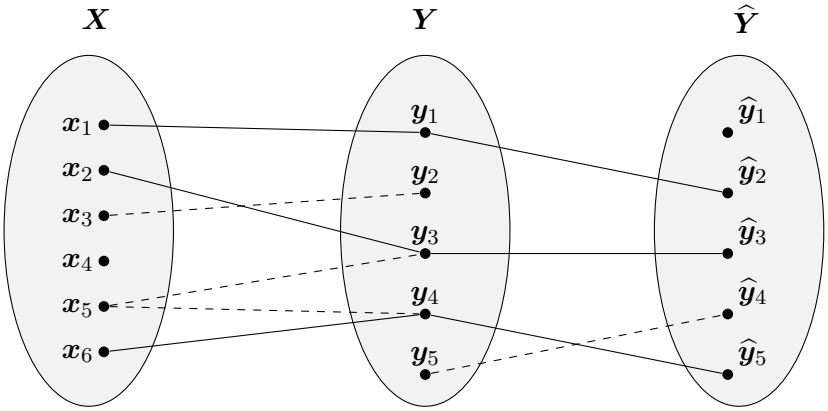


Figure 6.3: Illustration of the tripartite matching graph in the proof of Lemma 6.6. The solid lines highlight edges, which contribute to the joint typicality $T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}})$.

6.5.2 Typicality Matching

Recall Definition 5.19 of jointly typical sequences over the unordered parallel multinomial channel and denote by $\widehat{\boldsymbol{Y}}$ the estimated clusters from the clustering algorithm. Similar to the decoder presented in Section 5.4, we will define a decoder based on the joint typicality $T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}})$ between clustered sequences and a codeword. We will establish in the following a connection between the typicality of the estimated clusters $T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}})$ and the typicality of the actual correct clusters $T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \boldsymbol{Y})$.

Lemma 6.6. *The joint typicality of $\boldsymbol{X}$ and $\widehat{\boldsymbol{Y}}$ satisfies*

$$|T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}}) - T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \boldsymbol{Y})| \leq M - G.$$

Proof. First note that the ordering of the sequences in an estimated cluster $\widehat{\boldsymbol{y}}_i$ is arbitrary and thus might be different from that of the original cluster $\boldsymbol{y}_j$, even if the multisets of sequences are the same. However this does not affect the joint typicality over the multinomial channel, as it is invariant to permutations of the output sequences. Consider now the tripartite graph G_{tri} with vertices $\boldsymbol{x}_i$, $i \in [M]$ on the left, $\boldsymbol{y}_i$, $i \in [M]$ in the middle and $\widehat{\boldsymbol{y}}_i$, $i \in [M]$ on the right. We connect two vertices $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, if $(\boldsymbol{x}_i, \boldsymbol{y}_j) \in \mathcal{T}^{L,\epsilon}_{\mathsf{Mul}}(d, p, q)$. We further draw an edge from $\boldsymbol{y}_j$ to $\widehat{\boldsymbol{y}}_k$, if the multiset of sequences in $\boldsymbol{y}_j$ is equal to the multiset of sequences in $\widehat{\boldsymbol{y}}_k$. This tripartite graph is illustrated in Figure 6.3. Let $\mathcal{G} \subseteq [M]$ be the vertices in the middle which belong to the largest matching between the middle and right vertices, i.e., that correspond to the correct clusters, and let $\mathcal{H} \subseteq [M]$ be the vertices in the middle which belong to a matching between the middle and left vertices. With this definition,

$$T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}}) \geq |\mathcal{H} \cap \mathcal{G}| = |\mathcal{H}| + |\mathcal{G}| - |\mathcal{H} \cup \mathcal{G}| \overset{(a)}{\geq} |\mathcal{H}| + |\mathcal{G}| - M = |\mathcal{H}| + G - M,$$

where in inequality (a) we used that both $\mathcal{H}$ and $\mathcal{G}$ are subsets of $[M]$. Choosing $|\mathcal{H}| = T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \boldsymbol{Y})$ as the largest matching between left and middle vertices, yields an upper bound on the sough-after difference from the lemma statement. On the other hand,

$$T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}}) \leq |\mathcal{H} \cap \mathcal{G}| + |[M] \setminus \mathcal{G}| \leq T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \boldsymbol{Y}) + M - G,$$

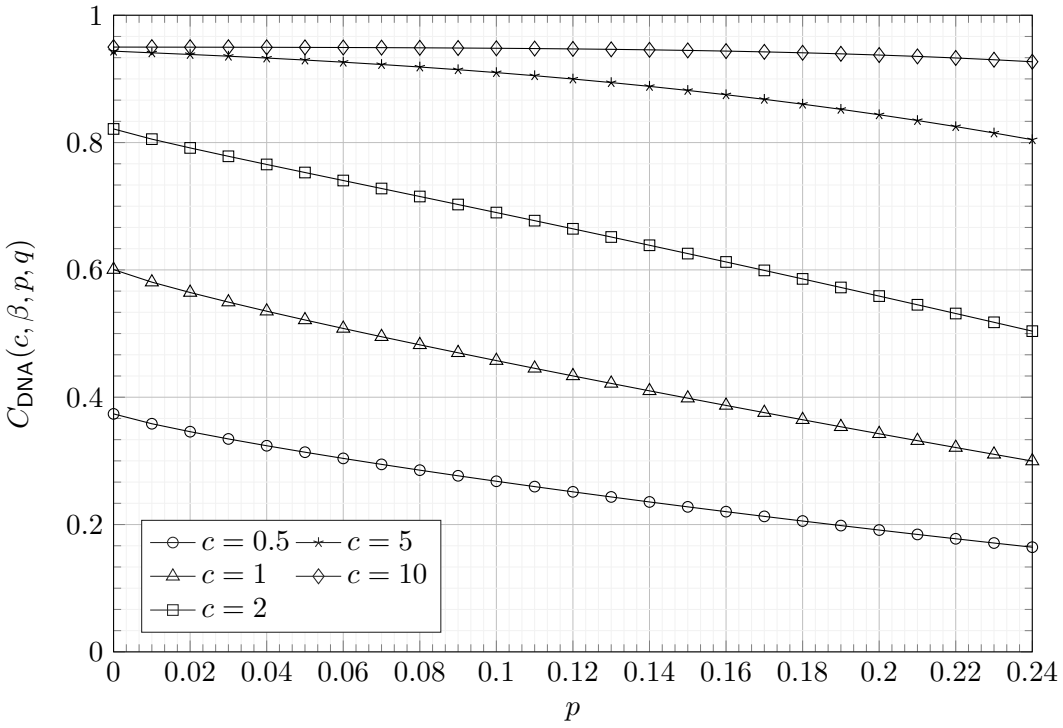


Figure 6.4: Capacity of the probabilistic DNA storage channel for $q = 4, \beta = \frac{1}{20}$ and different number of draws c over the channel error probability p.

since the number of correct clusters which can be matched to an input sequence is at most the size of the largest matching on the left $|\mathcal{H} \cap \mathcal{G}| \leq T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \boldsymbol{Y})$ and the $|[M] \setminus \mathcal{G}|$ incorrect clusters could potentially also add to the joint typicality. □

We conclude with our lemma on achievable rates over the probabilistic DNA storage channel.

Lemma 6.7. *Fix* $0 < \beta < 1, 0 < p < \frac{q-1}{2q}$, $q \in \mathbb{N}$ *with* $\beta < 1 - H_q(2p)$. *Then, any rate*

$$R < C_{\mathsf{DNA}}(c, \beta, p, q).$$

is achievable over the probabilistic DNA storage channel.

Proof. The statement directly follows from a random coding argument as in Lemma 5.18 using Lemmas 6.5 and 6.6 to show that for any $\epsilon > 0$, $|T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \widehat{\boldsymbol{Y}}) - T^{\epsilon}_{\mathsf{UPM}}(\boldsymbol{X}, \boldsymbol{Y})| < M\epsilon$ with high probability, together with Lemmas 5.20 and 5.21. □

6.6 Discussion and Efficiency Considerations

We start by discussing the capacity of probabilistic DNA storage channel for $q = 4$, which corresponds to the DNA alphabet and $\beta = \frac{1}{20}$, which is a typical value for current experiments (see Table 2.1 on page 17). Figure 6.4 shows the capacity $C_{\mathsf{DNA}}(c, \beta, p, q)$ for different number of draws c over the error probability p. The left-most points on the curves correspond to the error-free case

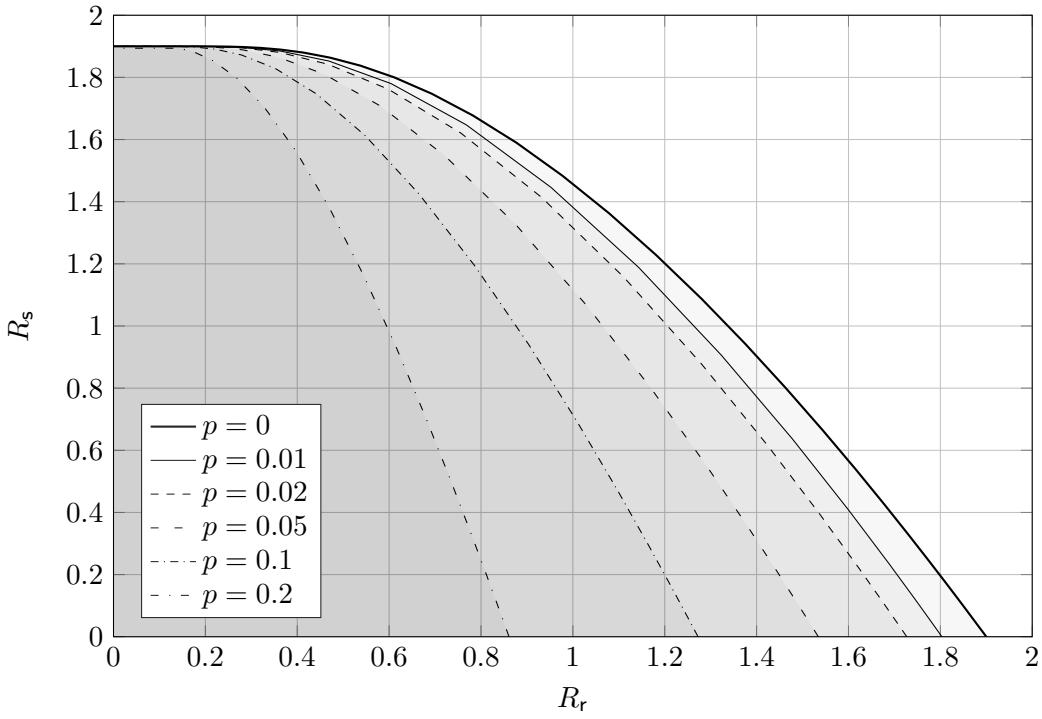


Figure 6.5: Storage and recovery rate trade-off for $\beta = \frac{1}{20}$ and different channel error probabilities. The graphs are generated by computing R_s and R_r for different values of c. On each curve, c decreases with increasing recovery rates R_r.

$p = 0$, where the capacity is equal to $C_{\mathsf{DNA}}(c, \beta, 0, q) = (1 - \beta)(1 - \mathrm{e}^{-c})$, as presented in [Hec+17]. For the discussion of efficient system design, we define the notions of *storage rate* and *recovery rate*. Assume $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ is a code used to store data in a DNA storage system. We define its storage rate to be the number of bits that can be stored per synthesized nucleotide, i.e.,

$$R_s = \frac{\log_2 |\mathcal{C}|}{ML}.$$

Note that here we choose the logarithm with respect to the base 2 such that the storage rate measures the number of information bits per stored nucleotide. Accordingly, if N sequences are drawn from the storage medium in order to recover the data, we define the recovery rate of a code $\mathcal{C}$ as the number of information bits that can be retrieved per nucleotide that is sequenced, i.e.,

$$R_r = \frac{\log_2 |\mathcal{C}|}{NL}.$$

With this definition, $R_s = cR_r$. Most publications to date focus on the storage rate R_s to evaluate the efficiency of their results. More recently, however, the interest in efficient design with respect to both storage rate R_s and recovery rate R_s has increased [Cha+19; Hec+17]. In this regard, Figure 6.5 shows the regions of achievable (R_r, R_s) rate pairs for different error probabilities p and $\beta = \frac{1}{20}$. Notably, the achievable region significantly flattens out for recovery rates $R_r \approx 0.2$bit/nt, which corresponds to $c \approx 10$, which should be considered for efficient system design. In particular, it becomes evident that an average sequencing depth of more than $\frac{N}{M} \approx 10$ sequences does not

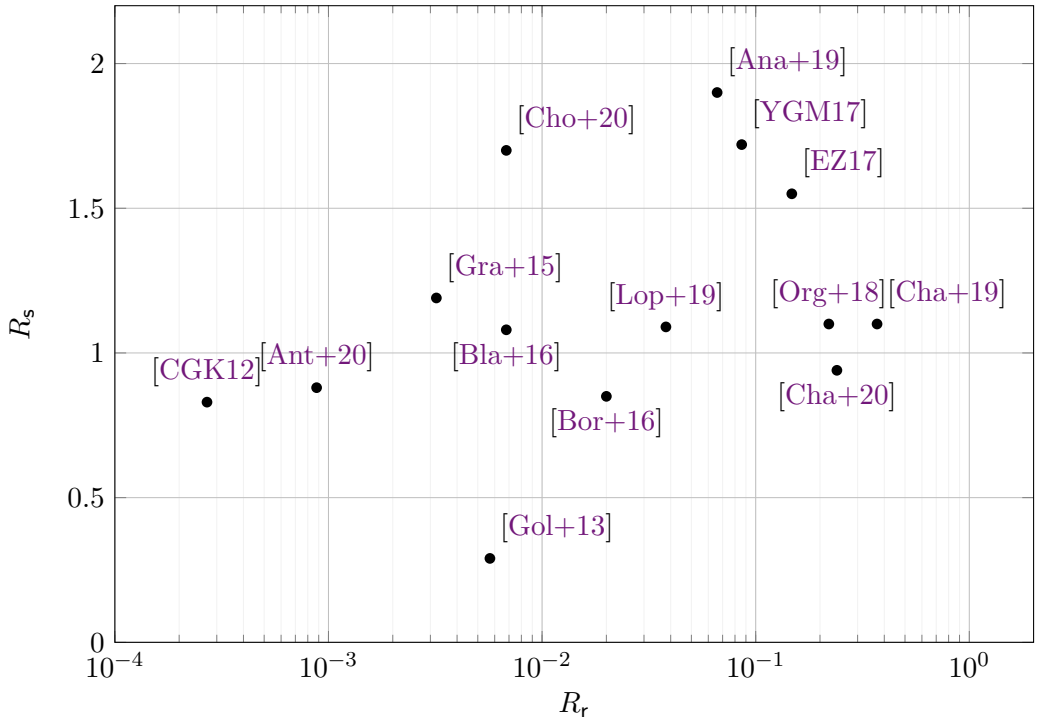


Figure 6.6: Storage and recovery rates of current experiments. The rates are accumulated from [EZ16; Org+18; Xu+21] and display the rates without taking eventual primers into account. We display the recovery rates as the largest rates for which the authors reported almost error-free decoding, if applicable. The recovery rates are visualized in a logarithmic scale in order to accommodate all recent experiments.

significantly improve the achievable storage rate. Note that this insight can be beneficial for the design of DNA-based data storage systems that are efficient with respect to both storage rates and recovery rates. Figure 6.6 displays the storage and recovery rates of current experiments.

6.7 Conclusion

This chapter was devoted to a probabilistic channel that originates from DNA-based data storage. In this channel, there are M input sequences, out of which N are drawn with replacement and received through a q-ary symmetric channel. We have shown that such a channel is a degraded unordered parallel multinomial channel, which allowed us to directly deduce a converse bound. On the other hand, we have introduced a novel notion of typicality between output sequences and a codeword. This typicality has been established using a greedy clustering algorithm that combines similar output sequences to one cluster and then matches estimated clusters with input sequences of a codeword. Using our capacity result, we have discussed the storage and recovery rate trade-off, which can be used for the design of DNA-based storage systems that are both storage and recovery efficient. An intriguing open problem that remains is the construction of a practical code operating close to the theoretical optimum. Another significant open problems is the generalization to channels that allow insertion and deletion errors.

Part III

Precise Asymptotic Analysis of Cost Constrained Channels

Chapter 7

Multivariate Singularity Analysis for Cost Constrained Channels

In the previous chapters, we have analyzed the transmission of information over erroneous channels. We now turn to a different subject, i.e., the analysis of discrete noiseless channels. The first work on cost constrained channels dates back to Shannon's seminal paper [Sha48]. Representing a cost constrained channel by a directed graph with labeled and costly edges, their *capacity*, i.e., the exponential growth rate of the number of limited-cost paths, was computed in [Sha48, Thm. 1]. Shannon showed that the capacity of a cost constrained channel is determined by the smallest singularity of a generating function that arises from the cost structure of the graph's edges. It was further shown in Theorem 8 that the capacity relates to the maximum entropy of a Markov chain that is defined based on the graph associated with the cost constrained channel. Shannon's results were later generalized to non-integer costs [BJP10; Böc+10; KMR00]. In this context, [KMR00] established a connection to analytic combinatorics in a single variable, and we build on this direction. The recent works [Liu+20; Sor05; SS06] treated the case of limited-cost and fixed-length paths. These studies showed the equivalence of the maximum entropy of an average cost constrained Markov chain and the fixed-length capacity, i.e., the exponential growth rate of the number of limited-cost and fixed-length paths.

In this chapter we derive the exact asymptotic expansion of the number of variable-length and fixed-length paths that have limited cost using recent results from analytical combinatorics in several variables [Mel21; PW13]. We start in Section 7.1 by introducing the notion of cost constrained channels and their capacity. We define generating functions that will serve as the basis for the multivariate singularity analysis later. Section 7.2 presents our main results of this chapter. In particular, we present novel and explicit theorems on the precise asymptotic growth of the size of limited-cost paths. While previous results on the capacity of cost constrained channels were mainly focused towards the probabilistic scenario, i.e., the maximization of the entropy of an associated Markov chain, our results include an explicit and easy-to-use algebraic formulation that can be used to compute the combinatorial capacity of cost constrained channels. We further exhibit with cost-diversity the precise property of costly graphs that differentiates between degenerate and smooth behavior of the fixed-length capacity. For the reader's convenience, we have summarized the roadmap with the main technical ideas and ingredients to prove our statements in Section 7.3. For our derivations, we build a comprehensive theory that extends the well-known results from Perron-Frobenius on irreducible matrices to costly matrices in Section 7.4 and derive their implications for cost-diverse graphs in Section 7.5. We use these results to infer

(a) Exemplary directed, labeled and costly graph

(b) Graph with period 2 and cost period 3

Figure 7.1: Exemplary directed graphs describing cost constrained channels with $\Sigma_2 = \{\mathsf{A}, \mathsf{C}\}$. The edges e are marked by their labels and costs $\sigma(e)|\tau(e)$.

properties on the singularities of the generating functions in Section 7.6, allowing to invoke a set of powerful results from the theory of analytical combinatorics in several variables that prove our main results. Our discussion is enriched with a concise exposition on multivariate singularity analysis. For an in-depth study of analytical combinatorics in several variables, however, we refer to the textbook [Mel21].

The results within this chapter have been published in [Len+20a; Len+21d].

7.1 Preliminaries

We start by setting up basic notation. In particular, we give a short overview over weighted and labeled graphs, followed by an introduction to generating functions of general integer series and a presentation of the generating function of the number of paths with limited cost.

7.1.1 Weighted and Labeled Graphs

Consider a labeled, directed graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ that has vertices $\mathcal{V}$ and edges $\mathcal{E}$. Each edge $e \in \mathcal{E}$ has an initial vertex $\mathsf{init}(e) \in \mathcal{V}$ and a terminal vertex $\mathsf{term}(e) \in \mathcal{V}$. Further, the edges are labeled with $\sigma : \mathcal{E} \mapsto \Sigma_q$ and have weights $\tau : \mathcal{E} \mapsto \mathbb{N}$. A path $\boldsymbol{p} = (e_1, \dots, e_n)$ of length n is a sequence of edges $e_1, \dots, e_n \in \mathcal{E}$ such that for all $i \in \{1, \dots, n-1\}$, the final vertex $\mathsf{term}(e_i)$ of the i-th edge is the same as the initial vertex $\mathsf{init}(e_{i+1})$ of the next edge. The path starts in $\mathsf{init}(e_1)$ and ends in $\mathsf{term}(e_n)$. A path generates a word $\sigma(\boldsymbol{p}) = (\sigma(e_1), \dots, \sigma(e_n)) \in \Sigma_q^n$ and has cost $\tau(\boldsymbol{p}) = \tau(e_1) + \cdots + \tau(e_n)$. Figure 7.1 shows a graph with edge labels and costs.

Definition 7.1. *A directed graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ is* strongly connected *if for any two vertices $v_i, v_j \in \mathcal{V}$, there exists a directed path that connects v_i with v_j.*

A key graph property is that all distinct paths emerging from a vertex generate distinct words. This is guaranteed by the following notion of a graph being *deterministic*, also known as *right-resolving* [LM95].

Definition 7.2. *A labeled, directed graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ is* deterministic *if for all vertices $v \in \mathcal{V}$, the labels of all edges $e \in \mathcal{E}$ with the same initial vertex $\mathsf{init}(e) = v$ are distinct.*

Here, we confine our analysis to deterministic graphs. Notice that known algorithms for the construction of a deterministic representation [MRS01, Prop. 2.2] may not directly work for costly graphs. We continue with presenting periodicity properties of graphs that are essential for the subsequent analysis. We start with the notion of the period of a graph.

Definition 7.3. *Let $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ be a strongly connected graph. We say G has period d, if for each pair of vertices v_i, v_j, the lengths of all paths $\boldsymbol{p}$ connecting v_i and v_j are congruent modulo d.*

With this definition, a graph can have several periods. In particular, if a graph has period d, all divisors of d are also periods of the graph. Note that our definition slightly differs from that in [MRS01, Section 3.3.2], as therein the period is defined as the greatest common divisor of all cycle lengths. However, as proven in Lemma A.8 in Appendix A.5, any graph that is periodic in the sense of Definition 7.3 is also periodic as defined in [MRS01]. We next establish the notions of uniformity and periodicity of the path costs in a strongly connected graph.

Definition 7.4. *A strongly connected graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ is* cost-uniform *if for each pair of vertices v_i, v_j and each length m, the costs of all length-m paths $\boldsymbol{p}$ connecting v_i and v_j are the same. If G is not cost-uniform, then we say that G is* cost-diverse*.*

Examples illustrating cost-uniformity and cost-diversity are displayed in Figure 7.2 on Page 140. For cost-diverse graphs, we further introduce the following notion of cost periodicity.

Definition 7.5. *Let $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ be a strongly connected graph. We say that G has cost period $c \in \mathbb{N}$, if for each pair of vertices v_i, v_j and each length m, the costs $\tau(\boldsymbol{p})$ of all length-m paths $\boldsymbol{p}$ connecting v_i and v_j are congruent modulo c.*

For convenience, we will in some cases use the convention that congruence modulo 0 means equivalence. We may then say that a cost-uniform graph is a graph with cost period 0, as in the cost-uniform case the costs of same-length paths connecting two vertices are equivalent. We proceed with a novel property that significantly facilitates our analysis and which we will prove to be equivalent to the cost period in Lemma 7.29 on Page 140.

Definition 7.6. *A strongly connected graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ satisfies the c-periodic coboundary condition if there exists a function $B : \mathcal{V} \to \mathbb{Q}$ and a constant $b \in \mathbb{Q}$ such that if $e \in \mathcal{E}$ is an edge from vertex v_i to vertex v_j, then the edge cost satisfies*

$$\tau(e) \equiv b + B(v_j) - B(v_i) \pmod{c}.$$

We say that a graph satisfies the coboundary condition, if the congruence above holds without the modulo operation.

A labeled and weighted graph G induces a costly language [KMR00; Sha48], which comprises all words that are generated by paths through the graph G of some limited cost. We thus use the term *cost constrained channel* to refer to a labeled, directed, weighted graph G that is deterministic and strongly connected. For many of our results, we analyze the spectrum of an adjacency matrix associated with the graph. In fact, we consider a family of adjacency matrices $\boldsymbol{P}_G(x)$, parameterized by a value x, which is also known as cost-enumerator matrix.

Definition 7.7. *Given a strongly connected graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ with vertices $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$, we define the* cost enumerator *matrix $\boldsymbol{P}_G(x)$ of G as the $|\mathcal{V}| \times |\mathcal{V}|$ matrix with entries*

$$[\boldsymbol{P}_G(x)]_{ij} = \sum_{e \in \mathcal{E}:\ \substack{\mathsf{init}(e)=v_i,\\ \mathsf{term}(e)=v_j}} x^{\tau(e)}.$$

We also define the spectral radius of $\boldsymbol{P}_G(x)$ by

$$\rho_G(x) = \max\{|\lambda(x)| : \lambda(x) \text{ is Eigenvalue of } \boldsymbol{P}_G(x)\}.$$

Throughout this chapter, we may treat x as either real-valued or complex-valued, depending on the context. Later we will see that $\rho_G(x)$ plays a central role in the asymptotic behavior of the number of limited weight paths through G. An important quantity is the number of distinct words that are contained in the language of a system.

Definition 7.8. *Given a graph $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$, for an arbitrary vertex $v \in \mathcal{V}$, we define $\mathcal{L}_{G,v}(t)$ to be the cost-t* follower set *of v, i.e., the set of all words that are generated by some path of cost at most t that starts in v. The size of the cost-t follower set is denoted as $N_{G,v}(t) \triangleq |\mathcal{L}_{G,v}(t)|$. Accordingly we define $\mathcal{L}_{G,v}(t,n) \triangleq \mathcal{L}_{G,v}(t) \cap \Sigma^n$ to be the fixed-length follower set with size $N_{G,v}(t,n) \triangleq |\mathcal{L}_{G,v}(t,n)|$.*

The central quantity of interest of a cost constrained channel is the exponential growth rate of the size of the follower set. This term is often referred to as *capacity*. The capacity of a strongly connected system is independent of the starting vertex, and we omit this in the definition.

Definition 7.9. *The **variable-length capacity** of a cost constrained channel G is defined to be*

$$C_G = \limsup_{t\to\infty} \frac{\log(N_{G,v}(t))}{t}.$$

*and similarly the **fixed-length capacity** is defined as*

$$C_G(\alpha) = \limsup_{t\to\infty} \frac{\log(N_{G,v}(t, \lfloor \alpha t \rfloor))}{t}.$$

7.1.2 Generating Functions

The methods of *analytic combinatorics* derive asymptotic properties of a sequence from analytic properties of its generating function [FS09; Mel21]. Throughout this chapter, the sequences of interest are the bivariate[1] sequences $N_{G,v}(t,n)$. We will denote their generating functions by

$$F_{G,v}(x,y) = \sum_{n\geq 0}\sum_{t\geq 0} N_{G,v}(t,n) x^t y^n,$$

where $x, y \in \mathbb{C}$. As the sequence $N_{G,v}(t,n)$ admits a linear recursion in the variables t and n, which we will elaborate on in Section 7.6.1, the generating function $F_{G,v}(x,y)$ is a fraction of polynomials, which we will denote by

$$F_{G,v}(x,y) = \frac{Q_{G,v}(x,y)}{H_G(x,y)}$$

for some polynomials $Q_{G,v}(x,y), H_G(x,y)$. Since $N_{G,v}(t) = \sum_{n\geq 0} N_{G,v}(t,n)$, for the variable-length case, we will regularly abbreviate $F_{G,v}(x) \triangleq F_{G,v}(x,1)$ as the generating function of the integer series $N_{G,v}(t)$ with numerator $Q_{G,v}(x) \triangleq Q_{G,v}(x,1)$ and denominator $H_G(x) \triangleq H_G(x,1)$.

The next lemma presents the generating function of the series $N_{G,v}(t,n)$.

Lemma 7.10. *Let G be a deterministic graph, and let v be a vertex. The generating function $F_{G,v}(x,y)$ of $N_{G,v}(t,n)$ is given by the entry of*

$$\boldsymbol{F}_G(x,y) = \frac{1}{1-x} \cdot (\boldsymbol{I} - y\boldsymbol{P}_G(x))^{-1}\mathbf{1}^{\mathrm{T}},$$

corresponding to the vertex v, where $\mathbf{1} = (1,\ldots,1) \in \mathbb{R}^{|\mathcal{V}|}$ and $\boldsymbol{I}$ is the $|\mathcal{V}| \times |\mathcal{V}|$ identity matrix.

[1] Bivariance refers to the fact that the integer series $N_{G,v}(t,n)$ depend on two variables t and n.

We will prove Lemma 7.10 in Section 7.6.1. Notice that $\boldsymbol{I} - y\boldsymbol{P}_G(x)$ is not always invertible. However, the singularities, i.e., the values of x and y for which $\boldsymbol{I} - y\boldsymbol{P}_G(x)$ is singular are precisely the objects of interest that determine the asymptotic behavior of the integer series $N_{G,v}(t, n)$.

Example 7.11. *Consider the graph in Figure 7.1a on Page 128. In this case,* $\boldsymbol{P}_G(x) = x + x^2$ *and thus the generating function of the single vertex is given by*

$$F_G(x, y) = \frac{1}{(1 - x)(1 - y(x + x^2))}.$$

7.2 Main Results

Our core results comprise the precise characterization of the number of limited-cost followers inside a given costly graph. For both the case of fixed-length and variable-length paths we find explicit expressions for the first order approximation of the number of followers in Theorem 7.14 and 7.16. This directly implies the exponential growth rate of the sequences, or *capacity*, which is stated in Theorems 7.12 and 7.15. We begin with a characterization of the fixed-length capacity for arbitrary strongly connected and cost-diverse graphs.

Theorem 7.12. *Let G be a strongly connected, deterministic, and cost-diverse graph. Abbreviate $\alpha_G^{\mathsf{lo}} \triangleq \rho_G(1)/\rho_G'(1)$ and $\alpha_G^{\mathsf{up}} \triangleq \lim_{x \to 0^+} \rho_G(x)/(x\rho_G'(x))$. For all α with $0 \leq \alpha \leq \alpha_G^{\mathsf{lo}}$, we have*

$$C_G(\alpha) = \alpha \log \rho_G(1).$$

For all α with $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$,

$$C_G(\alpha) = -\log x_0 + \alpha \log \rho_G(x_0),$$

where x_0 is the unique real solution to $\alpha x \rho_G'(x) = \rho_G(x)$ in the interval $0 < x < 1$. For all $\alpha > \alpha_G^{\mathsf{up}}$, $C_G(\alpha) = 0$.

Theorem 7.12 improves over previous work [JH84; KMR00; MR83; SS06] in several ways. First, the results of [JH84; KMR00; MR83] are only on the cost-constrained probabilistic capacity. Next, none of them explicitly recognizes the role of cost-diversity. Moreover, they do not address the full domain of the cost-constrained capacity. In contrast, our results explicitly determine the fixed-length capacity, they can be readily evaluated for cost-diverse graphs, and we consider the entire domain of the capacity function. Specifically, we identify a region for small α in which the capacity exhibits a linear scaling, we determine the exact slope in that region, and we explicitly find the threshold between the linear and non-linear regions. For examples illustrating Theorem 7.12, we refer to Proposition 8.10 in Chapter 8.

Remark 7.13. *Our results extend to the case of counting the number of followers of cost* exactly *t instead of* at most *t. In that case, the factor $(1 - x)$ in the numerator of the generating function $\boldsymbol{F}_G(x, y)$ is not present anymore, which has several effects on the results. First, the lower threshold α_G^{lo} decreases to $\alpha_G^{\mathsf{lo}} = \lim_{x \to \infty} \rho_G(x)/(x\rho_G'(x))$. Next, for all α outside the two thresholds, $C_G(\alpha) = 0$. This means that the linear region in α disappears.*

Theorem 7.12 is a direct consequence of the following stronger result, which gives the precise asymptotic behavior of $N_{G,v}(t, \alpha t)$. In the statement, we mean by largest period d and largest cost period c the largest integers such that the graph G has period d and cost period c, respectively.

Theorem 7.14. *Let G be a strongly connected, deterministic, and cost-diverse graph with largest period d and largest cost period c. Denote by b and $B(v_j)$ the quantities from the c-periodic coboundary condition.*

For all α with $0 < \alpha < \alpha_G^{\mathsf{lo}}$ and for any $v \in \mathcal{V}$, $N_{G,v}(t, \alpha t)$ has the asymptotic expansion

$$N_{G,v}(t, \alpha t) = \sum_{j=0}^{d-1} (\lambda_j(1))^{\alpha t} [\boldsymbol{u}_j^{\mathrm{T}}(1) \boldsymbol{v}_j(1) \mathbf{1}^{\mathrm{T}}]_v + O\left(\delta^t\right),$$

where $0 < \delta < (\rho_G(1))^{\alpha}$ and $\boldsymbol{u}_j(x), \boldsymbol{v}_j(x)$, $\boldsymbol{v}_j(x)\boldsymbol{u}_j^{\mathrm{T}}(x) = 1$ are the right and left Eigenvectors of $\boldsymbol{P}_G(x)$, corresponding to the Eigenvalues $\lambda_j(x) = \rho_G(x)\mathrm{e}^{2\pi \mathrm{i} j/d}$.

For all α with $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$ and t with $\alpha t \in \mathbb{N}$,

$$N_{G,v}(t, \alpha t) = \sum_{k=0}^{c-1} \sum_{j=0}^{d-1} \left(\frac{(\mathrm{e}^{2\pi \mathrm{i} bk/c} \lambda_j(x_0))^{\alpha}}{x_0 \mathrm{e}^{2\pi \mathrm{i} k/c}} \right)^t \frac{t^{-1/2}}{\sqrt{2\pi\alpha \mathcal{H}(x_0)}} \left(\frac{[\boldsymbol{D}_k^{-1} \boldsymbol{u}_j^{\mathrm{T}}(x_0) \boldsymbol{v}_j(x_0) \boldsymbol{D}_k \mathbf{1}^{\mathrm{T}}]_v}{(1 - x_0 \mathrm{e}^{2\pi \mathrm{i} k/c})} + O\left(\frac{1}{t}\right) \right),$$

where $\mathcal{H}(\mathrm{e}^s) = \frac{\partial^2}{\partial s^2} \ln \rho_G(\mathrm{e}^{\mathrm{s}})$, x_0 is the unique positive solution to $\alpha x \rho_G'(x) = \rho_G(x)$ and the $\boldsymbol{D}_k$ are diagonal matrices with $[\boldsymbol{D}_k]_{jj} = \mathrm{e}^{2\pi \mathrm{i} k B(v_j)/c}$.

For all $\alpha > \alpha_G^{\mathsf{up}}$, $N_{G,v}(t, \alpha t)$ is eventually 0.

To the best of our knowledge this, is the first first-order approximation of the number of limited-cost and fixed-length paths through arbitrary strongly connected graphs. Notably, disregarding the $O(1/t)$ term, the term following $t^{-1/2}$ is independent of t. We further present the results for the case of variable-length sequences. The following theorem is part of Shannon's famous results on discrete noiseless channels [Sha48] .

Theorem 7.15. *Let G be a strongly connected and deterministic graph and denote by x_0 the unique positive solution to $\rho_G(x) = 1$. Then, the combinatorial capacity of G satisfies*

$$C_G = -\log x_0.$$

For this theorem, we do not require the graph to be cost-diverse, as we are counting limited-cost paths of arbitrary lengths. We also obtain an exact expression for the size of the costly constrained language $N_{G,v}(t)$. This uses a univariate singularity analysis of the generating function $F_{G,v}(x)$.

Theorem 7.16. *Let G be a strongly connected and deterministic graph and denote by $x_1, \ldots, x_m$ the solutions to $(1-x)\det(\boldsymbol{I} - \boldsymbol{P}_G(x)) = 0$. Then, for any vertex $v \in \mathcal{V}$, there exist polynomials $\Pi_{G,v,i}(t)$, calculable from the generating function $F_{G,v}(x)$, such that*

$$N_{G,v}(t) = \sum_{i=1}^{m} \Pi_{G,v,i}(t) x_i^{-t}.$$

The degree of the polynomial $\Pi_{G,v,i}(t)$ is equal to the multiplicity of the root x_i minus one.

The following example illustrates that the polynomials $\Pi_{G,v,i}(t)$ can easily be computed with a partial fraction decomposition of the generating function.

Example 7.17. *Consider the graph from Figure 7.1 on Page 128. The cost-enumerator matrix of the graph is given by* $P(x) = x + x^2$ *and thus the generating function of* $N(t)$ *is equal to*

$$F(x) = \frac{1}{(1-x)(1-x-x^2)}.$$

The poles of this function are given by $x_1 = 1$, $x_2 = -\frac{\sqrt{5}+1}{2}$, $x_3 = \frac{\sqrt{5}-1}{2}$. *The coefficients of this generating function may be found by partial fraction decomposition [FS09], and we obtain*

$$F(x) = -\frac{1}{1-x/x_1} + \frac{1-\frac{2\sqrt{5}}{5}}{1-x/x_2} + \frac{1+\frac{2\sqrt{5}}{5}}{1-x/x_3},$$

such that $\Pi_1(t) = -1$, $\Pi_2(t) = 1 - \frac{2\sqrt{5}}{2}$ *and* $\Pi_3(t) = 1 + \frac{2\sqrt{5}}{2}$. *Using this decomposition, the sequence* $N(t)$ *is precisely*

$$N(t) = -1 + \left(1 - \frac{2\sqrt{5}}{5}\right)\left(\frac{1-\sqrt{5}}{2}\right)^t + \left(1 + \frac{2\sqrt{5}}{5}\right)\left(\frac{1+\sqrt{5}}{2}\right)^t.$$

Since the last summand is asymptotically dominant, the capacity of this constrained system is $C = \log\left((1+\sqrt{5})/2\right)$, *which can alternatively be derived from the positive solution* $x_0 = \frac{\sqrt{5}-1}{2}$ *to* $\rho_G(x) = 1$, *where* $\rho_G(x) = x + x^2$, *confirming Theorem 7.15.*

7.3 Technical Overview

We provide an overview of the ingredients required to prove Theorems 7.12, 7.14, 7.15, and 7.16. To begin with, we concisely highlight the main steps of a multivariate singularity analysis [Mel21] that connects properties of specific singularities to the asymptotic expansion of the *diagonal* coefficients $N_G(t, \alpha t)$. Afterwards, we discuss how we use the theory on irreducible matrices [HJ12] to show the implications of strong connectivity and cost-diversity on the spectral properties of cost-enumerator matrices and thus on the singularities of the generating functions.

7.3.1 Analytical Combinatorics in Several Variables

Analytic combinatorics [FS09] is a branch in mathematics that uses complex analysis to deduce the asymptotics of an integer sequence $N(t)$ from its generating function $F(x)$. Similary, *analytic combinatorics in several variables* (ACSV) [Mel21] treats multivariate integer sequences $N(t_1, t_2)$ (this discussion is specialized to the bivariate case we consider) and their generating functions $F(x, y)$ (see Section 7.1.2). The multivariate analysis resembles the univariate case, translating properties of the generating function near singularities to an asymptotic expansion of the integer series. Due to the multivariate nature of the series, there are several ways how the coefficients (t_1, t_2) can grow to infinity. Thus, usually, one sets $(t_1, t_2) = (t\alpha_1, t\alpha_2)$ and lets $t \to \infty$, as this entails a uniform asymptotic behavior of $N(\alpha_1 t, \alpha_2 t)$, and (α_1, α_2) is referred to as the *diagonal*. Similar to the case of the univariate analysis, the singularities closest to the origin determine the asymptotic behavior of the diagonal. In the multivariate case, however not all of those are relevant for the asymptotics. The two following properties of singularities thus come into play.

Minimal points are those singularities for which $H(x, y)$ has no other root with strictly smaller coordinate-wise modulus.[2] A minimal point is *strictly minimal* if no other singularity has the same coordinate-wise modulus, and *finite minimal* only a finite number of other singularities have the same coordinate-wise modulus. *Critical points*, for the case of rational generating functions $F(x, y) = Q(x, y)/H(x, y)$, are those satisfying $H(x, y) = \alpha_2 x H_x(x, y) - \alpha_1 y H_y(x, y) = 0$. For the generating function that we treat in our analysis, there are two types of critical points. First, the *smooth* points, where at least one of the partial derivatives does not vanish. Second, the non-smooth *multiple* points [PW04], where the singularity set is the union of two smooth surfaces that intersect in this point, meaning that both partial derivatives vanish.

Under a few additional conditions, the existence of minimal critical points means asymptotics of $N(t\alpha_1, t\alpha_2)$ can be determined from local properties of the generating function $F(x, y)$ near these points. For more details, see Section 7.6.2

7.3.2 From Cost-Diverse Graphs to Multivariate Analytical Combinatorics via Spectral Analysis

The starting point of our ACSV analysis is the generating function derived in Lemma 7.10. Before we can invoke the general results of ACSV however, we need to establish a comprehensive theory about cost constrained channels and their associated cost-enumerator matrix to gather the necessary understanding of the singularities. To start with, through the restriction to cost-diverse graphs (Definition 7.4), we avoid certain degenerate cases. Previous work [KMR00] observed that graphs with constant edge cost have the very specific property that the cost of any path is a linear function of its length, meaning that the capacity is simply determined by the number of paths through the graph of a given length. Generalizing this observation, we introduce the notion of cost-uniform and cost-diverse graphs (Definition 7.4). We show that if a graph is not cost-diverse, i.e., it is cost-uniform, then the cost of any path is an affine linear function in the path length and thus the average cost of any path approaches a constant. Hence, the fixed-length capacity is only non-zero if the graph is cost-diverse or if we restrict to a very specific path length α.

Focusing on cost-diverse and strongly connected graphs, we derive a variety of interesting properties of such graphs. To start with, due to the fact that the cost-enumerator matrix $\boldsymbol{P}_G(x)$ of a strongly-connected graph is irreducible (see Definition 7.18) for positive $x \in \mathbb{R}^+$, we start in Section 7.4 by deriving general properties of irreducible matrices. To this end, we use the famous Perron-Frobenius Theorem (Theorem 7.19) and a refinement [MRS01, Thm. 3.18] (see Theorem 7.20) to deduce properties of the parametrized cost-enumerator matrix. These results will serve us in Section 7.5 to derive spectral properties of the cost-enumerator matrix of a cost-diverse graph. A key milestone for our results is Lemma 7.29, which provides an equivalence between cost-diversity and the coboundary condition (Definition 7.6) and their implication on the cost-enumerator matrix, i.e., a nice behavior of the spectral radius under rotations, and the log-log-linearity or log-log-convexity of the spectral radius along the real axis.

Our equivalence result in Lemma 7.29 establishes key properties of the cost-enumerator matrix $\boldsymbol{P}_G(x)$ and is the basis for a derivation of the attributes of the generating function. This appears in Section 7.6. At a high level, we need to find the minimal singularities of our generating functions $\boldsymbol{F}_G(x, y)$ and characterize the critical points in order to apply the ACSV theorems in Section 7.6.2. More concretely, in Lemma 7.48, we identify the minimal singularities of $\boldsymbol{F}_G(x, y)$ and express

[2] We use the terms *modulus*, *absolute value*, and *magnitude* of a complex variable interchangeably in this dissertation.

them as a function of the graph period d, the cost period c and the spectral radius $\rho_G(x_0)$. Due to the Perron-Frobenius Theorem, $\rho_G(x_0)$ is the single real Eigenvalue of maximum modulus of $\boldsymbol{P}_G(x)$, which we use to show that $(x_0, 1/\rho_G(x_0))$ are minimal singularities. Next, we prove in Lemma 7.49 that the minimal points that we have found Lemma 7.48 are smooth points. We further derive a condition based on α, the spectral radius $\rho_G(x)$ and its derivative $\rho'_G(x)$ that determines criticality of the minimal singularities. A key component of the proof is Lemma 7.32, which shows that the rotation of x by multiples of $2\pi/c$ along the complex circle results in similar cost-enumerator matrices. Diving deeper into the critical point condition, Lemma 7.50 guarantees a unique smooth critical point when α is in a certain interval. The proof uses the strict log-log convexity of $\rho_G(x)$ proven in Lemma 7.39. The final component of our multivariate singularity analysis is Lemma 7.51, which proves that the singular set near the smooth critical points has non-degenerate geometry. For an overview over this roadmap, see Figure 7.3 on Page 141.

To establish Theorem 7.14, we then apply results from [Mel21] and use the spectral properties of $\boldsymbol{P}_G$ that we have derived from the graph properties. When $(x_0, 1/\rho_G(x_0))$ is a smooth point of the singular set of the generating function asymptotic behaviour is determined using Theorem 7.46, while in the non-smooth case it follows from an application of Theorem 7.47.

7.4 Perron-Frobenius Theory

In this section, we shortly revisit the central statements of the famous Perron-Frobenius theorem and derive associated results on irreducible matrices $\boldsymbol{P}(x)$, which are parametrized by a variable x. These results are key ingredients to prove our main statements.

7.4.1 Known Results from Perron-Frobenius Theory

The Perron-Frobenius Theorem is a well-known result about the spectral properties of irreducible matrices. For the following definition of irreducible matrices, recall the notion of strong connectivity of a graph from Definition 7.1.

Definition 7.18. *Let $\boldsymbol{P} \in \mathbb{R}^{M\times M}$ be a square real matrix with nonnegative entries. Associate with $\boldsymbol{P}$ the directed graph G with M vertices which is constructed by connecting state i to j if and only if $[\boldsymbol{P}]_{ij} > 0$. We call $\boldsymbol{P}$* irreducible *if G is strongly connected.*

Perron [Per07] and Frobenius [Fro12] revealed important properties on the spectral properties, i.e., the nature of the Eigenvalues, of irreducible matrices. Among those, they showed that irreducible matrices admit a single positive real Eigenvalue, which is equal to the spectral radius, i.e., the largest magnitude assumed by any Eigenvalue. In the following statements, which are an excerpt of the original Perron-Frobenius theorem, we collect those properties of irreducible matrices that are most relevant for our purposes.

Theorem 7.19 ([Fro12; Per07])**.** *Let $\boldsymbol{P}$ be an irreducible matrix with spectral radius ρ. Then,*

1. *ρ is an Eigenvalue with multiplicity one.*
2. *There exist positive right and left Eigenvectors $\boldsymbol{u} > 0$ and $\boldsymbol{v} > 0$ corresponding to the Eigenvalue ρ such that $\boldsymbol{P}\boldsymbol{u}^{\mathrm{T}} = \rho\boldsymbol{u}^{\mathrm{T}}$ and $\boldsymbol{v}\boldsymbol{P} = \rho\boldsymbol{v}$.*

By the Perron-Frobenius theorem, for an irreducible matrix with spectral radius ρ, there is a unique Eigenvalue λ, which is is equal to the spectral radius. We will refer to this Eigenvalue as the Perron root in the sequel. In fact, the structure of the Eigenvalues on the spectral circle are precisely known for irreducible matrices. To characterize these Eigenvalues, recall the definition of periodicity of a graph from Definition 7.3. If the largest period of an irreducible matrix $\boldsymbol{P}$ is d,[3] then $\boldsymbol{P}$ has precisely d simple Eigenvalues of maximum modulus. More precisely, those Eigenvalues precisely divide the complex circle into d equally sized segments. The following theorem summarizes this property.

Theorem 7.20 ([MRS01, Thm. 3.18])**.** *Let $\boldsymbol{P}$ be an irreducible matrix with largest period d. Then $\boldsymbol{P}$ has precisely d simple Eigenvalues of maximum modulus. Denoting ρ as the spectral radius of $\boldsymbol{P}$, those Eigenvalues have the form $\rho \mathrm{e}^{2\pi \mathrm{i} j/d}$, where $j \in \{0, 1, \ldots, d-1\}$.*

Notice that our definition of periodicity slightly differs from that in [MRS01], however it is possible to verify that periodicity in the sense of [MRS01] follows from our definition of periodicity, which we prove in Lemma A.8 in Appendix A.5. Another very useful result for irreducible matrices is Wielandt's theorem [Wie50]. We present the theorem in the following, as we will require it in several places of our subsequent derivations.

Theorem 7.21 ([Wie50])**.** *Let $\boldsymbol{P} \in \mathbb{R}^{M\times M}$ be an irreducible matrix and $\boldsymbol{Q} \in \mathbb{C}^{M\times M}$ be a matrix with $|[\boldsymbol{Q}]_{ij}| \leq [\boldsymbol{P}]_{ij}$. Then $\rho(\boldsymbol{Q}) \leq \rho(\boldsymbol{P})$. Further, equality holds (i.e., $\rho(\boldsymbol{P})\mathrm{e}^{\mathrm{i}\phi}$ is an Eigenvalue of $\boldsymbol{Q}$ for some ϕ) if and only if there exist $\theta_1, \ldots, \theta_M$ such that*

$$\boldsymbol{Q} = \mathrm{e}^{\mathrm{i}\phi} \boldsymbol{D}^{-1} \boldsymbol{P} \boldsymbol{D},$$

where $\boldsymbol{D}$ is a diagonal matrix with entries $[\boldsymbol{D}]_{jj} = \mathrm{e}^{\mathrm{i}\theta_j}$.

The power of this theorem lies in the exact characterization under which conditions, the spectral radii of two matrices, where one matrix is component-wise smaller than the other, agree. For a detailed proof of this theorem and for more details on irreducible matrices, including a comprehensive section on the Perron-Frobenius Theorem, we refer the reader to the textbooks [Mey00, Section 8.3] and [HJ12, Section 8.4].

7.4.2 Essentials on Irreducible Matrices

We proceed with establishing basic results on irreducible matrices, which will be used in the derivation of our main statements. Assume that $\boldsymbol{P}$ is an irreducible matrix with period d. We start with a simple result on the rank of the adjoint matrix $\rho \mathrm{e}^{2\pi \mathrm{i} j/d}\boldsymbol{I} - \boldsymbol{P}$, where ρ is the spectral radius of the irreducible matrix $\boldsymbol{P}$.

Lemma 7.22. *Let $\boldsymbol{P}$ be an irreducible matrix with period d and spectral radius ρ. Then, the adjoint matrix $\mathrm{adj}(\rho \mathrm{e}^{2\pi \mathrm{i} j/d}\boldsymbol{I} - \boldsymbol{P})$ has rank one for all $j \in \{0, 1, \ldots, d-1\}$.*

Proof. Note that the result can be deduced from, e.g., [Mey00, Prob. 6.2.11] and we provide a short proof for the readers convenience. Denote by M the number of rows (and columns) of $\boldsymbol{P}$ and abbreviate for convenience $\theta_j \triangleq 2\pi j/d$. We first show that $\mathrm{rank}(\rho\boldsymbol{I} - \boldsymbol{P}) = M - 1$. The Eigenvalues of $\rho \mathrm{e}^{\mathrm{i}\theta_j}\boldsymbol{I} - P$ are given by $(\rho \mathrm{e}^{\mathrm{i}\theta_j} - \lambda_i)$, $i \in \{1, \ldots, M\}$, where λ_i are the (not necessarily

[3] We say a matrix $\boldsymbol{P}$ has period d if the associated directed graph (see Definition 7.18) has period d.

distinct) Eigenvalues of $\boldsymbol{P}$. Since $\boldsymbol{P}$ is irreducible and has period d, by the Perron-Frobenius Theorem 7.19 and Theorem 7.20 $\rho e^{i\theta_j}$, $j \in \{0, 1, \ldots, d-1\}$ are Eigenvalues of multiplicity one and thus exactly one of the Eigenvalues $\rho e^{i\theta_j} - \lambda_i$ will be zero and all other non-zero. Therefore $\operatorname{rank}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = M - 1$. Next, we observe that $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = \det(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})\boldsymbol{I} = \boldsymbol{0}$ and thus $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$ spans a subspace of the left nullspace of $(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$. Since $\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}$ has rank $M - 1$, it follows that $\operatorname{rank}(\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})) \leq 1$. On the other hand, $\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}$ has rank $M - 1$ and thus there exists an $(M-1) \times (M-1)$ submatrix of $\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}$, which is non-singular [Mey00, Ch. 4.5], and it follows that at least one entry of $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$ is non-zero. It follows that $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$ cannot have rank zero and thus has rank one. □

Next, we establish a useful characterization of the adjoint matrix $\operatorname{adj}(\rho\boldsymbol{I} - \boldsymbol{P})$. In particular, we will show that we can represent this adjoint matrix as the outer product of the right and left Eigenvector associated with the Perron root ρ.

Lemma 7.23. *Let $\boldsymbol{P}$ be an irreducible matrix with period d. Then, there are d Eigenvalues $\rho e^{2\pi i j/d}$, $j \in \{0, 1, \ldots, d-1\}$ of maximum modulus and we denote their corresponding right and left Eigenvectors by $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$, which are normalized to $\boldsymbol{v}_j(x)\boldsymbol{u}_j^{\mathrm{T}}(x) = 1$. The adjoint matrix $\operatorname{adj}(\rho e^{2\pi i j/d}\boldsymbol{I} - \boldsymbol{P})$ is given by*

$$\operatorname{adj}(\rho e^{2\pi i j/d}\boldsymbol{I} - \boldsymbol{P}) = c_j \cdot \boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{v}_j,$$

where $c_j \neq 0$ is a linear scaling factor. Thus, $\operatorname{adj}(\rho\boldsymbol{I} - \boldsymbol{P})$ is either all-positive or all-negative.

Proof. Abbreviate for convenience $\theta_j \triangleq 2\pi j/d$. By Lemma 7.22, the adjoint matrix has rank one. It follows that $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$ can be written as the product $\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{v}_j$ of two vectors $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$, i.e., $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = \boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{v}_j$. The properties of the adjoint matrix [HJ12, p. 20] imply that

$$\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = (\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = \det(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})\boldsymbol{I}.$$

By Theorem 7.20, $\rho e^{i\theta_j}$ is an Eigenvalue of $\boldsymbol{P}$, which implies that $\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}$ is singular, so $\det(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = 0$. Hence,

$$\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = (\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = \boldsymbol{0}.$$

Therefore, the columns of $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$ are right eigenvectors of $\boldsymbol{P}$ associated to $\rho e^{i\theta_j}$. Similarly, the rows of $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})$ are left eigenvectors of $\boldsymbol{P}$ associated to $\rho e^{i\theta_j}$, and therefore, $\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P}) = \boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{v}_j$, where $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are right and left Eigenvectors corresponding to $\rho e^{i\theta_j}$. It is not possible that $c_j = 0$, since $\operatorname{rank}(\operatorname{adj}(\rho e^{i\theta_j}\boldsymbol{I} - \boldsymbol{P})) = 1$ by Lemma 7.22.

We proceed with proving the second statement. By the Perron-Frobenius Theorem, $\boldsymbol{u}_0$ is either all-zero, all-positive, or an all-negative vector and the same applies to $\boldsymbol{v}_0$. If we now assume that $\boldsymbol{B} \triangleq \operatorname{adj}(\rho\boldsymbol{I} - \boldsymbol{P})$ satisfies $[\boldsymbol{B}]_{11} > 0$, the observations above imply that every entry of $\boldsymbol{B}$ must be positive. Similarly, if $[\boldsymbol{B}]_{11} < 0$, we can conclude that every entry of $\boldsymbol{B}$ must be negative.

□

The cost-enumerator matrix $\boldsymbol{P}_G(x)$ is a matrix that is parametrized in a complex-valued variable $x \in \mathbb{C}$. In our analysis, due to the strong connectivity of the graph G, $\boldsymbol{P}_G(x)$ is irreducible for all positive and real-valued $x \in \mathbb{R}^+$. By Definition 7.7, the entries of $\boldsymbol{P}_G(x)$ are polynomials in x and

thus analytic[4] functions in x. This analyticity then implies, by the implicit function theorem for algebraic functions, that the Eigenvalue $\lambda(x)$ that is equal to $\rho_G(x)$ on the real axis, is analytic in a neighborhood around the positive real axis.

Lemma 7.24. *Let $\boldsymbol{P}(x)$ be a matrix with spectral radius $\rho(x)$, whose entries are analytic functions in $x \in \mathbb{C}$. Also assume that $\boldsymbol{P}(x)$ is irreducible with period d for all $x \in \mathbb{R}^+$. Then, for each $j \in \{0, 1, \dots, d-1\}$ and all real-valued $x \in \mathbb{R}^+$ there exists a unique Eigenvalue $\lambda_j(x)$ of $\boldsymbol{P}(x)$ with $\lambda_j(x) = \rho(x)\mathrm{e}^{2\pi \mathrm{i} j/d}$, which is analytic in a complex neighborhood around the positive real axis. Further, the associated right and left Eigenvectors $\boldsymbol{u}_j(x)$ and $\boldsymbol{v}_j(x)$, normalized to $\boldsymbol{v}_j(x)\boldsymbol{u}_j^{\mathrm{T}}(x) = 1$, are analytic on the same domain.*

Proof. By the Perron-Frobenius Theorem (Theorem 7.19) and the extension in Theorem 7.20 for every $j \in \{0, 1, \dots, d-1\}$ and $x_0 \in \mathbb{R}^+$, $\lambda_j(x_0) = \rho(x_0)\mathrm{e}^{\mathrm{i}2\pi \mathrm{i} j/d}$ is a simple root of the characteristic polynomial $\phi(\lambda) = \det(\lambda \boldsymbol{I} - \boldsymbol{P}(x_0))$. The coefficients of this polynomial $\phi(\lambda)$ are polynomials in analytic functions, as the entries of $\boldsymbol{P}(x)$ are analytic by assumption. The implicit function theorem for algebraic functions [Wil88, pp. 66-67] then implies that for each $x_0 > 0$ there exists an $\epsilon > 0$ such that $\lambda_j(x)$ is an Eigenvalue of $\boldsymbol{P}(x)$ and $\lambda_j(x)$ is an analytic function for all $x \in \mathbb{C}$ with $|x - x_0| < \epsilon$. As proven in [Wil88, pp. 66-67], the associated Eigenvectors are also analytic functions in x in a neighborhood around the positive real axis. □

Note that a continuous continuation of the Perron root to the whole complex plane does not in general have to be unique. This is because the Perron-Frobenius theorem only guarantees the uniqueness of the Perron root for positive x. For all other $x \in \mathbb{C} \setminus \mathbb{R}^+$ the Eigenvalues might intersect, meaning that the implicit function theorem does not hold, and thus a unique analytic extension of the root is not possible anymore. The following example illustrates the generic case, showing that the Eigenvalues at the origin $x = 0$.

Example 7.25. *Consider the graph G with cost-enumerator matrix*

$$\boldsymbol{P}_G(x) = \begin{pmatrix} x^2 & x \\ x & x^2 \end{pmatrix}.$$

The two Eigenvalues of this matrix are given by $\lambda_1(x) = x + x^2$ and $\lambda_2(x) = -x + x^2$. We directly see that $\lambda_1(0) = \lambda_2(0) = 0$ and thus, the two Eigenvalues intersect in the origin.

In fact, for any cost-enumerator matrix, all Eigenvalues intersect at $x = 0$, since $\boldsymbol{P}_G(0) = \boldsymbol{0}$. Besides the spectral radius $\rho(x)$ of the cost enumerator matrix, we are also interested in its derivative. This is because the derivative appears as a component of the critical point equation, see, e.g., Theorem 7.12 and it can be further used to analyze the convexity of $\rho(x)$.

Lemma 7.26. *Let $\boldsymbol{P}(x)$ be a matrix with spectral radius $\rho(x)$, whose entries are analytic functions in $x \in \mathbb{C}$. Further, let $\boldsymbol{P}(x)$ be irreducible with period d for all $x \in \mathbb{R}^+$. Then, the Eigenvalues $\lambda_j(x)$ of $\boldsymbol{P}(x)$ of maximum modulus and the associated right and left Eigenvectors $\boldsymbol{u}_j(x)$ and $\boldsymbol{v}_j(x)$, normalized to $\boldsymbol{v}_j(x)\boldsymbol{u}_j^{\mathrm{T}}(x) = 1$, are analytic in a neighborhood around $\mathbb{R}^+$ and it holds that*

$$\boldsymbol{v}_j(x) \frac{\partial \boldsymbol{P}(x)}{\partial x} \boldsymbol{u}_j^{\mathrm{T}}(x) = \frac{\partial \lambda_j(x)}{\partial x}.$$

[4] A function is analytic at a point x, if it can locally be represented by a power series. A function is analytic in a domain if and only if it is complex differentiable in the same domain, see, e.g. [FS09, Thm. IV.1]

Proof. To start with, denote by $\lambda_j(x)$ the Eigenvalues of maximum modulus whose existence is guaranteed from Lemma 7.24. The differentiability of $\lambda_j(x)$ and $\boldsymbol{u}_j(x), \boldsymbol{v}_j(x)$ then follows from the analyticity of $\lambda_j(x)$ proven in Lemma 7.24. Differentiating $\boldsymbol{P}(x)\boldsymbol{u}_j^{\mathrm{T}}(x) = \lambda_j(x)\boldsymbol{u}_j^{\mathrm{T}}(x)$ on both sides with respect to x yields

$$\boldsymbol{P}(x)\frac{\partial \boldsymbol{u}_j^{\mathrm{T}}(x)}{\partial x} + \frac{\partial \boldsymbol{P}(x)}{\partial x}\boldsymbol{u}_j^{\mathrm{T}}(x) = \lambda_j(x)\frac{\partial \boldsymbol{u}_j^{\mathrm{T}}(x)}{\partial x} + \frac{\partial \lambda_j(x)}{\partial x}\boldsymbol{u}_j^{\mathrm{T}}(x).$$

Multiplying with $\boldsymbol{v}_j(x)$ from the left, one obtains

$$\boldsymbol{v}_j(x)\frac{\partial \boldsymbol{P}(x)}{\partial x}\boldsymbol{u}_j^{\mathrm{T}}(x) = \frac{\partial \lambda_j(x)}{\partial x}.$$

as desired. □

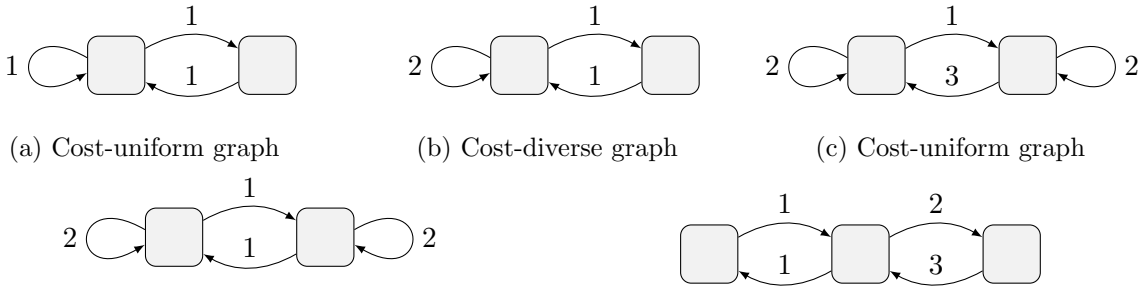
Note that, although $\lambda_1(x) = \rho(x)$ for all $x \in \mathbb{R}^+$ (where we denote by $\lambda_1(x)$ the Perron root), the spectral radius $\rho(x)$ is not necessarily differentiable with respect to complex-valued x, as $\rho(x)$ is equal to the *magnitude* of the largest Eigenvalue. Even though the Eigenvalues $\lambda_j(x)$ of maximum modulus are analytic in a neighborhood around the real axis, the magnitude function is not an analytic function on the whole complex plane.

7.5 Spectral Properties of Cost-Diverse Graphs

An important aspect and requirement of Theorem 7.14 is that the graph G is cost-diverse. Roughly speaking, by Definition 7.4, cost-diversity means that the spectrum of average costs assumed by paths connecting two vertices does not approach a constant for large path lengths. This property is important in the derivation of the asymptotics of the bivariate series $N_{G,v}(t, \alpha t)$ as it entails a smooth behavior of the series in the parameter α. Conversely, if G is cost-uniform, there is in fact only a single value for α for which the series $N_{G,v}(t, \alpha t)$ does not vanish eventually. Note that [KMR00] found that graphs for which all edge costs are the same have this discontinuous behavior, however these are not the only graphs that fall into this category. We generalize this observation and show that cost-diversity,[5] is the precise graph property that distinguishes between a smooth and discontinuous behavior. We further extend the notion of cost-diversity to cost period c (Definition 7.5) and show that it relates to a very special structure of the cost-enumerator matrix $\boldsymbol{P}_G(x)$, when x is rotated in multiples of $2\pi/c$ along the complex circle. The following examples illustrate the cost-diversity property.

Example 7.27. *Figure 7.2 illustrates cost-diversity on different graphs. Figure 7.2a is a graph with constant cost and thus all paths of length m have cost exactly m. The graph is therefore cost-uniform. Figure 7.2b on the other hand is cost-diverse; there are two paths of length 2 from the left vertex to itself having cost either 2 or 4. Figure 7.2c shows a cost-uniform graph, since any cycle of length m from the left vertex to itself has cost $2m$; any cycle of length m from the right vertex to itself has cost $2m$; any path of length m from the left to the right vertex has cost $2m-1$; and, any path of length m from the right to the left vertex has cost $2m+1$. The graph in Figure 7.2d describes a graph with cost period 2. This is because the cost of any cycle in the graph is a multiple of 2. Further, the costs of all paths connecting the left and right vertex have costs*

[5] Cost-diversity has been shown in [Liu20] to entail desirably properties on the Perron root.



(a) Cost-uniform graph (b) Cost-diverse graph (c) Cost-uniform graph

(d) Cost-diverse graph with cost period 2 (e) Graph with period 2 and cost period 3

Figure 7.2: Examples of graphs illustrating cost-diversity and cost periodicity

congruent to 1 *modulo* 2. *Thus, the graph has cost period* 2. *Similarly, the cost of all length-*m *cycles inside the graph in Figure 7.2e have costs congruent to* m *modulo* 3. *The same is true for paths connecting the left and middle vertex. Paths from these vertices to the right have costs congruent to* $m+1$ *modulo* 3 *and* $m+2$ *for the other direction. Thus, the graph has cost period* 3.

Connections between cost-diversity and the spectral radius will be integral to Theorem 7.14. As we will see, the coboundary condition defined in Definition 7.6 arises in a variety of results related to the Perron-Frobenius Theorem and is very useful for proving several of our results. We further need the notion of log-log-convexity, which is defined as follows

Definition 7.28. *Let* $I \subseteq \mathbb{R}^+$ *be an interval and* $f(x) : I \mapsto \mathbb{R}^+$ *a function on that interval. We call* $f(x)$ *log-log-convex, if* $\ln f(\mathrm{e}^s)$ *is convex in the variable* s *on the interval* $\ln I \triangleq \{\ln x : x \in I\}$. *Analogously, we introduce the notions of strict log-log-convexity and log-log-linearity.*

With these definitions we arrive at the central statement of this section. We will prove the next result using Lemmas 7.31 7.34, 7.39, and Corollaries 7.37 and 7.38. Figure 7.3 depicts the roadmap for our subsequent derivations that establish the connections between the graph properties, spectral properties of $\boldsymbol{P}_G(x)$ and the singularities of $\boldsymbol{F}_G(x, y)$.

Lemma 7.29. *Let* G *be a strongly connected graph. The following statements are equivalent.*

(a) The graph G *has cost period* c .

(b) The graph G *satisfies the* c*-periodic coboundary condition.*

(c) For all $x \in \mathbb{C}$ *and* $k \in \mathbb{Z}$, $\rho_G(x\mathrm{e}^{2\pi \mathrm{i} k/c}) = \rho_G(x)$.

Further, if G *is cost-uniform, then the spectral radius* $\rho_G(x)$ *is log-log-linear on* $x \in \mathbb{R}^+$. *If* G *is cost-diverse, then the spectral radius* $\rho_G(x)$ *is strictly log-log-convex on* $x \in \mathbb{R}^+$.

Remark 7.30. *Lemma 7.34 and Corollary 7.37 below state that for any strongly connected graph* G *the equation* $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(x)$ *either has a finite number of solutions* $\phi_k = 2\pi k/c$, *when* G *is cost-diverse with cost period* c, *or holds for all* ϕ *when* G *is cost-uniform. Because cost period zero refers to cost-uniformity, this means statement* (c) *in Lemma 7.29 covers the case of invariance of the spectral radius on the complex circle. Furthermore, we see that when* $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(x)$ *has an infinite number of solutions, then the set of such solutions comprise the full complex circle.*

We proceed with proving the lemmas required for the derivation of Lemma 7.29.

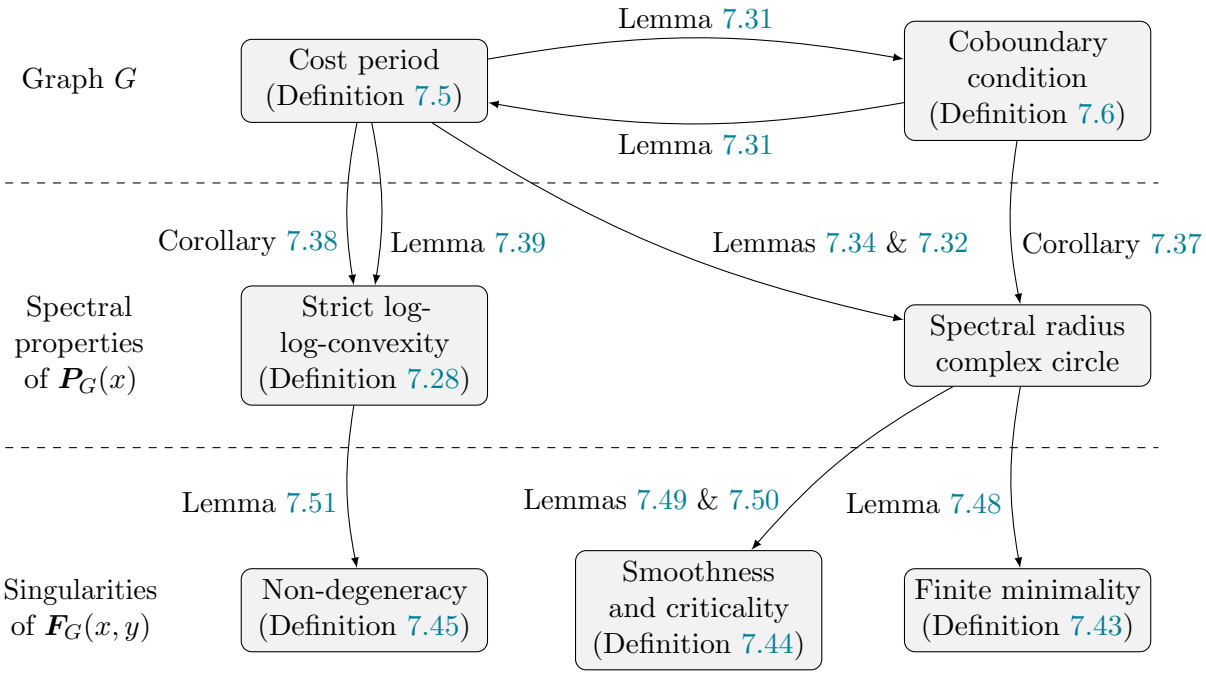


Figure 7.3: Relationships between the properties of strongly connected graphs, the nature of the cost-enumerator spectrum, and the singularities of the generating function.

7.5.1 Equivalance of Cost-Diversity and Coboundary Condition

We start with proving the equivalence of cost-uniformity and the coboundary condition. For convenience, we will say that two integers are congruent modulo 0 if and only if they are equal. The following result is a generalization of the equivalence between the coboundary condition and cost-uniformity observed in [Liu20] to arbitrary cost periods c.

Lemma 7.31. *Let G be a strongly connected graph. Then, G has cost period c if and only if it fulfills the c-periodic coboundary condition.*

Proof. **The c-periodic coboundary condition implies cost period c**: Let $\boldsymbol{p} = (e_1, e_2, \dots e_m)$ be a path from vertex v_i to vertex v_j with path cost $\tau(\boldsymbol{p}) = \sum_{k=1}^{m} \tau(e_k)$. Suppose $\boldsymbol{p}$ is represented by the vertex sequence $v_i = v_{i_0} \to v_{i_1} \to \cdots \to v_{i_m} = v_j$. The coboundary condition allows the path cost to be written as

$$\begin{aligned}\tau(\boldsymbol{p}) &= \sum_{k=1}^{m} (b + B(v_{i_k}) - B(v_{i_{k-1}})) + zc \\ &= mb + B(v_j) - B(v_i) + zc\end{aligned}$$

for some integer $z \in \mathbb{Z}$. Thus, the costs of all paths of length m that connect v_i and v_j are congruent modulo c and, by definition, the graph G has cost period c.

Cost period c implies c-periodic coboundary condition: We will start by showing that there exists $b \in \mathbb{Q}$ such that the cost of any cycle $\boldsymbol{p}$ of length m satisfies $\tau(\boldsymbol{p}) \equiv bm \pmod{c}$. Let $v_1 \in \mathcal{V}$ and let $\boldsymbol{p}_1$ be a cycle at vertex v_1 of length m_1. Such a cycle exists by strong

connectivity of the graph. Suppose $\tau(\boldsymbol{p}_1) = t_1$. Now, let $v_2 \in \mathcal{V}$ and let $\boldsymbol{p}_2$ be a cycle of length m_2 at vertex v_2 with cost $\tau(\boldsymbol{p}_2) = t_2$. Denote by $g_{12} = \gcd(m_1, m_2)$ the greatest common divisor of m_1 and m_2. Strong connectivity of G implies there is a path $\boldsymbol{p}_{1\to 2}$ from v_1 to v_2 with length $n \geq 1$ and cost $\tau(\boldsymbol{p}_{1\to 2}) = t$. Define the path $\boldsymbol{p}$ comprising m_2/g_{12} repetitions of cycle $\boldsymbol{p}_1$ followed by $\boldsymbol{p}_{1\to 2}$, and the path $\boldsymbol{p}'$ comprising $\boldsymbol{p}_{1\to 2}$ followed by m_1/g_{12} repetitions of the cycle $\boldsymbol{p}_2$. The paths $\boldsymbol{p}$ and $\boldsymbol{p}'$ both have length $m_1 m_2/g_{12} + n$. So as G has cost period c, $m_2 t_1/g_{12} + t = \tau(\boldsymbol{p}) = \tau(\boldsymbol{p}') + zc = m_1 t_2/g_{12} + t + zc$ for some $z \in \mathbb{Z}$. This implies that

$$m_2 t_1 - m_1 t_2 = g_{12} z c.$$

This puts us in the position to employ a variation of the Chinese Remainder Theorem in Lemma A.9, which implies that there exists $b \in \mathbb{Q}$ such that any cycle $\boldsymbol{p}$ of length m in G has a cost $\tau(\boldsymbol{p}) \equiv mb \pmod{c}$, which is congruent to mb modulo c.

Now, define a function $B : \mathcal{V} \to \mathbb{R}$ as follows. Set $B(v_1) = 0$. For a vertex $v_i \neq v_1$, choose a path $\boldsymbol{p}_{1\to i}$ from v_1 to v_i of length $n \geq 1$, and define $B(v_i) = \tau(\boldsymbol{p}_{1\to i}) - nb$. We claim that $(B(v_i) \bmod c)$ is independent of the chosen path $\boldsymbol{p}_{1\to i}$. To see this, suppose $\boldsymbol{p}'_{1\to i}$ and $\boldsymbol{p}''_{1\to i}$ are two such paths from v_1 to v_i of length n' and n'', respectively, and let $\boldsymbol{p}_{i\to 1}$ be a path of length p from v_i to v_1. The cycle $\boldsymbol{p}' = (\boldsymbol{p}'_{1\to i}, \boldsymbol{p}_{i\to 1})$ has length $n' + p$, so $\tau(\boldsymbol{p}') = (n' + p)b + z'c$, where $z' \in \mathbb{Z}$. Similarly, the cycle $\boldsymbol{p}'' = (\boldsymbol{p}''_{1\to i}, \boldsymbol{p}_{i\to 1})$ has length $n'' + p$ and cost $\tau(\boldsymbol{p}'') = (n'' + p)b + z''c$ for some $z'' \in \mathbb{Z}$. Then $\tau(\boldsymbol{p}'_{1\to i}) = \tau(\boldsymbol{p}') - \tau(\boldsymbol{p}_{i\to 1}) = (n' + p)b + z'c - \tau(\boldsymbol{p}_{i\to 1})$ and $\tau(\boldsymbol{p}''_{1\to i}) = \tau(\boldsymbol{p}'') - \tau(\boldsymbol{p}_{i\to 1}) = (n'' + p)b + z''c - \tau(\boldsymbol{p}_{i\to 1})$. It follows that

$$\tau(\boldsymbol{p}_{i\to 1}) = (n' + p)b + z'c - \tau(\boldsymbol{p}'_{1\to i}) = (n'' + p)b + z''c - \tau(\boldsymbol{p}''_{1\to i})$$

from which we conclude that

$$\tau(\boldsymbol{p}'_{1\to i}) - n'b = \tau(\boldsymbol{p}''_{1\to i}) - n''b + (z' - z'')c.$$

This confirms that by definition, $(B(v_i) \bmod c)$ is independent of the choice of path from v_1 to v_i.

Finally, let $e \in \mathcal{E}$ be an edge from vertex v_i to vertex v_j, and let $\boldsymbol{p}_{j\to 1}$ denote a path from vertex v_j to v_1 of length q. Consider the cycle $\boldsymbol{p}_1 = (\boldsymbol{p}_{1\to i}, e, \boldsymbol{p}_{j\to 1})$, with cost $\tau(\boldsymbol{p}_1) = (n + 1 + q)b + z_1 c$ for some $z_1 \in \mathbb{Z}$. Noting that $\tau(\boldsymbol{p}_1) = \tau(\boldsymbol{p}_{1\to i}, e) + \tau(\boldsymbol{p}_{j\to 1})$, and using the fact that $\tau(\boldsymbol{p}_{1\to i}, e) = B(v_j) + (n + 1)b + z_j c$ for some $z_j \in \mathbb{Z}$, we find

$$\tau(\boldsymbol{p}_{j\to 1}) = (n + 1 + q)b + z_1 c - (B(v_j) + (n + 1)b + z_j c) = qb - B(v_j) + (z_1 - z_j)c.$$

We can also write $\tau(\boldsymbol{p}_1) = \tau(\boldsymbol{p}_{1\to i}) + \tau(e) + \tau(\boldsymbol{p}_{j\to 1})$, implying that

$$\begin{aligned}
\tau(e) &= \tau(\boldsymbol{p}_1) - (\tau(\boldsymbol{p}_{1\to i}) + \tau(\boldsymbol{p}_{j\to 1})) \\
&= (n + 1 + q)b - ((B(v_i) + nb) + (qb - B(v_j))) + (z_j - z_i)c \\
&= b + B(v_j) - B(v_i) + (z_j - z_i)c.
\end{aligned}$$

This confirms that the c-periodic coboundary condition holds. □

7.5.2 Cost Period and Spectral Properties

We next show that cost-diversity implies that there can only be a finite number of solutions to $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(x)$ over $0 \leq \phi < 2\pi$. In fact, we will prove a stronger statement, that for all $x \in \mathbb{R}^+$, the exact and only solutions are $\phi_k = 2\pi k/c$. This property is vital as it implies that the minimal singularities of the generating functions will be finitely minimal. We start with an auxiliary result on the structure of the cost-enumerator matrix.

Lemma 7.32. *Let G be a strongly connected graph with cost period c. Then, for all $x \in \mathbb{C}$, $k \in \mathbb{Z}$,*

$$\boldsymbol{P}_G\left(x\mathrm{e}^{2\pi \mathrm{i}k/c}\right) = \mathrm{e}^{2\pi \mathrm{i}kb/c}\boldsymbol{D}_k^{-1}\boldsymbol{P}_G(x)\boldsymbol{D}_k,$$

where $\boldsymbol{D}_k$ is a diagonal matrix with entries $[\boldsymbol{D}_k]_{jj} = \mathrm{e}^{2\pi \mathrm{i}kB(v_j)/c}$ and b, $B(v_j)$ are defined by the coboundary condition. Denoting by $\lambda_1(x), \dots, \lambda_{|\mathcal{V}|}(x)$ the Eigenvalues of $\boldsymbol{P}_G(x)$, it holds that

$$\lambda_j\left(x\mathrm{e}^{2\pi \mathrm{i}k/c}\right) = \mathrm{e}^{2\pi \mathrm{i}kb/c}\lambda_j(x),$$

Proof. The graph G satisfies the c-periodic coboundary condition by Lemma 7.31. Hence, there exists a constant b and functions $B : \mathcal{V} \mapsto \mathbb{R}$ such that for any two vertices v_i and v_j, each edge e from v_i to v_j has cost $\tau(e)$, which can be written as

$$\tau(e) = b + B(v_j) - B(v_i) + z_e c,$$

for some integer $z_e \in \mathbb{Z}$. Abbreviate $\phi_k \triangleq 2\pi k/c$. By Definition 7.7, the entries of the cost-enumerator matrix are given by

$$\left[\boldsymbol{P}_G\left(x\mathrm{e}^{\mathrm{i}\phi_k}\right)\right]_{ij} = \sum_{\substack{e\in\mathcal{E}:\ \mathsf{init}(e)=v_i,\\ \mathsf{term}(e)=v_j}} x^{\tau(e)}\mathrm{e}^{\mathrm{i}\phi_k\tau(e)} = [\boldsymbol{P}_G(x)]_{ij}\mathrm{e}^{\mathrm{i}\phi_k(b+B(v_j)-B(v_i))}.$$

Introducing the diagonal matrix $\boldsymbol{D}_k$ with entries $[\boldsymbol{D}_k]_{ii} = \mathrm{e}^{\mathrm{i}\phi_k B(v_i)}$, we can decompose the cost-enumerator matrix to

$$\boldsymbol{P}_G\left(x\mathrm{e}^{\mathrm{i}\phi_k}\right) = \mathrm{e}^{\mathrm{i}\phi_k b}\boldsymbol{D}_k^{-1}\boldsymbol{P}_G(x)\boldsymbol{D}_k.$$

The second part of the lemma directly follows from the similarity[6] of the matrices $\boldsymbol{P}_G(x\mathrm{e}^{\mathrm{i}\phi_k})$ and $\mathrm{e}^{\mathrm{i}\phi_k b}\boldsymbol{P}_G(x)$ proven in the first part of the lemma. □

We will use this property to prove that the Eigenvalues of $\boldsymbol{P}_G(x)$ have a very special structure, when x varies over the complex circle. We continue with another auxiliary result that will serve to prove the subsequent result on the spectral structure of $\boldsymbol{P}_G(x)$ on the complex circle.

Lemma 7.33. *Let G be a strongly connected and cost-diverse graph with largest cost period c. Then, there exist two equal-length cycles at the same vertex whose cost difference is precisely c.*

Proof. For convenience we denote in the following for a path $\boldsymbol{p}$ by $\mathsf{init}(\boldsymbol{p})$ the initial vertex of its first edge and by $\mathsf{term}(\boldsymbol{p})$ the terminal vertex of its last edge. To this end, recall the definition of cost period c. Since c is the largest cost period, Definition 7.5 guarantees the existence of $\eta \in \mathbb{N}$ pairs of paths $\boldsymbol{p}_i$, $\boldsymbol{p}_i'$, $i \in [\eta]$, where for each i both paths have the same lengths m_i, $\boldsymbol{p}_i$ and $\boldsymbol{p}_i'$ start in the same vertex $v_i \triangleq \mathsf{init}(\boldsymbol{p}_i) = \mathsf{init}(\boldsymbol{p}_i')$ and end in the same vertex $u_i \triangleq \mathsf{term}(\boldsymbol{p}_i) = \mathsf{term}(\boldsymbol{p}_i')$ such that the greatest common divisor of their cost differences is $\tau(\boldsymbol{p}_i) - \tau(\boldsymbol{p}_i')$ is c. Hence, by Bézout's identity, there exist (possibly negative) integers $z_i \in \mathbb{Z}$ such that

$$\sum_{i=1}^{\eta}(\tau(\boldsymbol{p}_i) - \tau(\boldsymbol{p}_i'))z_i = c$$

[6] Two square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are *similar*, if there exists an invertible diagonal matrix $\boldsymbol{D}$ such that $\boldsymbol{A} = \boldsymbol{D}^{-1}\boldsymbol{B}\boldsymbol{D}$. Similar matrices have the same Eigenvalues with the same multiplicities [HJ12, Cor. 1.3.4].

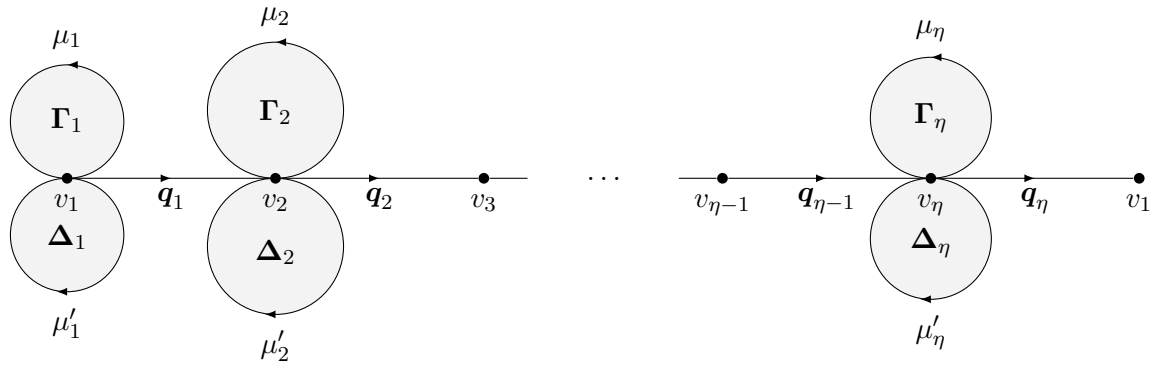


Figure 7.4: Construction of the path $\boldsymbol{\Gamma}$ in the proof of Lemma 7.33

For each i, choose an arbitrary path $\boldsymbol{p}_{u_i \to v_i}$ that connects u_i and v_i and construct two cycles $\boldsymbol{\Gamma}_i = (\boldsymbol{p}_i, \boldsymbol{p}_{u_i \to v_i})$ and $\boldsymbol{\Delta}_i = (\boldsymbol{p}'_i, \boldsymbol{p}_{u_i \to v_i})$ that share the same return path $\boldsymbol{p}_{u_i \to v_i}$ from u_i to v_i. Further choose arbitrary paths $\boldsymbol{q}_i$, $1 \leq i \leq \eta$ connecting v_i and v_{i+1} and v_η and v_1. Now, set $\mu_i = \max\{z_i, 0\}$ and $\mu'_i = \mu_i - z_i$, denote by $\boldsymbol{\Gamma}_i^\mu$, $\mu \in \mathbb{N}_0$ the $\mu-$fold repetition of the cycle $\boldsymbol{\Gamma}_i$ and construct two large cycles $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ by

$$\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1^{\mu_1}, \boldsymbol{\Delta}_1^{\mu'_1}, \boldsymbol{q}_1, \boldsymbol{\Gamma}_2^{\mu_2}, \boldsymbol{\Delta}_2^{\mu'_2}, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_\eta),$$
$$\boldsymbol{\Delta} = (\boldsymbol{\Gamma}_1^{\mu'_1}, \boldsymbol{\Delta}_1^{\mu_1}, \boldsymbol{q}_1, \boldsymbol{\Gamma}_2^{\mu'_2}, \boldsymbol{\Delta}_2^{\mu_2}, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_\eta).$$

That is, $\boldsymbol{\Gamma}$ starts at v_1, circles μ_1 times along $\boldsymbol{\Gamma}_1$, then μ'_1 times along $\boldsymbol{\Delta}_1$, then proceed to move along $\boldsymbol{q}_1$ to v_2. There it circles μ_2 times along $\boldsymbol{\Gamma}_2$ and μ'_2 times along $\boldsymbol{\Delta}_2$, and so on, until it moves back from v_η to v_1 along $\boldsymbol{q}_\eta$. Similarly, $\boldsymbol{\Delta}$ is created. For a visualization of the construction of the cycle $\boldsymbol{\Gamma}$, see Figure 7.4. Notice that μ_i and μ'_i are guaranteed to be non-negative by their definitions. Computing the cost difference of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$, one obtains

$$\tau(\boldsymbol{\Gamma}) - \tau(\boldsymbol{\Delta}) = \sum_{i=1}^{\eta} (\mu_i \tau(\boldsymbol{p}_i) + \mu'_i \tau(\boldsymbol{p}'_i)) - \sum_{i=1}^{\eta} (\mu'_i \tau(\boldsymbol{p}_i) + \mu_i \tau(\boldsymbol{p}'_i)) = \sum_{i=1}^{\eta} (\tau(\boldsymbol{p}_i) - \tau(\boldsymbol{p}'_i)) z_i = c.$$

Hence, there exist two cycles at the vertex v_1 of the same length m whose cost is precisely p. □

Lemmas 7.32 and 7.33 can be combined to prove the following result on the structure of the spectral radius on the complex circle.

Lemma 7.34. *Let G be a strongly connected and cost-diverse graph with largest cost period c. Then, for any $x \in \mathbb{R}^+$, there are precisely c solutions $\phi_k = 2\pi k/c$, $k \in \{0, 1, \ldots, c-1\}$ to the equation $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(x)$, in the interval $0 \leq \phi < 2\pi$. For all other ϕ, $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) < \rho_G(x)$.*

Proof. By Lemma 7.32, for all $k \in \{0, 1, \ldots, c-1\}$, and $j \in [|\mathcal{V}|]$, we have $\lambda_j(x\mathrm{e}^{\mathrm{i}\phi_k}) = \mathrm{e}^{\mathrm{i}\phi_k b} \lambda_j(x)$, which implies that $\rho_G(x\mathrm{e}^{\mathrm{i}\phi_k}) = \rho_G(x)$.

We proceed with proving that for all other values of ϕ the spectral radius $\rho_G(x\mathrm{e}^{\mathrm{i}\phi})$ is strictly less than $\rho_G(x)$. We start with the observation that for any $\phi \in \mathbb{R}$, we have

$$|[\boldsymbol{P}_G(x\mathrm{e}^{\mathrm{i}\phi})]_{ij}| = \left| \sum_{e \in \mathcal{E}: \mathsf{init}(e)=v_i, \mathsf{term}(e)=v_j} (x\mathrm{e}^{\mathrm{i}\phi})^{\tau(e)} \right| \leq \sum_{e \in \mathcal{E}: \mathsf{init}(e)=v_i, \mathsf{term}(e)=v_j} x^{\tau(e)} = [\boldsymbol{P}_G(x)]_{ij}.$$

By Wielandt's theorem [Mey00, Sec. 8.3] (Theorem 7.21) it follows that the spectral radius satisfies $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) \leq \rho_G(x)$ with equality if and only if there exist $\theta, \theta_1, \theta_2, \ldots, \theta_{|\mathcal{V}|}$ such that

$$\boldsymbol{P}_G(x\mathrm{e}^{\mathrm{i}\phi}) = \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{D}^{-1}\boldsymbol{P}_G(x)\boldsymbol{D},$$

where $\boldsymbol{D}$ is a diagonal matrix with entries $[\boldsymbol{D}]_{jj} = \mathrm{e}^{\mathrm{i}\theta_j}$. It therefore suffices to prove that this equality can not be fulfilled for any $0 \leq \phi < 2\pi$ that is not equal to some ϕ_k. Powering the above equation to $m \in \mathbb{N}$, it follows that

$$\boldsymbol{P}_G^m(x\mathrm{e}^{\mathrm{i}\phi}) = \mathrm{e}^{\mathrm{i}m\theta}\boldsymbol{D}^{-1}\boldsymbol{P}_G^m(x)\boldsymbol{D}.$$

In particular, the entry i, j of this equation reads as

$$[\boldsymbol{P}_G^m(x\mathrm{e}^{\mathrm{i}\phi})]_{ij} = \mathrm{e}^{\mathrm{i}(m\theta+\theta_j-\theta_i)}[\boldsymbol{P}_G^m(x)]_{ij}$$

and it follows that

$$|[\boldsymbol{P}_G^m(x\mathrm{e}^{\mathrm{i}\phi})]_{ij}| = |[\boldsymbol{P}_G^m(x)]_{ij}| = [\boldsymbol{P}_G^m(x)]_{ij}.$$

Denote now by $\mathcal{P}_{ij}(m) = \{\boldsymbol{p} = (e_1, \ldots, e_m) : \mathsf{init}(e_1) = v_i, \mathsf{term}(e_m) = v_j\}$ the set of paths of length m from v_i to v_j. It is well-known [KMR00] that $[\boldsymbol{P}_G^m(x)]_{ij} = \sum_{\boldsymbol{p}\in\mathcal{P}_{ij}(m)} x^{\tau(\boldsymbol{p})}$. By Lemma 7.33, for a graph with largest cost period c there exists a length m and a vertex v_i such that there are two cycles of length m at v_i whose cost differs by exactly c. Let in the following m, v_i be such that they fulfill this property and denote by $\tau, \tau + c$ the costs that are assumed by these two cycles. Thus, the polynomial $[\boldsymbol{P}_G^m(x)]_{ii}$ contains the sum of at least two monomials x^τ and $x^{\tau+c}$, each with integer-valued coefficients. Now, recall that the triangle inequality of a sum of complex numbers is tight if and only if the complex angles of all summands agree. Therefore, if $2\pi\phi c$ is not an integer multiple of 2π, then $|[\boldsymbol{P}_G^m(x\mathrm{e}^{\mathrm{i}\phi})]_{ii}| < [\boldsymbol{P}_G^m(x)]_{ii}$ and the claim follows. □

Conversely, also for cost-uniform graphs, the Eigenvalues of $\boldsymbol{P}_G(x)$ have a special structure that can be derived explicitly. The following lemma proves this fact.

Lemma 7.35. *Let $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ be a strongly connected graph that satisfies the coboundary condition. Then, for all $j \in [|\mathcal{V}|]$ and $x \in \mathbb{C}$, the Eigenvalues $\lambda_1(x), \ldots, \lambda_{|\mathcal{V}|}(x)$ of $\boldsymbol{P}_G(x)$ are*

$$\lambda_j(x) = \lambda_j(1)x^b,$$

where b is the constant of the coboundary condition.

Proof. The graph G satisfies the coboundary condition by assumption. Hence, there exists a constant b and functions $B : \mathcal{V} \mapsto \mathbb{R}$ such that for any two vertices v_i and v_j, each edge from v_i to v_j has cost τ_{ij}, which can be written as

$$\tau_{ij} = b + B(v_j) - B(v_i).$$

Further, the number of edges from v_i to v_j is precisely $[\boldsymbol{P}_G(1)]_{ij}$. It follows that each entry $[\boldsymbol{P}_G(x)]_{ij}$ of $\boldsymbol{P}_G(x)$ is equal to

$$[\boldsymbol{P}_G(x)]_{ij} = [\boldsymbol{P}_G(1)]_{ij}x^{\tau_{ij}} = [\boldsymbol{P}_G(1)]_{ij}x^{b+B(v_j)-B(v_i)}.$$

Introducing the diagonal matrix $\boldsymbol{D}(x)$ with entries $[\boldsymbol{D}(x)]_{ii} = x^{B(v_i)}$, we can decompose the cost-enumerator matrix to

$$\boldsymbol{P}_G(x) = x^b \boldsymbol{D}^{-1}(x) \boldsymbol{P}_G(1) \boldsymbol{D}(x).$$

Thus, the characteristic polynomial $\phi(\lambda, x)$ of the cost-enumerator matrix $\boldsymbol{P}_G(x)$ becomes

$$\begin{aligned}\det(\lambda \boldsymbol{I} - \boldsymbol{P}_G(x)) &= \det(\lambda \boldsymbol{I} - x^b \boldsymbol{D}^{-1}(x) \boldsymbol{P}_G(1) \boldsymbol{D}(x)) \\ &= \det(\boldsymbol{D}) \det(\lambda \boldsymbol{I} - x^b \boldsymbol{D}^{-1}(x) \boldsymbol{P}_G(1) \boldsymbol{D}(x)) \det \boldsymbol{D}^{-1} \\ &\overset{(a)}{=} \det(\lambda \boldsymbol{I} - x^b \boldsymbol{P}_G(1)),\end{aligned}$$

where in (a) we used the multiplicativity of the determinant. Thus, $\phi(\lambda, x) = x^{b|\mathcal{V}|}\phi(x^{-b}\lambda, 1)$. Since the Eigenvalues of $\boldsymbol{P}_G(x)$ are precisely the roots of the characteristic polynomial, we can identify $\lambda_j(x)$ as the roots of $\phi(\lambda, x)$ and $\lambda_j(1)$ as the roots of $\phi(\lambda, 1)$. By a variable substitution, it follows that $\lambda_j(x) = \lambda_j(1)x^b$ for all $j \in [|\mathcal{V}|]$ and $x \in \mathbb{C}$.

□

Lemma 7.35 illustrates that the Eigenvalues of cost-uniform graphs have the very special structure of being monomials in x. All Eigenvalues share the same exponent b from the coboundary condition and their coefficient is given by the corresponding Eigenvalue of the matrix $\boldsymbol{P}_G(1)$. The following example illustrates Lemma 7.35.

Example 7.36. *Consider the cost-uniform graph from Figure 7.2c on Page 140. We can verify, by analyzing the cost of the edges which are self-loops, that the constant from the coboundary condition is given by $b = 2$. Computing the Eigenvalues, we obtain $\lambda_1(x) = 2x^2$ and $\lambda_2(x) = 0$, confirming the statement from Lemma 7.35.*

This puts us in the position to prove the converse to Lemma 7.34. That is, we can show that if a graph is cost-uniform, or equivalently, satisfies the coboundary condition, then the spectral radius is invariant on the complex circle.

Corollary 7.37. *Let G be a strongly connected graph that satisfies the coboundary condition. Then, for any $x \in \mathbb{C}$, $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(x)$, for all $0 \leq \phi < 2\pi$.*

Proof. The corollary directly follows from Lemma 7.35, using that the coboundary condition implies that for all $x \in \mathbb{C}$,

$$\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(1)|x\mathrm{e}^{\mathrm{i}\phi}|^b = \rho_G(1)|x|^b = \rho_G(x).$$

□

For illustrative purposes, an alternative proof of Corollary 7.37 that does not use Lemma 7.35 is presented in Appendix A.4.

7.5.3 Cost-Diversity and Strict Log-Log-Convexity

We proceed with discussing log-log-convexity of the spectral radius. This property will help in several places to prove Theorem 7.16. First, we show that for cost-uniform graphs the spectral radius is log-log-linear on the real axis.

Corollary 7.38. *Let G be a strongly connected graph. If G is cost-uniform, then $\rho_G(x)$ is log-log-linear on the interval $x \in \mathbb{R}^+$.*

Proof. By Lemma 7.35, for all real-valued $s \in \mathbb{R}$, $\ln \rho_G(\mathrm{e}^s) = \ln(\rho_G(1)) + bs$, which is a linear function in s. □

Another way to prove Corollary 7.38 without the employment of Lemma 7.35 is to use Nussbaum's theorem [Nus86], which we present for the interested reader in Appendix A.4. We turn towards proving the converse to the previous corollary, i.e., we show that if the graph G is cost-diverse, then $\rho_G(x)$ is strictly log-log-convex. Notice that the (non-strict) log-log convexity of the Perron root is known from classical results on irreducible matrices [Coh78; Coh81; Kin61].

Lemma 7.39. *Let G be a strongly connected, cost-diverse graph. Then $\rho_G(x)$ is strictly log-log-convex for all $x \in \mathbb{R}^+$.*

The proof will make use of the following lemma [SWB06, Thm. 1.37].

Lemma 7.40 ([SWB06, Thm. 1.37])**.** *Let $\boldsymbol{P}(s)$, $s \in \mathbb{R}$ be an irreducible matrix, such that all non-zero entries are log-convex functions of s. Then the spectral radius $\rho(s)$ of $\boldsymbol{P}(s)$ is log-convex. If additionally, at least one entry of $\boldsymbol{P}(s)$ is strictly log-convex, then $\rho(s)$ is strictly log-convex.*

Proof of Lemma 7.39. Consider the m-th power $\boldsymbol{P}_G^m(x)$ of $\boldsymbol{P}_G(x)$. We know [KMR00] that, denoting $\mathcal{P}_{ij}(m)$ as the set of paths of length m from v_i to v_j, the entry i, j of the matrix $\boldsymbol{P}_G^m(x)$ is given by $[\boldsymbol{P}_G^m(x)]_{ij} = \sum_{\boldsymbol{p} \in \mathcal{P}_{ij}(m)} x^{\tau(\boldsymbol{p})}$. We will show that this entry is strictly log-log-convex, if there exist two paths of length m from v_i to v_j with different costs. Taking the second derivative of the log-log expression, we obtain

$$\frac{\partial^2}{\partial s^2} \ln([\boldsymbol{P}_G^m(\mathrm{e}^s)]_{ij}) = \frac{\sum_{\boldsymbol{p}} \mathrm{e}^{s\tau(\boldsymbol{p})} \sum_{\boldsymbol{p}} \tau(\boldsymbol{p})^2 \mathrm{e}^{s\tau(\boldsymbol{p})} - \left(\sum_{\boldsymbol{p}} \tau(\boldsymbol{p}) \mathrm{e}^{s\tau(\boldsymbol{p})}\right)^2}{\left([\boldsymbol{P}_G^m(\mathrm{e}^s)]_{ij}\right)^2}.$$

Identifying the vectors $\boldsymbol{u} = (\mathrm{e}^{s\tau(\boldsymbol{p})/2} : \boldsymbol{p} \in \mathcal{P}_{ij}(m))$ and $\boldsymbol{v} = (\tau(\boldsymbol{p})\mathrm{e}^{s\tau(\boldsymbol{p})/2} : \boldsymbol{p} \in \mathcal{P}_{ij}(m))$ that both have length $|\mathcal{P}_{ij}(m)|$, the numerator is equal to $(\boldsymbol{u} \cdot \boldsymbol{u})(\boldsymbol{v} \cdot \boldsymbol{v}) - (\boldsymbol{u} \cdot \boldsymbol{v})^2$, where $\boldsymbol{u} \cdot \boldsymbol{v}$ denotes the inner product of the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. The numerator is therefore non-negative by Cauchy-Schwarz inequality, see, e.g., [HJ12, Ch. 0.6.3], and thus the entries $[\boldsymbol{P}_G^m(x)]_{ij}$ are either 0 or positive and log-convex. Further, due to the cost-diversity of the graph G, there exist m, i, j such that there exist two paths of length m from v_i to v_j with different costs and thus $\boldsymbol{u}$ and $\boldsymbol{v}$ are linearly independent. In this case, Cauchy-Schwarz inequality holds with strict inequality and thus the numerator is positive, which implies that $[\boldsymbol{P}_G^m(\mathrm{e}^s)]_{ij}$ is strictly convex in s.

The spectral radius of $\boldsymbol{P}_G^m(\mathrm{e}^s)$ is given by $\rho_G^m(\mathrm{e}^s)$. With Lemma 7.40 it follows that $\rho_G^m(\mathrm{e}^s)$ is strictly log-convex. Since powering with positive integers does not change log-convexity, $\rho_G(\mathrm{e}^s)$ is also strictly log-convex. By definition, $\rho_G(x)$ is thus strictly log-log-convex. □

7.6 Multivariate Singularity Analysis

The main step in proving Theorem 7.14 is showing that the prerequisites of [Mel21, Thm. 5.1 and 9.1] are fulfilled, which requires exhibiting certain properties of the singularities. We start by deriving the generating function and reviewing the properties of the generating function required to understand [Mel21, Thm. 5.1 and 9.1]. Afterwards, we prove that these properties apply for the generating functions of the size of the limited-cost follower sets.

7.6.1 Derivation of the Generating Function

The beginning point of the multivariate singularity analysis is the derivation of the generating function of the series $N_{G,v}(t,n)$. Together with a profound analysis of the singularities of the generating function in the proceeding sections, this will allow to use the powerful machinery of analytical combinatorics in several variables. The starting point of our derivations is the following recursion, which is the key observation to derive the generating function of the series $N_{G,v}(t,n)$.

Lemma 7.41. *Let $G = (\mathcal{V}, \mathcal{E}, \tau, \sigma)$ be a deterministic graph. Then, the size of the follower set of any vertex $v \in \mathcal{V}$ obeys the recursion,*

$$N_{G,v}(t,n) = \sum_{e \in \mathcal{E} : \mathsf{init}(e) = v} N_{G,\mathsf{term}(e)}(t - \tau(e), n-1),$$

for all $n > 0$, $t \geq 0$, and $N_{G,v}(t,0) = 1$ for all $t \geq 0$, and, $N_{G,v}(t,n) = 0$ for all $t < 0$ or $n < 0$.

Proof. Denote by $\mathcal{P}_{G,v}(t,n)$ the set of all length-n paths through G that start from vertex v and have cost at most t. By the deterministic property of the graph, $N_{G,v}(t,n) = |\mathcal{P}_{G,v}(t,n)|$. Partition the paths $\mathcal{P}_{G,v}(t,n)$ according to the first traversed edge $e \in \mathcal{E}$ into the distinct parts $\mathcal{P}_{G,v,e}(t,n) = \{\boldsymbol{p} \in \mathcal{P}_{G,v}(t,n) : \boldsymbol{p} = (e, e_2, \dots)\}$, for any $e \in \mathcal{E}$ that emits from v, i.e., $\mathsf{init}(e) = v$. To start with, $\mathcal{P}_{G,v}(t,n) = \bigcup_{e \in \mathcal{E}} \mathcal{P}_{G,v,e}(t,n)$ and the parts are distinct by definition. Now, any path $\boldsymbol{p} \in \mathcal{P}_{G,v,e}(t,n)$ starts by traversing e, which costs $\tau(e)$ and results in the vertex $\mathsf{term}(e)$. Therefore, each path $\boldsymbol{p} \in \mathcal{P}_{G,v,e}(t,n)$ can be assembled by prepending e to some path of cost at most $T - \tau(e)$ and length $n-1$ that starts from $\mathsf{term}(e)$, i.e.

$$\mathcal{P}_{G,v,e}(t,n) = \left\{\boldsymbol{p} = (e, \boldsymbol{p}') : \boldsymbol{p}' \in \mathcal{P}_{G,\mathsf{term}(e)}(t - \tau(e), n-1)\right\}.$$

Thus $|\mathcal{P}_{G,v,e}(t,n)| = N_{G,\mathsf{term}(e)}(t-\tau(e), n-1)$, which proves the recursive statement of the lemma. The initial condition $N_{G,v}(t,0) = 1$ for all $t \geq 0$ comes from the fact that we include the length-0 string in our computations. □

This recursion allows us to derive the exact generating function of the integer sequences $N_{G,v}(t,n)$. Further, we can extract the asymptotic behavior of integer sequences by means of powerful methods in complex analysis [Mel21; PW08]. Note that here we restrict our attention to the series $N_{G,v}(t,n)$, which directly implies the generating function for $N_{G,v}(t)$ because we have $N_{G,v}(t) = \sum_{n \geq 0} N_{G,v}(t,n)$. We proceed with the proof of Lemma 7.10.

Proof of Lemma 7.10. Starting from the recursive expression of $N_{G,v}(t,n)$, we first incorporate the beginning of the recursion and obtain

$$N_{G,v}(t,n) = \sum_{e \in \mathcal{E} : \mathsf{init}(e) = v} N_{G,\mathsf{term}(e)}(t - \tau(e), n-1) + U(t,n),$$

where $U(t,n) = 1$ if $n = 0$ and $t \geq 0$ and 0, otherwise. Multiplying with $x^t y^n$ on both sides and

summing over n, t yields

$$\begin{aligned} F_{G,v}(x,y) &= \sum_{e\in\mathcal{E}:\mathsf{init}(e)=v} \sum_{t,n\geq 0} N_{G,\mathsf{term}(e)}(t-\tau(e),n-1)x^t y^n + \sum_{t,n\geq 0} U(t,n)x^t y^n \\ &= \sum_{e\in\mathcal{E}:\mathsf{init}(e)=v} x^{\tau(e)} y \sum_{t\geq -\tau(e),n\geq -1} N_{G,\mathsf{term}(e)}(t,n)x^t y^n + \sum_{t\geq 0} x^t \\ &\overset{(a)}{=} \sum_{e\in\mathcal{E}:\mathsf{init}(e)=v} x^{\tau(e)} y F_{G,\mathsf{term}(e)}(x,y) + \frac{1}{1-x}, \end{aligned}$$

where in (a) we used that $N_{G,v}(t,n) = 0$ for any $t < 0$ or $n < 0$ and the fact that the geometric series is equal to $1/(1-x)$. Combining the generating functions of all vertices into one vector $\boldsymbol{F}_G(x,y)$, we obtain

$$\boldsymbol{F}_G(x,y) = y\boldsymbol{P}_G(x)\boldsymbol{F}_G(x,y) + \frac{1}{1-x}\mathbf{1}^{\mathrm{T}}.$$

Rearranging the above equality gives the claim. □

7.6.2 Analytical Combinatorics in Several Variables

We shortly present and review the ingredients required to invoke the ACSV results [Mel21]. To this end, for reasons of clarity, we present the definitions for the case, where we wish to compute the asymptotic behavior of $N(\alpha_1 t, \alpha_2 t)$, as $t \to \infty$. Notice that in our setup $\alpha_1 = 1$ and $\alpha_2 = \alpha$. We further restrict the definitions to the bivariate case, as this is our case of interest. Further, we will assume that the generating function has the form $F(x,y) = Q(x,y)/H(x,y)$ for two polynomials $Q(x,y)$ and $H(x,y)$. We start with the notion of singularities of a generating function.

Definition 7.42 ([Mel21, Def. 3.5])**.** *A point $(x_0, y_0) \in \mathbb{C}^2$ is called a* singularity *of $F(x,y)$, if $F(x,y)$ is unbounded in any neighborhood around (x_0, y_0).*

Similar to the univariate case, a sufficient condition for a point to be a singularity is that $H(x_0, y_0) = 0$ and $Q(x_0, y_0) \neq 0$. The important singularities will be those, which are minimal. They are defined as follows.

Definition 7.43 ([Mel21, Def. 3.9])**.** *A point $(x_0, y_0) \in \mathbb{C}^2$ is called a* minimal singularity *of $F(x,y) = Q(x,y)/H(x,y)$, if it is a singularity of $F(x,y)$ and there exists no other singularity $(x', y') \in \mathbb{C}^2$ with $|x'| < |x|$ and $|y'| < |y|$.*

A minimal singularity is called *finitely minimal* [Mel21, Def. 5.6], if there exist only a finite number of singularities with the same coordinate-wise modulus. In contrast to the case of univariate generating functions, not all minimal singularities contribute to the asymptotic behavior of the series under consideration. The following notion of critical points helps to separate those singularities which are important for the asymptotic expansion.

Definition 7.44 ([Mel21, Def. 5.4])**.** *A point $(x_0, y_0) \in \mathbb{C}^2$ is called a* critical point *of $F(x,y) = Q(x,y)/H(x,y)$, if*

$$H(x,y) = \alpha_2 x H_x(x_0, y_0) = \alpha_1 y H_y(x_0, y_0)$$

If further either $H_x(x_0, y_0) \neq 0$ or $H_y(x_0, y_0) \neq 0$, then the point is a smooth critical point*.*

As remarked in [Mel21, Def. 5.5], if the direction (α_1, α_2) is positive, then any minimal smooth critical point is a point that contributes to the asymptotics.

Definition 7.45 ([Mel21, Def. 5.7])**.** *Let $(x_0, y_0) \in \mathbb{C}^2$ be a smooth critical point, assume w.l.o.g. $H_y(x_0, y_0) \neq 0$ and let $g(x)$ be the explicit function characterizing the singularities $(x, g(x))$ in a neighborhood around x_0. The point (x_0, y_0) is called a* non-degenerate critical point*, if the Hessian matrix $\mathcal{H}$ of*

$$\phi(\theta) = \ln\left(\frac{g(x_0 \mathrm{e}^{\mathrm{i}\phi})}{g(x_0)}\right) + \mathrm{i}\alpha_1/\alpha_2\theta$$

is non-singular at $\theta = 0$.

We now introduce the theorems from ACSV that we will use to prove our asymptotic results, taking care to note what properties we need to establish to determine asymptotics. These theorems are specialized to the bivariate case we consider.

First, we see a theorem for 'smooth' asymptotics, which applies when asymptotics are determined by a smooth critical singularity. We will apply this result in the regime when $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$.

Theorem 7.46 ([Mel21, Thm. 5.1])**.** *Let $\alpha_1, \alpha_2 > 0$ and let $Q(x, y), H(x, y)$ be coprime polynomials such that the generating function $F(x, y) = Q(x, y)/H(x, y)$ admits a power series expansion $F(x, y) = \sum_{t,n\geq 0} N(t, n)x^t y^n$. Suppose that the system of polynomial equations*

$$H(x, y) = \alpha_2 x H_x(x, y) - \alpha_1 y H_y(x, y) = 0 \tag{7.1}$$

admits a finite number of solutions, exactly one of which $(x_0, y_0) \in \mathbb{C}^2$ is minimal. Suppose further that (x_0, y_0) has non-zero coordinates, $H_y(x_0, y_0) \neq 0$, and (x_0, y_0) is non-degenerate. Then, as $t \to \infty$,

$$N(t\alpha_1, t\alpha_2) = x_0^{-t\alpha_1} y_0^{-t\alpha_2} t^{-1/2} \frac{1}{\sqrt{2\pi\alpha_2 \mathcal{H}_{x_0,y_0}}} \left(\frac{-Q(x_0, y_0)}{y_0 H_y(x_0, y_0)} + O\left(\frac{1}{t}\right)\right) \tag{7.2}$$

when $t(\alpha_1, \alpha_2) \in \mathbb{N}^2$.

Theorem 7.46 has been extended to the case, when the critical point equations (7.1) admit a finite set of minimal singularities, which all have the same coordinate-wise modulus. Provided that all such points fulfill the conditions of Theorem 7.46, an asymptotic expansion of the integer series is obtained by summing over the right-hand side of (7.2) at each of the singularities [Mel21, Cor. 5.2]. In order to compute the asymptotic expansion in the smooth case, we thus need to verify the following properties. First, we need to characterize the minimal points that satisfy (7.1) and show that H_y does not vanish at these points. Second, the points need to be non-degenerate and the numerator should be non-zero to guarantee a dominant asymptotic term.

The other case of interest is the *multiple-point* case where two smooth branches of the singular set collide. In this case, the asymptotic behavior is obtained using the following theorem.

Theorem 7.47 ([Mel21, Prop. 9.1 and Thm. 9.1])**.** *Let $\alpha_1, \alpha_2 > 0$ and let $Q(x, y), H(x, y)$ be coprime polynomials such that $F(x, y) = Q(x, y)/H(x, y)$ admits a power series expansion $F(x, y) = \sum_{t,n\geq 0} N(t, n)x^t y^n$. Suppose that (x_0, y_0) is a strictly minimal point, and near (x_0, y_0) the zero set of H is locally the union of the sets defined by the vanishing of polynomials $R(x, y)$ and $S(x, y)$ such that $R(x_0, y_0) = S(x_0, y_0) = 0$ and the gradients of R and S are linearly independent*

at (x_0, y_0) (in particular, both gradients must be non-zero so each of the zero sets are locally smooth near (x_0, y_0)). If there exist $\nu_1, \nu_2 > 0$ such that

$$(\alpha_1, \alpha_2) = \nu_1 \left(1, \frac{y_0 R_y(x_0, y_0)}{x_0 R_x(x_0, y_0)}\right) + \nu_2 \left(1, \frac{y_0 S_y(x_0, y_0)}{x_0 S_x(x_0, y_0)}\right)$$

then, as $t \to \infty$,

$$N(t\alpha_1, t\alpha_2) = x_0^{-t\alpha_1} y_0^{-t\alpha_2} \frac{Q(x_0, y_0)}{|\det \mathbf{H}|} + O(\delta^t)$$

where $0 < \delta < x_0^{-\alpha_1} y_0^{-\alpha_2}$ and

$$\mathbf{H} = \begin{pmatrix} x_0 R_x(x_0, y_0) & y_0 R_y(x_0, y_0) \\ x_0 S_x(x_0, y_0) & y_0 S_y(x_0, y_0) \end{pmatrix}.$$

We will apply this theorem, if α is in the range $0 < \alpha < \alpha_G^{\mathsf{lo}}$. As in the smooth case, if there exist a finite number of singularities with the same coordinate-wise modulus as (x_0, y_0) that all satisfy the conditions of Theorem 7.47, then we get an asymptotic expansion by summing the asymptotic contributions of each. Applying Theorem 7.47 is easier than applying Theorem 7.46, as we only need to prove the existence of the constants ν_1 and ν_2.

7.6.3 Singularity and Critical Point Analysis

The main challenge in proving Theorem 7.14 is showing that the prerequisites of Theorems 7.46 and 7.47 are fulfilled. We establish the necessary conditions through a careful study of the singularities of our generating functions

$$\boldsymbol{F}_G(x, y) = \frac{1}{1 - x} \cdot (\boldsymbol{I} - y\boldsymbol{P}_G(x))^{-1} \mathbf{1}^{\mathrm{T}}.$$

We can write $\boldsymbol{F}_G(x, y) = \boldsymbol{Q}_G(x, y)/H_G(x, y)$ for a polynomial vector $\boldsymbol{Q}_G(x, y) = \operatorname{adj}(\boldsymbol{I} - y\boldsymbol{P}_G(x))\mathbf{1}^{\mathrm{T}}$ and polynomial $H_G(x, y) = (1 - x)\det(\boldsymbol{I} - y\boldsymbol{P}_G(x))$. In particular, all entries of $\boldsymbol{F}_G(x, y)$ share the same denominator, which allows us to analyze the crucial properties such as minimal and critical points once instead of for each coordinate. According to Definition 7.9, we always work with respect to the diagonal $(\alpha_1, \alpha_2) = (1, \alpha)$, and this direction is assumed when discussing notions like critical points and non-degeneracy.

The first step in our multivariate singularity analysis is to identify those singularities, which are minimal, i.e., for which there exists no other singularity that has a smaller magnitude in all coordinates (see Definition 7.43).

Lemma 7.48. *Let G be a strongly connected and cost-diverse graph with largest period d and largest cost period c. The points*

$$\{(x_0, 1/\rho_G(x_0)) : 0 < x_0 < 1\} \cup \{(1, y_0) : y_0 \in \mathbb{C}, |y_0| \leq 1/\rho_G(1)\}$$

are minimal singularities of each coordinate of $\boldsymbol{F}_G(x, y)$. All other minimal singularities are

$$\left(x_0 \mathrm{e}^{\mathrm{i}2\pi k/c}, \mathrm{e}^{-2\pi \mathrm{i}(kb/c + j/d)}/\rho_G(x_0)\right)$$

for some $0 < x_0 \leq 1$, $k \in \{0, 1, \ldots, c-1\}$, and $j \in \{0, 1, \ldots, d-1\}$, where b is the constant of the c-periodic coboundary condition.

Proof. The singularities of the coordinates of $\boldsymbol{F}_G(x,y)$ are a subset of the solutions to the equation $(1-x)\det(\boldsymbol{I}-y\boldsymbol{P}_G(x))=0$, and any root of the denominator where the numerator does not vanish is a singularity. Using that $\det(\boldsymbol{I}-y\boldsymbol{P}_G(x)) = \prod_j(1-y\lambda_j(x))$, where $\lambda_j(x)$ are the Eigenvalues of $\boldsymbol{P}_G(x)$, the singularities of $\boldsymbol{F}_G$ are thus a subset of the variety

$$\mathcal{X} = \{(x, 1/\lambda_j(x)) : x \in \mathbb{C}, 1 \le j \le |\mathcal{V}|\} \cup \{(1,y) : y \in \mathbb{C}\}.$$

We start by investigating the first set of singularities. Right away, we see that for all $x \in \mathbb{C}$ with $|x| > 1$ the singularities $(x, 1/\lambda_j(x))$ cannot be minimal, since there exists $y \in \mathbb{C}$ such that $(1,y)$ has a coordinate-wise smaller modulus as $(x, 1/\lambda_j(x))$. We thus focus on those singularities with $0 < |x| \le 1$. Due to the fact that the graph G is strongly connected, it follows that $\boldsymbol{P}_G(x_0)$ is irreducible for all $x_0 \in \mathbb{R}^+$ and thus, by the Perron-Frobenius Theorem, has a single real Eigenvalue $\rho_G(x_0)$ of maximum modulus. In the following we identify the Perron-Frobenius Eigenvalue as the first Eigenvalue $\rho_G(x_0) = \lambda_0(x_0)$.

We now show that for all $0 < x_0 \le 1$ the points $(x_0, 1/\rho_G(x_0))$ are minimal singularities. To begin, the numerator of $\boldsymbol{F}_G$ at this point can be expressed as

$$\boldsymbol{Q}_G(x_0, 1/\rho_G(x_0)) = \operatorname{adj}\left(\boldsymbol{I} - \frac{\boldsymbol{P}_G(x_0)}{\rho_G(x_0)}\right)\mathbf{1}^{\mathrm{T}} = \rho_G(x_0)^{1-|\mathcal{V}|}\operatorname{adj}(\rho_G(x_0)\boldsymbol{I} - \boldsymbol{P}_G(x_0))\mathbf{1}^{\mathrm{T}},$$

so an application of Lemma 7.23 shows that the numerator is non-zero, as $\operatorname{adj}(\rho_G(x_0)\boldsymbol{I} - \boldsymbol{P}_G(x_0))$ is either all-positive or all-negative. In particular, these points are singularities of each coordinate and it remains to show minimality. We prove minimality using Proposition 5.4 of [Mel21], which states that a singularity $(x_0, 1/\rho_G(x_0))$ with positive coordinates is minimal if and only if $H_G(tx_0, t/\rho_G(x_0))$ is non-zero for all $0 < t < 1$. The term $(1 - tx_0)$ does not vanish for $0 < t < 1$, so if $H_G(tx_0, t/\rho_G(x_0)) = 0$ then $t/\rho_G(x_0) = 1/\lambda_j(tx_0)$ for some $0 < t < 1$ and $j \ge 1$. However,

$$t/\rho_G(x_0) < 1/\rho_G(x_0) \overset{(a)}{\le} 1/\rho_G(tx_0) \le |1/\lambda_j(tx_0)|,$$

where inequality (a) uses that each entry of $\boldsymbol{P}_G(x_0)$ is monotonically increasing in x_0 and thus $\rho_G(x_0)$ is also monotonically increasing in x_0. Hence $H_G(tx_0, t/\rho_G(x_0))$ does not vanish on $0 < t < 1$ and it follows that any point $(x_0, 1/\rho_G(x_0))$ with $0 < x_0 < 1$ is a minimal singularity.

We next prove that the only other minimal singularities in $\{(x, 1/\lambda_j(x)) : x \in \mathbb{C}, 1 \le j \le |\mathcal{V}|\}$ are as given in the statement of the lemma. To start with, by Theorem 7.20, for each $0 < x_0 \le 1$ there are precisely d simple Eigenvalues $\lambda_0(x_0), \dots, \lambda_{d-1}(x_0)$ with the same modulus as the spectral radius and they are given by

$$\lambda_j(x_0) = \rho_G(x_0)\mathrm{e}^{2\pi\mathrm{i}(j-1)/d}.$$

Due to the similarity of $\boldsymbol{P}_G(x\mathrm{e}^{\mathrm{i}\phi_k})$ and $\mathrm{e}^{\mathrm{i}\phi_k b}\boldsymbol{P}_G(x)$ for all $\phi_k = 2\pi k/c$ and $k \in \{0, 1, \dots, c-1\}$, which was derived in Lemma 7.32, the Eigenvalues of $\boldsymbol{P}_G(x_0\mathrm{e}^{\mathrm{i}\phi_k})$ are given by $\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi_k}) = \mathrm{e}^{\mathrm{i}\phi_k b}\lambda_j(x_0)$. Therefore, for each j and k we obtain one candidate for a minimal singularity,

$$\left(x_0\mathrm{e}^{\mathrm{i}\phi_k}, \mathrm{e}^{-\mathrm{i}(\phi_k b + 2\pi j/d)}/\rho_G(x_0)\right).$$

For all other ϕ that are not integer multiples of $2\pi/c$, the singularities $(x_0\mathrm{e}^{\mathrm{i}\phi}, 1/\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi}))$ are not minimal, as $\rho_G(x_0\mathrm{e}^{\mathrm{i}\phi}) < \rho_G(x_0)$ in this case was proven in Lemma 7.34. Furthermore, all other Eigenvalues $\lambda_j(x)$ with $j \ge d$ have $|\lambda_j(x)| < \rho_G(x)$, which implies that they cannot be minimal.

Finally, we study the singularities in $\{(1, y) : y \in \mathbb{C}\}$. All points $(1, y_0)$, $y_0 \in \mathbb{C}$, $|y_0| \leq 1/\rho_G(1)$ are singularities, since the matrix $\boldsymbol{I} - y_0 \boldsymbol{P}_G(1)$ is invertible. Furthermore, these singularities are minimal due to the fact that $(1, 1/\rho_G(1))$ is minimal as proven above. Conversely, for $|y_0| > 1/\rho_G(1)$ the points $(1, y_0)$ are not minimal due to the existence of the singularities $(x_0, 1/\rho_G(x_0))$. □

Noteworthily, while we have proven that the points $(x_0, 1/\rho_G(x_0))$ are indeed singularities, the same is not necessarily true for the other minimal points. This is because for these points, the numerator is not guaranteed to be non-negative. Next is a statement on the smoothness and criticality of the singularities.

Lemma 7.49. *Let G be a strongly connected and cost-diverse graph with period d and cost period c. For all $x_0 \in \mathbb{R}^+$ with $x_0 \neq 1$ and all $k \in \{0, 1, \dots, c-1\}$, $j \in \{0, 1, \dots, d-1\}$, the points*

$$\left(x_0 \mathrm{e}^{2\pi \mathrm{i} k/c}, \mathrm{e}^{-2\pi \mathrm{i}(kb/c + j/d)}/\rho_G(x_0)\right)$$

are smooth points of $\boldsymbol{F}_G(x, y)$ and critical if and only if $\alpha x_0 \rho_G'(x_0) = \rho_G(x_0)$. Any point $(1, y_0)$ with $y_0 \in \mathbb{C}$ and $|y_0| < \rho_G(1)$ is not a root of $\det(\boldsymbol{I} - y\boldsymbol{P}_G(x))$ and thus is a smooth point that is never critical.

Proof. Abbreviate for convenience $\phi_k \triangleq 2\pi k/c$ and $\theta_j \triangleq 2\pi j/d$. We start by verifying that for all $x_0 \in \mathbb{R}^+$ with $x_0 \neq 1$ and $k \in \{0, 1, \dots, c-1\}$, $j \in \{0, 1, \dots, d-1\}$, the points $(x_0 \mathrm{e}^{\mathrm{i}\phi_k}, \mathrm{e}^{-\mathrm{i}(\phi_k b + \theta_j)}/\rho_G(x_0))$ are smooth. By Jacobi's Formula, the partial derivative satisfies

$$\frac{\partial H_G(x, y)}{\partial y} = -(1 - x) \operatorname{tr}\left(\operatorname{adj}(\boldsymbol{I} - y\boldsymbol{P}_G(x))\boldsymbol{P}_G(x)\right).$$

For the rest of this proof we write $\lambda_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k})$ for the d Eigenvalues of $\boldsymbol{P}_G(x_0 \mathrm{e}^{\mathrm{i}\phi_k})$ of maximum modulus, which satisfy $\lambda_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k}) = \mathrm{e}^{\mathrm{i}(\phi_k b + \theta_j)} \lambda_0(x_0)$, where $\lambda_0(x_0) = \rho_G(x_0)$ is the Perron root of $\boldsymbol{P}_G(x_0)$, according to Theorem 7.20 and Lemma 7.32. The corresponding normalized Eigenvectors are $\boldsymbol{u}_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k})$ and $\boldsymbol{v}_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k})$ and plugging in the points of interest, we obtain

$$\left.\frac{\partial H_G(x, y)}{\partial y}\right|_{\substack{x = x_0 \mathrm{e}^{\mathrm{i}\phi_k}, \\ y = 1/\lambda_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k})}} = -(1 - x_0 \mathrm{e}^{\mathrm{i}\phi_k}) \operatorname{tr}\left(\operatorname{adj}(\boldsymbol{I} - \boldsymbol{P}_G(x_0 \mathrm{e}^{\mathrm{i}\phi_k})/\lambda_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k}))\boldsymbol{P}_G(x_0 \mathrm{e}^{\mathrm{i}\phi_k})\right).$$

Here we can use Lemma 7.32 to simplify the cost-enumerator matrix and Lemma 7.23 to find an explicit representation of the adjoint matrix, simplifying the above expression to

$$\begin{aligned} &- c_j(x_0)(1 - x_0 \mathrm{e}^{\mathrm{i}\phi_k})(\lambda_j(x_0))^{1-|\mathcal{V}|} \operatorname{tr}(\mathrm{e}^{\mathrm{i}\phi_k b} \boldsymbol{v}_j(x_0) \boldsymbol{P}_G(x_0) \boldsymbol{u}_j^{\mathrm{T}}(x_0)) \\ = &- c_j(x_0)(1 - x_0 \mathrm{e}^{\mathrm{i}\phi_k})(\lambda_j(x_0))^{1-|\mathcal{V}|} \lambda_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k}), \end{aligned}$$

where $c_j(x_0) \in \mathbb{R} \setminus \{0\}$ is a non-zero constant. This expression is non-zero for all $x_0 \in \mathbb{R}^+$ with $x_0 \neq 1$ and $k \in \{0, 1, \dots, c-1\}$, $j \in \{0, 1, \dots, d-1\}$ and thus the points are smooth.

We now examine when these minimal points are solutions of the critical point equations

$$\alpha x \frac{\partial H_G(x, y)}{\partial x} = y \frac{\partial H_G(x, y)}{\partial y}.$$

The partial derivative of the denominator with respect to x is given by

$$\frac{\partial H_G(x,y)}{\partial x} = -\det(\boldsymbol{I} - y\boldsymbol{P}_G(x)) - (1-x)y \ \operatorname{tr}\left(\operatorname{adj}(\boldsymbol{I} - y\boldsymbol{P}_G(x))\frac{\partial \boldsymbol{P}_G(x)}{\partial x}\right).$$

Evaluating this partial derivative at the points $(x_0 \mathrm{e}^{\mathrm{i}\phi_k}, 1/\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi_k}))$, we obtain

$$\left.\frac{\partial H_G(x,y)}{\partial x}\right|_{\substack{x=x_0\mathrm{e}^{\mathrm{i}\phi_k},\\ y=1/\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi_k})}} = -(1-x_0\mathrm{e}^{\mathrm{i}\phi_k}) \operatorname{tr}\left(\operatorname{adj}(\boldsymbol{I} - \boldsymbol{P}_G(x_0\mathrm{e}^{\mathrm{i}\phi_k})/\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi_k}))\boldsymbol{P}'_G(x_0\mathrm{e}^{\mathrm{i}\phi_k})\right),$$

where $\boldsymbol{P}'_G(x)$ is the partial derivative of the cost-enumerator matrix with respect to x. Here we use that $\det(\boldsymbol{I} - y\boldsymbol{P}_G(x))$ evaluated at these points is 0, as $\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi_k})$ is an Eigenvalue of $\boldsymbol{P}_G(x_0\mathrm{e}^{\mathrm{i}\phi_k})$. Similar to the case of the derivative with respect to y, we simplify this expression to

$$\begin{aligned} &- c_j(x_0)(1-x_0\mathrm{e}^{\mathrm{i}\phi_k})(\lambda_j(x_0))^{1-|\mathcal{V}|}\mathrm{e}^{\mathrm{i}\phi_k(b-1)} \operatorname{tr}(\boldsymbol{v}_j(x_0)\boldsymbol{P}'_G(x_0)\boldsymbol{u}_j^{\mathrm{T}}(x_0)) \\ &\overset{(a)}{=} - c_j(x_0)(1-x_0\mathrm{e}^{\mathrm{i}\phi_k})(\lambda_j(x_0))^{1-|\mathcal{V}|}\lambda'_0(x_0)\mathrm{e}^{\mathrm{i}(\phi_k(b-1)+\theta_j)} \end{aligned}$$

where, in the first step, we used that $\boldsymbol{P}'_G(x_0\mathrm{e}^{\mathrm{i}\phi_k}) = \mathrm{e}^{\mathrm{i}\phi_k(b-1)}\boldsymbol{D}_k^{-1}\boldsymbol{P}'_G(x_0)\boldsymbol{D}_k$ according to Lemma 7.32, and equality (a) follows from an application of Lemma 7.26. Substituting our expressions for the partial derivatives into the critical point equations shows that the critical point equations simplify to $\alpha x_0\lambda'_0(x_0) = \lambda_0(x_0)$. Since $\lambda_0(x_0) = \rho_G(x_0)$, the first part of the lemma follows.

The singularities $(1, y_0)$ with $|y_0| < \rho_G(1)$ are not roots of $\det(\boldsymbol{I} - y\boldsymbol{P}_G(x))$ as ρ_G is an Eigenvalue of $\boldsymbol{P}_G$ of largest modulus. Thus, near these points the zero set of the denominator is locally the zero set of the factor $1-x$ and is therefore smooth (algebraically, the partial derivative with respect to x is non-zero at these points). These points can never be critical because the partial derivative of $H_G(x,y)$ with respect to y vanishes at any such point. □

Notice that the derivative $\rho'_G(x)$ in the statement of Lemma 7.49 should crucially be understood with respect to real-valued x. The complex derivative does not necessarily exist, since the spectral radius is the largest magnitude of all Eigenvalues, $\rho_G(x) = |\lambda_0(x)|$, and the magnitude function is not complex differentiable on the whole complex plane.

Lemma 7.50. *Let G be a strongly connected and cost-diverse graph. Then, the critical point equation $\alpha x\rho'_G(x) = \rho_G(x)$ has a positive real solution x_0 if and only if*

$$\lim_{x\to\infty}\frac{\rho_G(x)}{x\rho'_G(x)} < \alpha < \lim_{x\to 0^+}\frac{\rho_G(x)}{x\rho'_G(x)}.$$

This solution, if it exists, is unique among all positive real x. If $\alpha > \rho_G(1)/\rho'_G(1)$ then $x_0 < 1$ and if $\alpha < \rho_G(1)/\rho'_G(1)$, then $x_0 > 1$.

Proof. Since $\rho_G(x) > 0$ for $x \in \mathbb{R}^+$, we can rewrite the equation we are trying to solve as $f(x) = 1$, where $f(x) \triangleq \alpha x\rho'_G(x)/\rho_G(x)$. To start we investigate the limit of $f(x)$ as $x \to 0^+$. Note that $f(x) > 0$ for all $x \in \mathbb{R}^+$. Furthermore, the strict log-log-convexity of $\rho_G(x)$ proven in Lemma 7.39 implies that $f'(x) > 0$: strict log-log-convexity of $\rho_G(x)$ means that $\log\rho_G(\mathrm{e}^s)$ is strictly convex in s, and substituting $x = \mathrm{e}^s$ gives

$$\frac{\partial}{\partial x}f(x) = \mathrm{e}^{-s}\frac{\partial}{\partial s}f(\mathrm{e}^s) = \alpha\mathrm{e}^{-s}\frac{\partial}{\partial s}\frac{\mathrm{e}^s\rho'_G(\mathrm{e}^s)}{\rho_G(\mathrm{e}^s)} = \alpha\mathrm{e}^{-s}\frac{\partial^2}{\partial s^2}\log\rho_G(\mathrm{e}^s) > 0.$$

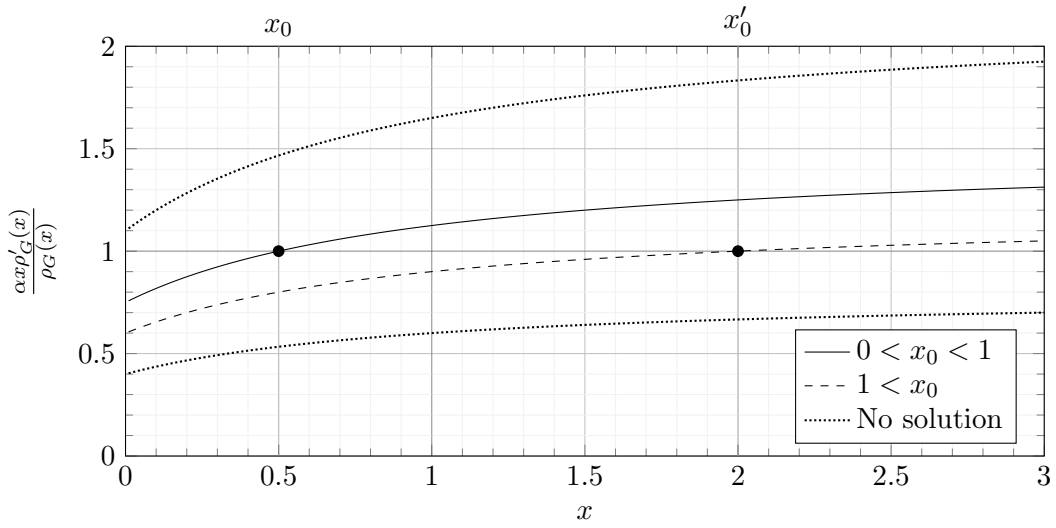


Figure 7.5: Visualization of the solutions of the critical point equation for $\rho_G(x) = x + x^2$. The illustrated cases correspond to different values of α.

Since $f'(x) > 0$ and $f(x) > 0$ we see that $f(x)$ is a bounded and decreasing function as $x \to 0$ from above, and the monotone convergence theorem implies $\lim_{x\to 0^+} f(x)$ exists. Consequently, if $\alpha < \lim_{x\to 0^+} \rho_G(x)/(x\rho_G'(x))$ then $\lim_{x\to 0^+} f(x) < 1$, as both limits exist. Notice that we allow the upper-bound on α to diverge to ∞, in which case we can take α as large as desired. This can happen, for example, when there exists a cycle of weight 0 in G. Similarly, the limit $\lim_{x\to\infty} 1/f(x)$ exists, as $1/f(x)$ is decreasing and positive. Hence, if $\alpha > \lim_{x\to\infty} \rho_G(x)/(x\rho_G'(x))$ then $\lim_{x\to\infty} 1/f(x) < 1$.

To summarize, under our conditions on α we have $\lim_{x\to\infty} f(x) < 1$ and $\lim_{x\to 0^+} f(x) > 1$. By the intermediate value theorem, there is at least one solution to $f(x) = 1$ in $x \in \mathbb{R}^+$. This solution is unique, due to the strict monotonicity of f coming from $f'(x) > 0$. We further see that if α is not within these boundaries, there will be no solution in $x \in \mathbb{R}^+$ due to this monotonicity.

If $\alpha\rho_G'(1) > \rho_G(1)$ then $f(1) > 1$, and the solution to $f(x) = 1$ must occur at $x_0 < 1$. Similarly, if $\alpha\rho_G'(1) < \rho_G(1)$ then $f(1) < 1$ and it follows that $x_0 > 1$. For a visualization, see Figure 7.5. □

Another requirement of Theorem 7.46 is the following non-degeneracy of the singularities.

Lemma 7.51. *Let G be a strongly connected and cost-diverse graph with period d and cost period c. For all $x_0 \in \mathbb{R}^+$ and $k \in \{0, 1, \ldots, c-1\}$, $j \in \{0, 1, \ldots, d-1\}$, the points*

$$\left(x_0 \mathrm{e}^{2\pi \mathrm{i} k/c}, \mathrm{e}^{-2\pi \mathrm{i}(kb/c + j/d)}/\rho_G(x_0)\right)$$

are non-degenerate.

Proof. Write $\xi_k \triangleq x_0 \mathrm{e}^{2\pi \mathrm{i} k/c}$. According to Theorem 7.20 and Lemma 7.32, there are d Eigenvalues of maximum modulus $\lambda_j(\xi_k)$ of $\boldsymbol{P}_G(\xi_k)$ that satisfy $\lambda_j(\xi_k) = \mathrm{e}^{\mathrm{i}(\phi_k b + \theta_j)}\rho_G(x_0)$, where $\rho_G(x_0)$ is the Perron root of $\boldsymbol{P}_G(x_0)$. Lemma 5.5 of [Mel21] implies that the quantity $\mathcal{H}$ determining non-degeneracy in the smooth case is the second derivative of

$$\phi(\theta) = \log\left(\frac{\lambda_j(\xi_k)}{\lambda_j(\xi_k \mathrm{e}^{\mathrm{i}\theta})}\right) + \frac{\mathrm{i}\theta}{\alpha}$$

at $\theta = 0$. Differentiating twice with respect to θ gives

$$\frac{\partial^2}{\partial\theta^2}\phi(\theta)\Big|_{\theta=0} \overset{(a)}{=} -\frac{\partial^2}{\partial\theta^2}\log\lambda_j(x_0\mathrm{e}^{\mathrm{i}\theta})\Big|_{\theta=0} = \frac{\partial^2}{\partial s^2}\log\lambda_j(\mathrm{e}^s)\Big|_{s=\log x_0} \overset{(b)}{=} \frac{\partial^2}{\partial s^2}\log\rho_G(\mathrm{e}^s)\Big|_{s=\log x_0} \overset{(c)}{>} 0.$$

In (a) we used Lemma 7.32 to conclude that $\lambda_j(\xi_k\mathrm{e}^{\mathrm{i}\phi}) = \mathrm{e}^{2\pi\mathrm{i}kb/c}\lambda_j(x_0\mathrm{e}^{\mathrm{i}\phi})$. Note that the differentiation to the left and right hand side of (b) should be understood with respect to complex-valued s and real-valued s, respectively, as $\rho_G(\mathrm{e}^s)$ is not complex differentiable in s in general. In (b) we used that for analytic functions, by the definition of complex differentiation, the derivative along the real line equals the complex derivative. Inequality (c) follows from the strict log-log-convexity of $\rho_G(x)$ for $x \in \mathbb{R}^+$, as was proven in Lemma 7.39. □

7.6.4 Proof of Theorem 7.14

The final ingredient to proving Theorem 7.14 is to identify the critical singularities that contribute to the asymptotic expansion, depending on the value of α,.

Lemma 7.52. *Let G be a strongly connected and cost-diverse graph with largest period d and largest cost period c. Let $\lambda_j(x_0) = \mathrm{e}^{2\pi\mathrm{i}j/d}\rho_G(x_0)$ with $j \in \{0, 1, \dots, d-1\}$ denote the d Eigenvalues of maximum modulus of $\boldsymbol{P}_G(x_0)$.*

- *If $0 < \alpha < \alpha_G^{\mathsf{lo}}$, then $(1, 1/\lambda_j(1))$, $j \in \{0, 1, \dots, d-1\}$ are contributing critical points.*
- *If $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$, then $(x_0\mathrm{e}^{2\pi\mathrm{i}k/c}, 1/\lambda_j(x_0\mathrm{e}^{2\pi\mathrm{i}k/c})$, $j \in \{0, 1, \dots, d-1\}$, $k \in \{0, 1, \dots, c-1\}$, with $\alpha x_0\rho_G'(x_0) = \rho_G(x_0)$ are contributing smooth critical points.*

In both cases, there are no contributing points other than those mentioned.

Proof. We first discuss the multiple-point, non-smooth case $0 < \alpha < \alpha_G^{\mathsf{lo}}$. We start by proving that $(x_0, y_0) = (1, 1/\rho_G(1))$ satisfies the conditions of Theorem 7.47. The two surfaces defined by the vanishing of $R(x, y) = 1 - x$ and $S(x, y) = \det(\boldsymbol{I} - y\boldsymbol{P}_G(x))$ intersect at this point. Direct computation shows $R_x(x, y) = -1$ and $R_y(x, y) = 0$, while Jacobi's formula implies $S_x(x, y) = -y\operatorname{tr}(\operatorname{adj}(\boldsymbol{I} - y\boldsymbol{P}_G(x))\boldsymbol{P}_G'(x))$ and $S_y(x, y) = -\operatorname{tr}(\operatorname{adj}(\boldsymbol{I} - y\boldsymbol{P}_G(x))\boldsymbol{P}_G(x))$. Hence, $(1, 1/\rho_G(1))$ is a contributing point if there exist $\nu_1, \nu_2 > 0$ such that

$$\nu_1\left(1, \frac{y_0R_y(x_0, y_0)}{x_0R_x(x_0, y_0)}\right) + \nu_2\left(1, \frac{y_0S_y(x_0, y_0)}{x_0S_x(x_0, y_0)}\right) = \left(\nu_1 + \nu_2, \nu_2\frac{\rho_G(1)}{\rho_G'(1)}\right) = (1, \alpha).$$

We can set $\nu_1 = 1 - \nu_2$, $\nu_2 = \alpha\rho_G'(1)/\rho_G(1)$,which are both positive, due to $\alpha < \rho_G(1)/\rho_G'(1)$ and the required conditions hold. Using the same arguments, the singularities $(1, 1/\lambda_j(x_0))$ also contribute to the asymptotics. The remaining singularities $(x, y) \in \mathbb{C}^2$ with the same coordinate-wise modulus $(|x|, |y|) = (1, 1/\rho_G(1))$ are smooth, however, by [Mel21, Cor. 5.6], none of them are critical as $(1, 1/\rho_G(1))$ is not critical by Lemma 7.50.

We now move to the smooth case $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$. Lemmas 7.48,7.49,7.50, 7.51 show that the point $(x_0, 1/\rho_G(x_0))$ where $0 < x_0 < 1$ and $\alpha x_0\rho_G'(x_0) = \rho_G(x_0)$ is unique and a smooth, finitely minimal, critical and non-degenerate singularity and is thus contributing. Furthermore, all other singularities with the same coordinate-wise modulus, which are $(x_0\mathrm{e}^{2\pi\mathrm{i}k/c}, 1/\lambda_j(x_0\mathrm{e}^{2\pi\mathrm{i}k/c})$ for some $k, j \in \mathbb{Z}$, fulfill these properties as well. □

We are finally ready to prove Theorem 7.14 by combining Lemmas 7.48,7.49,7.50, 7.51, and Lemma 7.52 with Theorems 7.46 and 7.47.

Proof of Theorem 7.14. We differentiate between the two cases $0 < \alpha < \alpha_G^{\mathsf{lo}}$ and $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$. In the first case, the non-smooth singularity $(1, 1/\rho_G(1))$ and those with the same coordinate-wise modulus are the singularities that determine the asymptotic behavior. In the second case, the singularities $(x_0, 1/\rho_G(x_0))$ with $0 < x_0 < 1$ and $\alpha x_0 \rho_G'(x_0) = \rho_G(x_0)$, and those with the same coordinate-wise modulus, are the ones contributing.

We start with the multiple-point, non-smooth case $0 < \alpha < \alpha_G^{\mathsf{lo}}$, aiming to apply Theorem 7.47 with the extension [Mel21, Cor. 9.1]. For any $x_0 \in \mathbb{R}^+$ let $\lambda_j(x_0) = \mathrm{e}^{2\pi \mathrm{i} j/d}\rho_G(x_0)$ with $j \in \{0, 1, \dots, d-1\}$ denote the d Eigenvalues of maximum modulus of $\boldsymbol{P}_G(x_0)$ and let $\boldsymbol{u}_j(x_0)$ and $\boldsymbol{v}_j(x_0)$ be the corresponding right and left Eigenvectors. By Lemma 7.52, Theorem 7.47 is applicable for the contributing singularities $(1, 1/\lambda_j(1))$ and it remains to compute the required terms. The numerator of the generating function is given by

$$\boldsymbol{Q}_{G,v}(1, 1/\lambda_j(1)) = \lambda_j(1)^{1-|\mathcal{V}|} \operatorname{adj}(\lambda_j(1)\boldsymbol{I} - \boldsymbol{P}_G(1))\mathbf{1}^{\mathrm{T}} \overset{(a)}{=} c_j(1)\lambda_j(1)^{1-|\mathcal{V}|}\boldsymbol{u}_j^{\mathrm{T}}(1)\boldsymbol{v}_j(1)\mathbf{1}^{\mathrm{T}},$$

where (a) follows from an application of Lemma 7.23. Similarly, we obtain for the numerator

$$\det \mathbf{H} = -\frac{1}{\lambda_j(1)} S_y(1, 1/\lambda_j(1)) = \frac{1}{\lambda_j(1)} \operatorname{tr}(\operatorname{adj}(\boldsymbol{I} - \boldsymbol{P}_G(1)/\lambda_j(1))\boldsymbol{P}_G(1)) = c_j(1)\lambda_j(1)^{1-|\mathcal{V}|}.$$

Plugging these results into the expressions of Theorem 7.47 and summing over all contributing points $(1, 1/\lambda_j(1))$ according to [Mel21, Cor. 9.1] proves the first statement of Theorem 7.14.

In the smooth case $\alpha_G^{\mathsf{lo}} < \alpha < \alpha_G^{\mathsf{up}}$, the point $(x_0, 1/\rho_G(x_0))$ where $0 < x_0 < 1$ and $\alpha x_0 \rho_G'(x_0) = \rho_G(x_0)$ is unique and a smooth, finitely minimal, critical and non-degenerate singularity and thus contributing by Lemma 7.52. The same applies to the other singularities with the same coordinate-wise modulus, which are $(x_0 \mathrm{e}^{2\pi \mathrm{i} k/c}, \mathrm{e}^{-2\pi \mathrm{i}(kb/c + j/d)}/\rho_G(x_0))$ for some $k, j \in \mathbb{Z}$. This allows us to invoke the extension [Mel21, Cor. 5.2] of Theorem 7.46. Notice that there may be values of j and k where the numerator vanishes, however we have shown in Lemma 7.48 that this does not occur when $k = j = 0$. Thus, it is possible that the leading asymptotic terms from some of these points vanishes, but the sum of all terms always captures the dominant asymptotic behavior of the sequence under consideration. The quantity $\mathcal{H}$ appearing in the asymptotic expansion was derived in Lemma 7.51. Abbreviating $\phi_k = 2\pi k/c$ in the following, we find that

$$\boldsymbol{Q}_{G,v}(x_0 \mathrm{e}^{\mathrm{i}\phi_k}, 1/\lambda_j(x_0 \mathrm{e}^{\mathrm{i}\phi_k})) = c_j(x_0)\lambda_j(x_0)^{1-|\mathcal{V}|}\boldsymbol{D}_k^{-1}\boldsymbol{u}_j^{\mathrm{T}}(x_0)\boldsymbol{v}_j(x_0)\boldsymbol{D}_k\mathbf{1}^{\mathrm{T}},$$

where we used that for any two square matrices $\boldsymbol{D}$ and $\boldsymbol{P}$, the adjoint of their products is given by $\operatorname{adj}(\boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{D}) = \boldsymbol{D}^{-1}\operatorname{adj}(\boldsymbol{P})\boldsymbol{D}$.

□

7.6.5 Proof of the Other Theorems

We continue with proving the remaining theorems.

Proof of Theorem 7.12. Theorem 7.12 directly follows from Theorem 7.14. By the definition of the capacity, we take the logarithm of the asymptotic expansion $N_{G,v}(t, \alpha t)$ and divide by t. Computing the limit $t \to \infty$, all terms except for the exponential in t vanish. □

Theorems 7.15 and 7.16 can be proven using standard univariate singularity analysis [FS09]. We start with proving Theorem 7.16, which depicts the more general statement of the exact representation of the follower set size.

Proof of Theorem 7.16. By Lemma 7.10, the generating functions of $N_{G,v}(t)$ are given by the fractions $F_{G,v}(x) = Q_{G,v}(x)/H_G(x)$, with the polynomials $Q_{G,v}(x) = [\mathrm{adj}(\boldsymbol{I} - \boldsymbol{P}(x))\mathbf{1}^{\mathrm{T}}]_v$ and $H_G(x) = (1-x)\det(\boldsymbol{I} - \boldsymbol{P}_G(x))$, and thus the singularities are a subset of the solutions to $(1-x)\det(\boldsymbol{I} - \boldsymbol{P}_G(x)) = 0$. Invoking [FS09, Thm. IV.9] then proves Theorem 7.16. Note that in principle not all solutions have to be singularities, as the numerator and denominator are not guaranteed to be coprime. This case is covered by setting $\Pi_{G,v,i}(t) = 0$ for all roots which share common factors with the numerator, in the partial fraction decomposition. □

Proof of Theorem 7.15. Theorem 7.15 follows from Theorem 7.16 by the direct computation of $C_G = \lim_{t\to\infty} \log N_{G,v}(t)/t$ and the fact that the roots of $H_G(x) = (1-x)\prod_j(1-\lambda_j(x))$, where $\lambda_j(x)$ are the Eigenvalues of $\boldsymbol{P}_G(x)$. Since $\boldsymbol{P}_G(x)$ is an irreducible matrix, there is an Eigenvalue, which is equal to the spectral radius and thus the singularity of smallest magnitude of $F_{G,v}(x)$ is that for which $\rho_G(x) = 1$. The numerator at this singularity is non-zero due to Lemma 7.23. □

7.7 Conclusion

In this chapter we have analyzed cost constrained channels, i.e., directed graphs with labeled and costly edges. We have derived the precise asymptotic behavior of the size of the number of limited-cost paths for arbitrary strongly connected and cost-diverse graphs. That is, we have explicitly derived an easily computable function, whose fraction with respect to the true number of followers approaches one for large costs. Our theorems imply explicit expressions for the fixed-length and variable-length capacity, i.e., the exponential growth rate of the number of paths. Interestingly, through the direct derivation of the capacity, we recover a known result on the equivalence of the combinatorial and probabilistic capacity of cost constrained channels. While previous works have shown this equivalence using the central limit theorem, this proof resembles this of Shannon for the case of the variable-length capacity through the expression of the capacity in terms of singularities of a generating function. Establishing an explicit and comprehensible framework to compute both the fixed-length and variable capacity for arbitrary strongly connected graphs, our results do not only open the way for future research but can also directly be employed in suitable applications. For our derivations, we have introduced novel properties of costly graphs that extend the well-known notions of periodicity to costly graphs. Noteworthily, we show that the notion of cost-diversity is the precise property that distinguishes between sharp and smooth behavior of the fixed-length capacity. In our exposition we use results from analytical combinatorics in several variables, which establishes a novel and intriguing connection between noiseless information theory and complex analysis. In order to prove this connection we have build a comprehensive theory that extends results from Perron and Frobenius on irreducible matrices. These results were then related to properties of the singularities of the generating functions of the follower set size, which built the bridge to the theory of analytical combinatorics in several variables.

Chapter 8

Cost-Efficient DNA Synthesis

For DNA-based data storage to become a feasible technology, all aspects of the encoding and decoding pipeline must be optimized. Writing the data into DNA, which is known as DNA synthesis, is currently the most costly part of existing storage systems. In this regard, several recent works suggested methods to optimize the cost-efficiency of the synthesis process. Among these, [Tab+20] developed a method that encodes information through the modification of existing DNA, circumventing the expensive synthesis step. [Jai+20] discussed codes that optimize the writing rate of terminator-free synthesis. Their work was motivated by a novel inexpensive synthesis technology, where the number of nucleotides attached in one step is a random variable, whose mean can be controlled by the synthesis machine. On the other hand, [Mak+21] studied a popular synthesis method that synthesizes many sequences in parallel in a step-by-step fashion using a fixed supersequence. In their work, the authors proposed to divide the strands into batches of sequences such that in each batch, the sequences have a similar structure, and analyze the resulting synthesis times.

As a step toward more efficient synthesis, we study the design of codes that minimize the time and number of required materials needed to produce the DNA strands. We consider a popular synthesis process, which builds many strands in parallel in a step-by-step fashion using a fixed supersequence $\boldsymbol{s}$. The machine iterates through $\boldsymbol{s}$, one nucleotide at a time, and in each cycle, it adds the next nucleotide to a subset of the strands. The synthesis time is determined by the number of iterations, i.e., the length of $\boldsymbol{s}$. In order to improve the cost-efficiency of this synthesis process we introduce so-called synthesis codes that restrict the set of admissible DNA sequences. We derive the maximum amount of information per synthesis cycle that is possibly achievable with this synthesis process, assuming that $\boldsymbol{s}$ is an arbitrary periodic sequence.

Section 8.1 gives a detailed introduction of the synthesis process and introduces the precise problem statement and preliminaries. We proceed with deriving the capacity, i.e., the maximum achievable information rate, measured in bits per synthesis cycle, for a given periodic synthesis sequence $\boldsymbol{s}$, in Section 8.2. This is achieved by constructing a labeled and costly synthesis graph based on $\boldsymbol{s}$ and then relating the sequences that can be synthesized in a given number of cycles to limited-cost paths through this graph, which allows to use results from Chapter 7. As an exemplary application, we compute the capacity for the case of using an alternating synthesis sequence over arbitrary alphabets. In many applications it is desirable to synthesize only sequences which fulfill a certain constraint, such as that of avoiding long runs of homopolymers. Representing a constraint by a directed labeled graph, we show how to extend our results such that we can compute the achievable information rates with a given synthesis sequence and constraint in Section 8.3. To

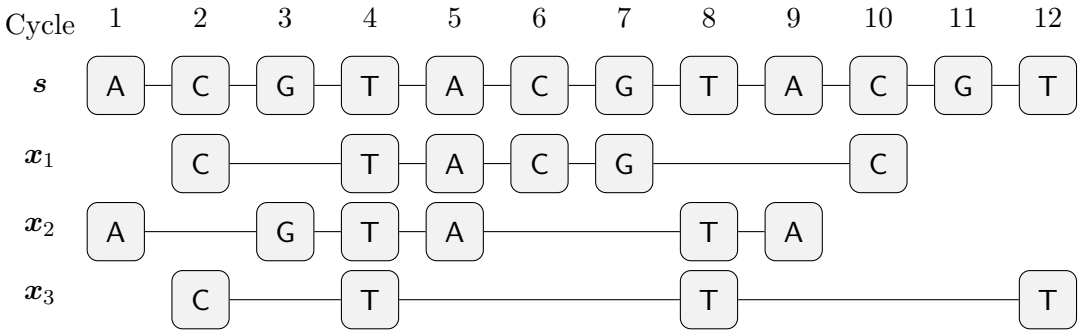


Figure 8.1: Synthesis of three strands $\boldsymbol{x}_1 = (\mathsf{CTACGC}), \boldsymbol{x}_2 = (\mathsf{AGTATA})$, and $\boldsymbol{x}_3 = (\mathsf{CTTT})$ using the synthesis sequence $\boldsymbol{s} = (\mathsf{ACGTACGTACGT})$. The strand $\boldsymbol{x}_1$ is synthesized by attaching the nucleotides in cycles $2, 4, 5, 6, 7, 10$, $\boldsymbol{x}_2$ is synthesized in cycles $1, 3, 4, 5, 8, 9$, and similarly $\boldsymbol{x}_3$ is synthesized in cycles $2, 4, 8, 12$. Hence, $\boldsymbol{x}_1$ can be synthesized in 10 cycles, $\boldsymbol{x}_2$ in 9 cycles and $\boldsymbol{x}_3$ in 12 cycles, when using the synthesis sequence $\boldsymbol{s}$.

this end, we introduce a novel labeled graph product between the costly and labeled synthesis graph and a labeled constraint graph. Finally, we show in Section 8.4, that our results can be used to compute the precise asymptotic expansion of the number of subsequences of arbitrary periodic supersequences.

Parts of the results in this chapter have been published in [Len+20a; Len+21d].

8.1 Preliminaries and Problem Statement

Consider a system, where digital data shall be encoded and synthesized into several DNA strands $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \in \Sigma_q^*$ *in parallel*. These strands can be of equal or different lengths and we will treat both cases separately in the subsequent analysis. The synthesis is performed by the following procedure. First, a synthesis sequence of nucleotides $\boldsymbol{s} = (s_1, s_2, \ldots) \in \Sigma_q^*$ is chosen. The synthesis machine then assembles the DNA strands in a nucleotide-by-nucleotide fashion. To start with, each strand is build from scratch, starting from a strand of length zero. Then, the synthesis machine cycles through the synthesis sequence $\boldsymbol{s}$ and in each cycle $i = 1, 2, \ldots$, for each DNA strand $\boldsymbol{x}_j$, it is possible to either attach the nucleotide s_i to the strand $\boldsymbol{x}_j$ or to perform no action. This procedure continues until all strands are synthesized. By the nature of the synthesis process, a DNA strand $\boldsymbol{x}$ can thus be synthesized in t cycles using the synthesis sequence $\boldsymbol{s}$, if and only if $\boldsymbol{x}$ is a subsequence of $(s_1, \ldots, s_t)$. The synthesis process is visualized in Figure 8.1.

Remark 8.1. *Notice that in general, it is conceivable that the synthesis sequence is chosen based on the DNA strands that shall be synthesized. For example, in the synthesis process visualized in Figure 8.1, it would be possible to skip the synthesis cycle* 11*, as in this step the nucleotide* G *is attached to none of the sequences. More generally, choosing* $\boldsymbol{s}$ *as the shortest common supersequence of all sequences is optimal [Len+20a]. However, it is known that finding the shortest common supersequence is computationally intensive in general [Mai78], especially for a large number of sequences, and thus impractical for large synthesis pools, i.e., for a large number of sequences to be synthesized in parallel. While it is possible to find approximate solutions to the shortest common supersequence problem [NL06], in our setup we are considering the case, where the synthesis sequence is chosen in advance without knowledge of the sequences to be synthesized.*

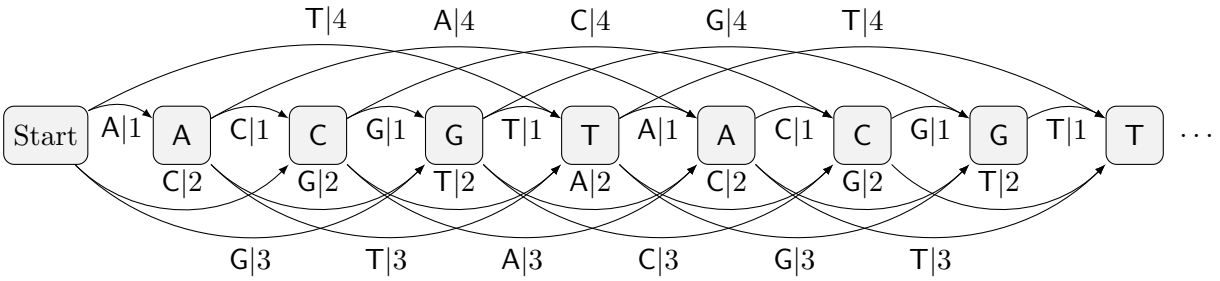


Figure 8.2: Subsequence graph $G_{\mathsf{sub}}(\boldsymbol{s})$ for the synthesis sequence $\boldsymbol{s} = (\mathsf{ACGTACGT...})$. The label of an edge is equal to the symbol of its final vertex and the cost is equal to the number of symbols that are skipped plus one.

We proceed with introducing the subsequence graph, which will be helpful for an abstraction of the synthesis problem. Afterwards, we will turn towards the main problem statement of this chapter, that is the analysis of cost-efficient synthesis.

8.1.1 Synthesis and Subsequence Graphs

It is possible to directly associate with the synthesis process a cost constrained channel that compactly describes the sequences that can be synthesized with a given synthesis sequence $\boldsymbol{s}$ as follows. The system is defined by a so-called *subsequence graph* or *synthesis graph*, which will be constructed such that its associated costly language with maximum cost t is exactly the set of all sequences that can be synthesized with $\boldsymbol{s}$ in at most t cycles. This allows to use the general results from cost constrained channels as in Chapter 7 to infer results for the synthesis process. As it was presented earlier, a DNA strand $\boldsymbol{x}$ can be synthesized using $\boldsymbol{s}$ in time t if and only if it is a subsequence of $\boldsymbol{s}_{1:t}$. We thus introduce the *subsequence graph* $G_{\mathsf{sub}}(\boldsymbol{s})$ of a synthesis sequence $\boldsymbol{s}$, which is defined as follows.

Definition 8.2. *The* subsequence graph $G_{\mathsf{sub}}(\boldsymbol{s})$ *of a synthesis sequence* $\boldsymbol{s} = (s_1, s_2, \dots) \in \Sigma_q^*$ *is a directed, labeled and weighted graph. It has vertices* $\mathcal{V} = \{v_0, v_1, v_2, \dots\}$*, where the vertex* v_i*,* $i \geq 1$ *is associated with the symbol* s_i *and* v_0 *is an auxiliary starting vertex. The vertices* v_i *and* v_j*, are connected by an edge* e*, if* $j > i$ *and* $s_k \neq s_j$ *for all* $i < k < j$*. The label of such an edge is* $\sigma(e) = s_j$ *and the cost is* $\tau(e) = j - i$*.*

Figure 8.2 shows the subsequence graph $G_{\mathsf{sub}}(\boldsymbol{s})$ for $\boldsymbol{s} = (\mathsf{ACGTACGT...})$. By construction of the graph $G_{\mathsf{sub}}(\boldsymbol{s})$ a sequence $\boldsymbol{x}$ can be synthesized using $\boldsymbol{s}$ if and only if there exists a path through $G_{\mathsf{sub}}(\boldsymbol{s})$ whose traversed edge labels generate $\boldsymbol{x}$. Further, the cost of the path is equal to the number of cycles required to synthesize $\boldsymbol{x}$ using $\boldsymbol{s}$. For example, synthesizing the sequence $\boldsymbol{x}_1 = (\mathsf{CTACGC})$ from Figure 8.1 using the synthesis sequence $\boldsymbol{s} = (\mathsf{ACGTACGT...})$ corresponds to the path Start$\to \mathsf{C} \to \mathsf{T} \to \mathsf{A} \to \mathsf{C} \to \mathsf{G} \to \mathsf{C}$ and has cost $2 + 2 + 1 + 1 + 1 + 3 = 10$. Note that although the synthesis process would allow edges from v_i to v_j for all $j > i$, we only draw q outgoing edges to the next appearance of each $\sigma \in \Sigma_q$ for the following two reasons. First, it is desirable to attach a nucleotide to a sequence as soon as possible to minimize the required synthesis cycles. Second, this property is useful as it makes the graph *deterministic* in the sense, that for each vertex v, the labels from any two edges that start from v have distinct labels. This directly leads to the following equivalence.

Proposition 8.3. *The sequence $\boldsymbol{x}$ can be synthesized in t cycles using the synthesis sequence $\boldsymbol{s}$ if and only if it is contained in the cost-t follower set of the starting vertex, i.e., $\boldsymbol{x} \in \mathcal{L}_{G_{\mathsf{sub}}(\boldsymbol{s}),v0}(t)$.*

Proof. A sequence $\boldsymbol{x}$ can be synthesized in t cycles using the synthesis sequence $\boldsymbol{s}$ if and only if $\boldsymbol{x}$ is a subsequence of $(s_1, \dots, s_t)$. We thus prove that $\mathcal{L}_{G_{\mathsf{sub}}(\boldsymbol{s}),v_0}(t)$ is the set of all subsequences of $(s_1, \dots, s_t)$. On the one hand, each word in $\mathcal{L}_{G_{\mathsf{sub}}(\boldsymbol{s}),v_0}(t)$ is a subsequence of $(s_1, \dots, s_t)$ due to the following. Each path $\boldsymbol{p} = (e_1, \dots, e_n)$ that starts at v_0 generates a word $\Sigma_q(\boldsymbol{p}) = (s_{i_1}, \dots, s_{i_n})$, where $1 \leq i_1 < i_2 < \cdots < i_n$ are the indices of the traversed vertices. Further, the cost of each edge is $\tau(e_j) = i_j - i_{j-1}$ and henceforth $\tau(\boldsymbol{p}) = i_1 + (i_2 - i_1) + \cdots + (i_n - i_{n-1}) = i_n$. Thus $i_n \leq t$ and $\Sigma_q(\boldsymbol{p})$ is a subsequence of $(s_1, \dots, s_t)$. On the other hand, if $\boldsymbol{x} = (x_1, \dots, x_n)$ is a subsequence of $(s_1, \dots, s_t)$, we can left align it in the subsequence graph $G_{\mathsf{sub}}(\boldsymbol{s})$ such that it will be generated from a path start starts at v_0. Due to the left alignment, the last symbol x_n will be aligned with a symbol s_i with $i \leq t$ and thus the path has cost at most t. □

Since the subsequence graph is deterministic, for each vertex v of $G_{\mathsf{sub}}(\boldsymbol{s})$, there is one-to-one correspondence between the follower set $\mathcal{L}_{G_{\mathsf{sub}}(\boldsymbol{s}),v}(t)$ and paths that start at v and have cost at most t. Thus, the subsequence graph can be used to efficiently count the number of subsequences of a given supersequence $\boldsymbol{s}$ using standard algorithms that enumerate paths through weighted graphs. However, due to the lack of cycles in this graphs we cannot directly use our results on cost constrained channels in Chapter 7, as those require strongly connected graphs. We therefore fold the subsequence graph for the case of *periodic* synthesis sequences to obtain an equivalent strongly connected graph.

To start with, a periodic synthesis sequence $\boldsymbol{s}$ is a sequence of the form $\boldsymbol{s} = (\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{r}, \dots)$, where $\boldsymbol{r} \in \Sigma_q^M$ is the period of $\boldsymbol{s}$ and $M \in \mathbb{N}$ is the period length. For periodic sequences, the subsequence graph can be folded to obtain a strongly connected graph with M vertices as follows.

Definition 8.4. *The* periodic subsequence graph $G(\boldsymbol{r})$ *of a period $\boldsymbol{r} = (r_1, \dots, r_M) \in \Sigma_q^M$ is a directed, labeled and costly graph. It has M vertices $\mathcal{V} = \{v_1, \dots, v_M\}$, corresponding to the symbols of $\boldsymbol{r}$. There is an edge e from v_i to v_j if either $i < j$ and $r_k \neq r_j$ for all $k \in \{i+1, \dots, j-1\}$ or $i \geq j$ and $r_k \neq r_j$ for all $k \in \{i+1, \dots, M, 1, \dots, j-1\}$. This edge has a label $\sigma(e) = r_j$ and cost*

$$\tau(e) = \begin{cases} j - i, & \text{if } i < j, \\ M - j + i, & \text{if } i \geq j \end{cases}.$$

The periodic subsequence graph $G(\mathsf{ACGT})$ is visualized in Figure 8.3. It is straightforward to verify that the analogue of Proposition 8.3 holds also for the periodic subsequence graph with starting vertex v_M. Note that the periodic subsequence graph has self loops with cost M for symbols that occur exactly once in the period. In fact, we can derive the following properties of the synthesis graph.

Lemma 8.5. *Let $\boldsymbol{r} \in \Sigma_q^M$ be an arbitrary sequence which contains at least two different symbols with corresponding synthesis graph $G(\boldsymbol{r})$. Then $G(\boldsymbol{r})$ is deterministic, strongly connected and cost-diverse. Further, $G(\boldsymbol{r})$ has period 1 and largest cost period M.*

Proof. To start with, $G(\boldsymbol{r})$ is deterministic because all outgoing edges from the same vertex have distinct labels by definition. The vertices $v_1, \dots, v_M$ in $G(\boldsymbol{r})$ correspond to the symbols in $\boldsymbol{r} = (r_1, r_2, \dots, r_M)$. Next, $G(\boldsymbol{r})$ is strongly connected, since there is a Hamiltonian cycle $\boldsymbol{p}_1$ of length M starting at vertex v_1 and connecting v_i to v_{i+1} for $i = 1, 2, \dots, M-1$ and v_M to v_1.

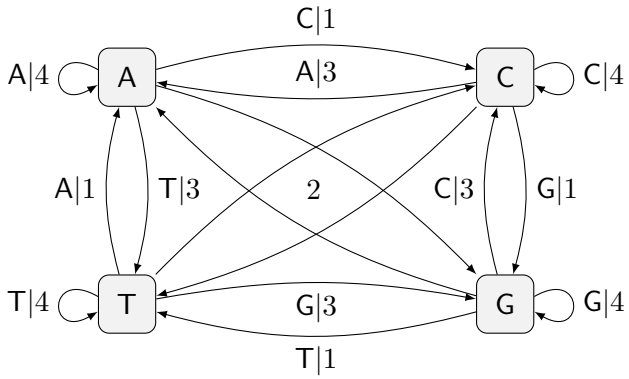


Figure 8.3: Periodic subsequence graph $G(\mathsf{ACGT})$. Hereby, the last symbol of the period, T, can take the role of the starting vertex.

Next, there is also a cycle $\boldsymbol{p}_2$ starting at vertex v_1 of length $M-1$ and cost M. This cycle can be constructed as follows. Let i be such that $r_i \neq r_{i+1}$, which is guaranteed to exist because $\boldsymbol{r}$ has two different symbols. Note that v_{i-1} and v_{i+1} are connected since $r_i \neq r_{i+1}$. Then, define a cycle $\boldsymbol{p}_2$ that contains all the vertices except v_i, namely,

$$\boldsymbol{p}_2 = (v_1 \to v_2 \to \cdots \to v_{i-1} \to v_{i+1} \to v_{i+2} \to \cdots \to v_M \to v_1),$$

The path $\boldsymbol{p}_2$ has length $M-1$ and cost M. Hence, there exist two paths $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ of length M and $M-1$, respectively, implying that $G(\boldsymbol{r})$ has period 1. Consider the $(M-1)$-fold repetition of cycle $\boldsymbol{p}_1$ and the M-fold repetition of cycle $\boldsymbol{p}_2$. Both have length $M(M-1)$, while the first has cost $M(M-1)$ and the second has cost M^2, implying cost-diversity. Finally, $G(\boldsymbol{r})$ has cost period M since it satisfies the M-periodic coboundary condition with $b=0$ and $B(v_i)=i$.

□

8.1.2 Synthesis Information Rates

In the sequel, we characterize the maximum amount of information that can be synthesized in t synthesis cycles using the synthesis sequence $\boldsymbol{s}$. To this end, we introduce a synthesis code, consisting precisely of those DNA sequences that can be synthesized in time t using $\boldsymbol{s}$. Therefore, the maximum number of information words that can be encoded to DNA strands is equal to the the number of DNA sequences that can be synthesized in time t. Recall the definition of the periodic subsequence graph presented in the previous section. By Proposition 8.3, a sequence $\boldsymbol{x}$ can be synthesized if and only if there is a path through the subsequence graph with the label $\boldsymbol{x}$. Using the notation $\mathcal{L}_{G,v}(t)$ and $N_{G,v}(t)$ from Definition 7.8 of limited cost followers and their number, we naturally arrive at the following definition.

Definition 8.6. *For an arbitrary $\boldsymbol{r} \in \Sigma_q^M$, we define $\mathcal{L}_{\boldsymbol{r}}(t) \triangleq \mathcal{L}_{G(\boldsymbol{r}),v_M}(t)$ and $N_{\boldsymbol{r}}(t) \triangleq |\mathcal{L}_{\boldsymbol{r}}(t)|$. Accordingly, we define $\mathcal{L}_{\boldsymbol{r}}(t,n) \triangleq \mathcal{L}_{\boldsymbol{r}}(t) \cap \Sigma_q^n$ and $N_{\boldsymbol{r}}(t,n) \triangleq |\mathcal{L}_{\boldsymbol{r}}(t,n)|$.*

With this definition, $\mathcal{L}_{\boldsymbol{r}}(t)$ is the set of sequences that can be synthesized in time t when using the periodic synthesis sequence $\boldsymbol{s} = (\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{r}, \dots)$ and $\log N_{\boldsymbol{r}}(t)$ is the number of information bits that can be used for encoding. Note that in principle it is also possible to define $N_{\boldsymbol{r}}(t)$ for

aperiodic sequences, however in such a case an asymptotic analysis is not possible using cost constrained systems as presented in this chapter. Given the above definitions, we are now in the position to present the main figures of merit for the synthesis process.

Let $\boldsymbol{s} = (\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{r}, \dots)$ be a semi-infinite sequence and let $0 \leq \alpha \leq 1$ be a parameter controlling the length of the synthesized sequences.

Definition 8.7. *Let $\boldsymbol{r} \in \Sigma_q^M$ be an arbitrary period. The **synthesis capacity**, measured by number of bits per synthesis cycle, is defined as*

$$C_{\boldsymbol{r}} = \limsup_{t \to \infty} \frac{\log(N_{\boldsymbol{r}}(t))}{t}.$$

*and similarly the **fixed-length synthesis capacity** is defined as*

$$C_{\boldsymbol{r}}(\alpha) = \limsup_{t \to \infty} \frac{\log(N_{\boldsymbol{r}}(t, \lfloor \alpha t \rfloor))}{t}.$$

8.2 Achievable Synthesis Information Rates

We proceed with our main theorems on the synthesis capacity for arbitrary periodic sequences. We start with the case of synthesizing fixed length sequences.

Theorem 8.8. *Let $\boldsymbol{r} \in \Sigma_q^M$ be an arbitrary sequence. Abbreviate $\alpha_G^{\mathsf{lo}} \triangleq \rho_{G(\boldsymbol{r})}(1)/\rho'_{G(\boldsymbol{r})}(1)$. For all α with $0 \leq \alpha \leq \alpha_G^{\mathsf{lo}}$, we have*

$$C_{\boldsymbol{r}}(\alpha) = \alpha \log \rho_{G(\boldsymbol{r})}(1).$$

For all α with $\alpha_G^{\mathsf{lo}} < \alpha < 1$,

$$C_{\boldsymbol{r}}(\alpha) = -\log x_0 + \alpha \log \rho_{G(\boldsymbol{r})}(x_0),$$

where x_0 is the unique real solution to $\alpha x \rho'_{G(\boldsymbol{r})}(x) = \rho_{G(\boldsymbol{r})}(x)$ in the interval $0 < x < 1$. For all $\alpha > 1$, $C_{\boldsymbol{r}}(\alpha) = 0$.

Proof. By Lemma 8.5, the graph $G(\boldsymbol{r})$ is deterministic, strongly connected and cost-diverse. This allows to use Theorem 7.12. Further, $\alpha_G^{\mathsf{up}} = 1$, since the minimum average cost of any cycle is precisely equal to 1. For a more detailed discussion on this connection, see, e.g., [KMR00]. □

The next result is on the variable-length synthesis capacity.

Theorem 8.9. *Let $\boldsymbol{r} \in \Sigma_q^M$ be an arbitrary sequence. Denote by x_0 the unique positive solution to $\rho_{G(\boldsymbol{r})}(x) = 1$. Then, the synthesis capacity is given by*

$$C_{\boldsymbol{r}} = -\log x_0.$$

Proof. The proof directly follows from the fact that $G(\boldsymbol{r})$ is deterministic and strongly connected by Lemma 8.5, which allows to invoke Theorem 7.15. □

8.2.1 Alternating Sequences

The alternating sequence is a prominent sequence due to the fact that it is known to maximize its number of distinct subsequences [HR00] and hence also the synthesis capacity. We proceed with deriving the synthesis capacity of alternating sequences over arbitrary alphabets.

Proposition 8.10. *Consider the q-ary alternating sequence with period $\boldsymbol{a}_q = (0, 1, \ldots, q-1)$. The synthesis capacity of this sequence is given by*

$$C_{\boldsymbol{a}_q} = -\log x_q,$$

where x_q is the unique positive solution to $\sum_{i=1}^{q} x^i = 1$. The fixed-length synthesis capacity is

$$C_{\boldsymbol{a}_q}(\alpha) = \begin{cases} \alpha \log q & \alpha < \frac{2}{q-1}, \\ -\log\left(\alpha \sum_{i=1}^{q} i x_q(\alpha)^{i+1}\right), & \frac{2}{q-1} < \alpha < 1 \end{cases},$$

where $x_q(\alpha)$ is the unique solution to $\sum_{i=1}^{q}(1-\alpha i)x^i = 0$, on the interval $0 < x < 1$.

For $2 \le q \le 4$ we obtain $C_{\boldsymbol{a}_2} \approx 0.694$, $C_{\boldsymbol{a}_3} \approx 0.879$, $C_{\boldsymbol{a}_4} \approx 0.947$, and $\lim_{q\to\infty} C_{\boldsymbol{a}_q} = 1$,

$$\begin{aligned} C_{\boldsymbol{a}_2}(\alpha) &= \alpha H_2\left(\frac{1-\alpha}{\alpha}\right), \\ C_{\boldsymbol{a}_3}(\alpha) &= \alpha H_2\left(\frac{\gamma}{\alpha}\right) + \gamma H_2\left(\frac{1-\alpha-\gamma}{\gamma}\right), \\ \lim_{q\to\infty} C_{\boldsymbol{a}_q}(\alpha) &= H_2(\alpha), \end{aligned}$$

where $\gamma = -\frac{2}{3}\alpha + \frac{1}{6}\sqrt{-8\alpha^2 + 12\alpha - 3} + \frac{1}{2}$ and $H_2(\bullet)$ is the binary entropy function.

We will make use of the following lemma. Denote by $\mathcal{T}(v) = \{\{\tau(e) : e \in \mathcal{E}, \mathsf{init}(e) = v\}\}$ the multiset of costs of all outgoing edges from $v \in \mathcal{V}$.

Lemma 8.11. *Let G be a strongly connected graph, where $\mathcal{T}(v)$ is invariant over all $v \in \mathcal{V}$. Then,*

$$F_{G,v}(x, y) = \frac{1}{(1-x)\left(1 - y\rho_G(x)\right)},$$

for all $v \in \mathcal{V}$, and, highlighting $\mathcal{T} \triangleq \mathcal{T}(v)$, the Peron root is equal to

$$\rho_G(x) = \sum_{\tau\in\mathcal{T}} x^\tau.$$

Proof. For simplicity denote by $\mathbf{1} = (1, \ldots, 1)$ the all-ones vector of length $|\mathcal{V}|$. since $\mathcal{T}(v)$ is the same for all $v \in \mathcal{V}$, it follows that

$$\boldsymbol{P}_G(x)\mathbf{1}^{\mathrm{T}} = \left(\sum_{\tau\in\mathcal{T}(v_1)} x^\tau, \ldots, \sum_{\tau\in\mathcal{T}(v_{|\mathcal{V}|})} x^\tau\right)^{\mathrm{T}} = \left(\sum_{\tau\in\mathcal{T}} x^\tau\right)\mathbf{1}^{\mathrm{T}}$$

and thus $\mathbf{1}^{\mathrm{T}}$ is a right Eigenvector of $\boldsymbol{P}_G(x)$ with Eigenvalue $\rho_G(x) = \sum_{\tau\in\mathcal{T}} x^\tau$. It follows that $(I - y\boldsymbol{P}_G(x))\mathbf{1}^{\mathrm{T}} = (1 - y\rho_G(x))\mathbf{1}^{\mathrm{T}}$ and therefore

$$\boldsymbol{F}_G(x, y) = \frac{1}{1-x} \cdot (I - y\boldsymbol{P}_G(x))^{-1}\mathbf{1}^{\mathrm{T}} = \frac{1}{(1-x)(1-y\rho_G(x))}\mathbf{1}^{\mathrm{T}}.$$

□

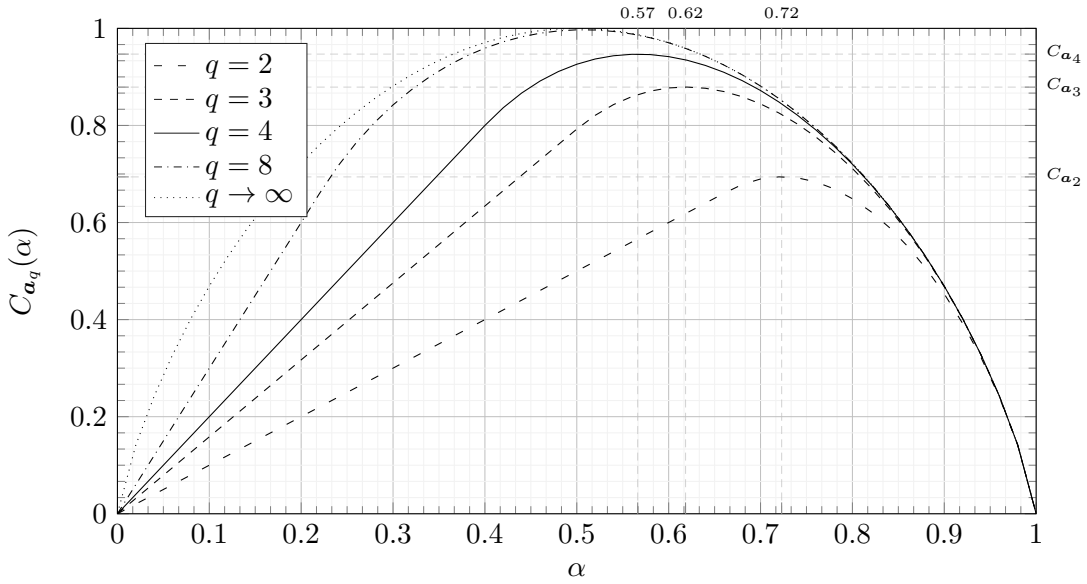


Figure 8.4: Synthesis capacity of the alternating sequences $\boldsymbol{a}_q$ over different alphabet sizes. The maxima are highlighted for $q \in \{2, 3, 4\}$ together with their maximizing α. Notice that these plots confirm the concavity of the fixed-length capacity in α and its maximum at the variable-length capacity, which is derived in Proposition A.10.

Proof of Proposition 8.10. For the special case of the alternating sequence, the synthesis graph $G_{\boldsymbol{a}_q}$ is a complete graph, where each vertex has q outgoing edges. The cost spectrum of the outgoing edges is $\mathcal{T}(v) = \{1, 2, \ldots, q\}$ for all vertices $v \in \mathcal{V}_{G_{\boldsymbol{a}_q}}$. Applying Lemma 8.11, yields

$$\rho_{G_{\boldsymbol{a}_q}}(x) = \sum_{i=1}^{q} x^i.$$

The results on the fixed length capacity then directly follow from applying Theorem 7.12. Similarly, the variable-length capacity follows from Theorem 7.15. The results for $q = 2$ and $q = 3$ follow from solving the determining equations followed by some algebraic reformulations. □

Figure 8.4 visualized the synthesis capacity for the alternating sequences over different alphabets. Notice that the relevant case for DNA synthesis is $q = 4$, and we include the remaining alphabet sizes for comparison. We see that using the proposed synthesis codes, we can improve the synthesis information rate from 0.5 bit per cycle[1] to roughly 0.947 bit per cycle.

8.2.2 Repeated Alternating Sequences

For illustrative purposes, we present the synthesis capacity for another class of synthesis sequences. We will discuss sequences with periods of the form $\boldsymbol{w}_{q,k} = (0^k, 1^k, 2^k, \ldots, (q-1)^k)$, where the k-th power represents the k-fold repetition of a letter. Hence, $\boldsymbol{w}_{q,1} = \boldsymbol{a}_q$ is the standard alternating sequence. It is possible to use the symmetry of the periodic synthesis graph, to find the spectral

[1] We state here the synthesis rate for synthesizing the *worst case* sequence $\boldsymbol{x} = (\mathsf{TTTT}\ldots)$. In *average*, the uncoded synthesis rate is 0.8 bit per cycle, see, e.g., [Len+20a].

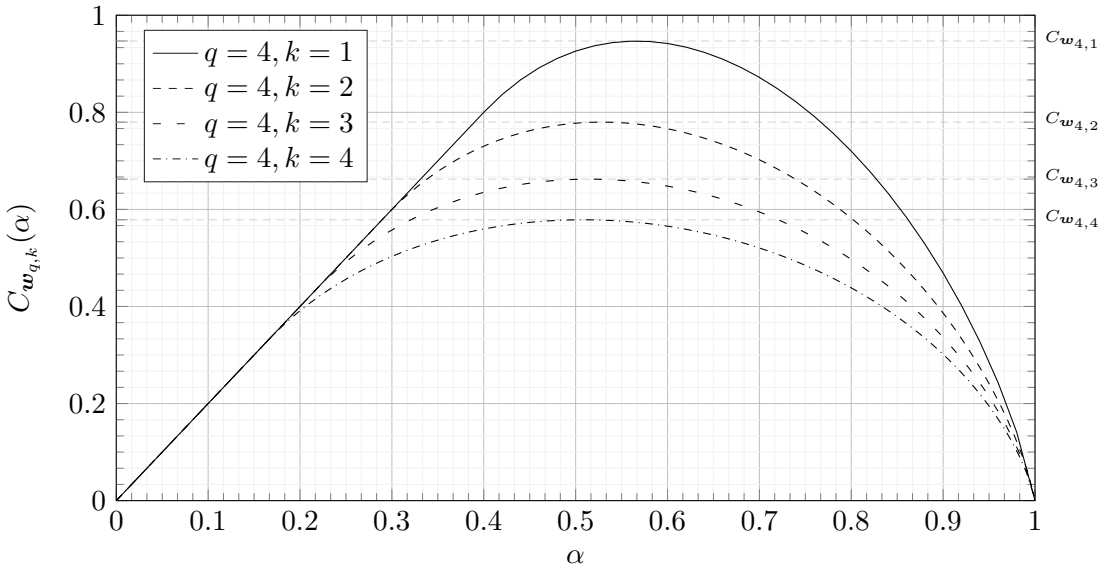


Figure 8.5: Synthesis capacity of the repeated alternating sequences $\boldsymbol{w}_{q,k}$ for $q = 4$ and different number of repetitions k.

radius as the Perron root of a simplified cost-enumerator matrix. In particular, when $k \geq 2$, the spectral radius is given by $\rho_{G_{\boldsymbol{w}_{q,k}}}(x) = \rho(\boldsymbol{W}_{q,k}(x))$, where $\boldsymbol{W}_{q,k}(x)$ is the $k \times k$ matrix

$$\boldsymbol{W}_{q,k}(x) = \begin{pmatrix} \sum_{i=1}^{q-1} x^{ki} & x & 0 & \dots & 0 & 0 \\ \sum_{i=1}^{q-1} x^{ki-1} & 0 & x & \dots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ \sum_{i=1}^{q-1} x^{ki-k+2} & 0 & 0 & \dots & 0 & x \\ \sum_{i=1}^{q} x^{ki-k+1} & 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

This follows from the symmetry of the synthesis sequence.[2] Associating the i-th vertex of the graph defined by $\boldsymbol{W}_{q,k}(x)$ with the i-th symbols in each repetition, we see that there is a one-to-one correspondence between the limited-cost sequences in the graph $G_{\boldsymbol{w}_{q,k}}$ and the graph defined by $\boldsymbol{W}_{q,k}(x)$. Figure 8.5 visualizes the synthesis capacity of the repeated alternating sequences for $q = 4$ and different number of repetitions k.

[2]The first symbols in each repetition have the same followers, up to a relabeling of the symbols, and the same is true for the second symbols, the third symbols, and so on.

8.3 Constrained Synthesis

In many applications it is required that the synthesized strands fulfill some given constraints. Such constraints can be, for example, the avoidance of runs of homopolymers[3] or a balanced GC content. We thus present a method to compute the maximum information rate under the constraint of synthesizing DNA sequences that have a specific structure. More formally, we let $\boldsymbol{s} = (\boldsymbol{r}, \boldsymbol{r}, \boldsymbol{r}, \dots)$ be a periodic synthesis sequence. Further, let the constraint that we wish to impose on the DNA strands be given in form of a directed, labeled graph $G_{\mathsf{c}} = (\mathcal{V}_{\mathsf{c}}, \mathcal{E}_{\mathsf{c}}, \sigma_{\mathsf{c}})$ with vertices $\mathcal{V}_{\mathsf{c}}$, edges $\mathcal{E}_{\mathsf{c}}$ and labels $\sigma_{\mathsf{c}} : \mathcal{E}_{\mathsf{c}} \mapsto \Sigma_q$. The set of possible constrained DNA sequences that are allowed to be synthesized is given by all words that are generated by paths through the graph $G_{\mathsf{c}} = (\mathcal{V}_{\mathsf{c}}, \mathcal{E}_{\mathsf{c}}, \sigma_{\mathsf{c}})$ that start from a dedicated starting state $v_{\mathsf{s}} \in \mathcal{V}_{\mathsf{c}}$. The following graph product will generate the set of constrained DNA sequences that can be synthesized in t synthesis cycles.

Definition 8.12. *Let $G_1 = (\mathcal{V}_1, \mathcal{E}_1, \sigma_1, \tau_1)$ be a directed, labeled, and weighted graph and $G_2 = (\mathcal{V}_2, \mathcal{E}_2, \sigma_2)$ be a directed, labeled graph. We define their label product as the directed, labeled, weighted graph $G = G_1 \times G_2$, $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ constituent of vertices $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2$, edges*

$$\mathcal{E} = \{(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2 : \sigma_1(e_1) = \sigma_2(e_2)\},$$

labeling $\sigma : \mathcal{E} \mapsto \sigma$ such that for $e = (e_1, e_2) \in \mathcal{E}$, it holds that $\sigma(e) = \sigma_1(e_1) = \sigma_2(e_2)$. The weights are $\tau : \mathcal{E} \mapsto \mathbb{N}$ with $\tau(e) = \tau_1(e_1)$.

With this definition, we can define the central quantity of interest for constrained synthesis, which is the maximal information rate, measured in bits per synthesis cycle, when synthesizing sequences that are constrained by the graph G_{c}.

Definition 8.13. *Let $\boldsymbol{r} \in \Sigma_q^M$ be an arbitrary period and $G_{\mathsf{c}} = (\mathcal{V}_{\mathsf{c}}, \mathcal{E}_{\mathsf{c}}, \sigma_{\mathsf{c}})$ a strongly connected, directed, labeled graph with starting vertex $v_{\mathsf{s}} \in \mathcal{V}_{\mathsf{c}}$. The **constrained synthesis capacity** measured by number of bits per synthesis cycle is defined as*

$$C_{\boldsymbol{r}, G_{\mathsf{c}}, v_{\mathsf{s}}} = \limsup_{t \to \infty} \frac{\log(N_{G(\boldsymbol{r}) \times G_{\mathsf{c}}, (v_M, v_{\mathsf{s}})}(t))}{t}.$$

*and similarly the **fixed-length constrained synthesis capacity** is defined as*

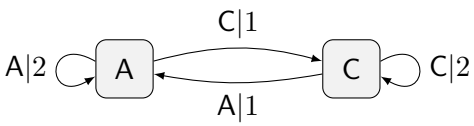
$$C_{\boldsymbol{r}, G_{\mathsf{c}}, v_{\mathsf{s}}}(\alpha) = \limsup_{t \to \infty} \frac{\log(N_{G(\boldsymbol{r}) \times G_{\mathsf{c}}, (v_M, v_{\mathsf{s}})}(t, \lfloor \alpha t \rfloor))}{t}.$$

It remains to prove that the label product of the synthesis graph G and the constrained graph G_{c} exactly generate the set of constrained sequences that can be synthesized in t cycles. This claim is proven in a more general form in the following lemma, which extends [LM95, Prop. 3.4.10] to costly graphs.

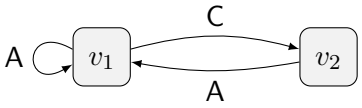
Lemma 8.14. *Let $G_1 = (\mathcal{V}_1, \mathcal{E}_1, \sigma_1, \tau_1)$ be a directed, labeled, weighted graph and $G_2 = (\mathcal{V}_2, \mathcal{E}_2, \sigma_2)$ be a directed, labeled graph. Then, for any $v_1 \in \mathcal{V}_1$ and $v_2 \in \mathcal{V}_2$*

$$\mathcal{L}_{G_1 \times G_2, (v_1, v_2)}(t) = \mathcal{L}_{G_1, v_1}(t) \cap \mathcal{L}_{G_2, v_2}.$$

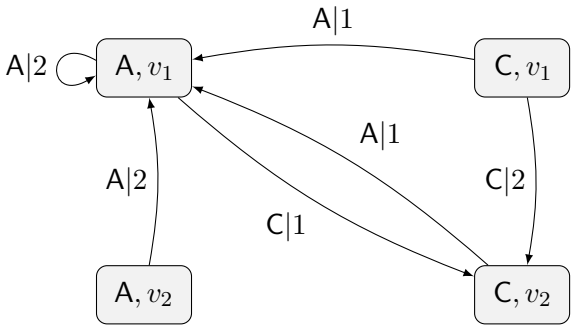
[3] A *run* of homopolymers is a consecutive string of nucleotides of the same type.



(a) Periodic subsequence graph $G(\mathsf{AC})$.



(b) Constraint graph G_c imposing the constraint that the letter C is always followed by the letter A.



(c) Label product graph of the graphs $G(\mathsf{AC})$ and G_c.

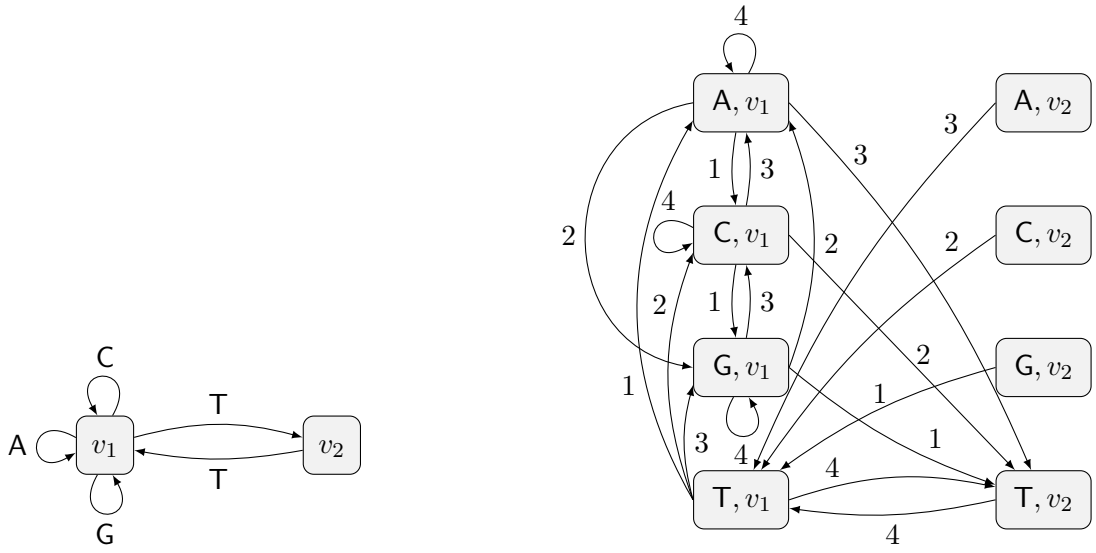
Figure 8.6: Periodic synthesis graph, constraint graph and their label product from Example 8.15.

Proof. We denote $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau) = G_1 \times G_2$ as the label product of G_1 and G_2. To start with, assume that a sequence $\boldsymbol{x}$ is contained in $\boldsymbol{x} \in \mathcal{L}_{G_1,v_1}(t) \cap \mathcal{L}_{G_2,v_2}$. This means that there exists a path $\boldsymbol{p}_1$ starting from v_1 through G_1 with $\sigma_1(\boldsymbol{p}_1) = \boldsymbol{x}$ and $\tau_1(\boldsymbol{p}_1) \leq t$ and there exists another path $\boldsymbol{p}_2$ starting from v_2 through G_2 with $\sigma(\boldsymbol{p}_2) = \boldsymbol{x}$. For every i, the i-th edge in $\boldsymbol{p}_1$ has the same label as the i-th edge of $\boldsymbol{p}_2$, such that $\boldsymbol{p} \triangleq (\boldsymbol{p}_1, \boldsymbol{p}_2)$ is a path through $G_1 \times G_2$ that starts from (v_1, v_2). Its label is $\sigma(\boldsymbol{p}) = \sigma_1(\boldsymbol{p}_1) = \sigma_2(\boldsymbol{p}_2) = \boldsymbol{x}$ and its cost is $\tau(\boldsymbol{p}) = \tau_1(\boldsymbol{p}_1) \leq t$. Henceforth, $\mathcal{L}_{G_1\times G_2,(v_1,v_2)}(t) \supseteq \mathcal{L}_{G_1,v_1}(t) \cap \mathcal{L}_{G_2,v_2}$. Let now conversely $\boldsymbol{x} \in \mathcal{L}_{G_1\times G_2,(v_1,v_2)}(t)$. Thus, there exists a path $\boldsymbol{p}$ through $G_1 \times G_2$, starting from (v_1, v_2) with $\sigma(\boldsymbol{p}) = \boldsymbol{x}$ and $\tau(\boldsymbol{p}) \leq t$. Writing $\boldsymbol{p} = (\boldsymbol{p}_1, \boldsymbol{p}_2)$, we see that $\boldsymbol{p}_1$ is a path through G_1 that starts at v_1 and has cost at most $\tau_1(\boldsymbol{p}_1) = \tau(\boldsymbol{p}) \leq t$ and label $\sigma_1(\boldsymbol{p}_1) = \sigma(\boldsymbol{p}) = \boldsymbol{x}$. Further, $\boldsymbol{p}_2$ is a path through G_2 that starts at v_2 and has label $\sigma_2(\boldsymbol{p}_2) = \sigma(\boldsymbol{p}) = \boldsymbol{x}$ and thus $\mathcal{L}_{G_1\times G_2,(v_1,v_2)}(t) \subseteq \mathcal{L}_{G_1,v_1}(t) \cap \mathcal{L}_{G_2,v_2}$, which proves the claim. □

With this result, the set of constrained sequences that can be synthesized in t cycles is precisely $\mathcal{L}_{G(\boldsymbol{r})\times G_\mathsf{c},(v_M,v_\mathsf{s})}(t)$. Note that the graph $G(\boldsymbol{r}) \times G_\mathsf{c}$ is deterministic, however it is not always strongly connected as we will show in the following example.

Example 8.15. *Consider the periodic synthesis sequence $\boldsymbol{s}$ with period $\boldsymbol{r} = (\mathsf{AC})$. Further, let G_c be the constrained graph of the language of words over $\Sigma_q = \{\mathsf{A}, \mathsf{C}\}$ comprising only words that do not contain runs of the letter C of length more than 1. Both graphs and their label product are displayed in Figure 8.6. The vertex (A, v_2) has no incident edges and thus the graph is not strongly connected. However, this vertex does not play a role, if we let either C be the starting vertex for the synthesis graph or v_1 be the starting vertex for the constraint graph. In these cases, the vertex can simply be discarded, resulting in a strongly connected graph.*

While our previous analysis has shown that for unconstrained synthesis, the alternating sequence with period $\boldsymbol{r} = (\mathsf{ACGT})$ maximizes the synthesis information rate, we show in the following example that such a claim is not necessarily true for constrained synthesis.



(a) Constraint graph G_{TT} of sequences, where the letter T only occurs in runs of even lengths.

(b) Label product graph of the graphs $G(\mathsf{ACGT})$ and G_{TT}. The edge labels are equal to the label of the final state and have been omitted for readability.

Figure 8.7: Constraint graph and their label product from Example 8.16.

Example 8.16. *Consider the periodic synthesis sequence* $\boldsymbol{s}$ *with period* $\boldsymbol{r} = (\mathsf{ACGT})$. *Further, let* G_{TT} *be the constraint graph that enumerates all words over* $\Sigma_q = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ *that only contain runs of the letter* T *that have length at least* 2. *Figure 8.7 shows the constraint graph and the product graph, while the synthesis graph is displayed in Figure 8.3 on Page 163. We find that* $C_{(\mathsf{ACGT}),G_{\mathsf{TT}}} \approx 0.7188$, *however if we use a synthesis sequence with period* $\boldsymbol{r} = (\mathsf{ACG})$ *instead, we can get a higher constrained synthesis rate of* $C_{(\mathsf{ACG}),G_{\mathsf{TT}}} \approx 0.8791$. *This result is intuitive, as compared to the unconstrained case, the constraint increases the cost of synthesizing* T *by* 4 *cycles (following the constraint, we always need to synthesize two* T*'s in succession), which means that it is more efficient to remove* T *from the synthesis sequence in order to reduce the cost of the remaining symbols. Notice that it is not a coincidence that* $C_{(\mathsf{ACG}),G_{\mathsf{TT}}} = C_{(\mathsf{ACG})} = C_{\boldsymbol{a}_3}$ *(c.f. Proposition 8.10), as the constraint* G_{TT} *contains all sequences over the alphabet* $\Sigma_3 = \{\mathsf{A}, \mathsf{C}, \mathsf{G}\}$.

Note that strictly speaking, in the previous example, the sequence (ACG) is again an alternating sequence over a smaller alphabet. It is indeed also possible to construct constraints for which no alternating sequence is optimal, however these examples can become quite extensive, especially over the DNA alphabet and are beyond the scope of this dissertation.

8.4 Counting Subsequences Using Costly Constrained Channels

Counting subsequences of a given supersequence is a problem that arises in manifold research areas, such as bioinformatics, information theory, and coding theory. While explicit formulas for the number of subsequences of arbitrary supersequences exist [MKB08], these expressions are in general difficult to analyze and compact and explicit expressions for arbitrary sequences remain unknown to date, except for some special sequences, such as alternating sequences [HR00]. In this

work, we provide a compact and precise characterization of both the total number of subsequences and the number of fixed-length subsequences of a given *arbitrary* supersequence.

We shortly comment on how it is possible to use our results to efficiently and precisely compute the number of subsequences of an arbitrary sequence. The central observation is that, by definition of the synthesis problem, the sequences that can be synthesized in time t using $\boldsymbol{s}$ are precisely the subsequences of $(s_1, \ldots, s_t)$. Thus, by Proposition 8.3, we can count the number of subsequences of $\boldsymbol{s}$ by counting the number of cost-t followers in the subsequence graph $G_{\mathsf{sub}}(\boldsymbol{s})$, i.e., $N_{G_{\mathsf{sub}}(\boldsymbol{s}),v_0}(t)$. Analogously, we can count the length-n subsequences of $\boldsymbol{s}$ by the length-n followers of cost at most t, i.e., $N_{G_{\mathsf{sub}}(\boldsymbol{s}),v_0}(t,n)$. Therefore, using the recursion in the proof of Lemma 7.41, it is possible to count the number of subsequences with dynamic programming. Even more, if $\boldsymbol{s}$ is periodic, then Theorems 8.8 and 8.9 can be used to compute the exponential growth rate of the number of variable-length or fixed-length subsequences. In fact, due to Lemma 8.5 we can even compute the precise asymptotic behavior of the number of subsequences using Theorems 7.12 and 7.15.

Finally, remarkably, our analysis extends to the case of counting subsequences that fulfill a certain constraint that is represented in the form of a labeled and directed graph. This is evident from the discussion in Section 8.3.

8.5 Conclusion

The object of study within this chaper was a popular synthesis process, known as array-based synthesis. We have shown that by restricting the set of DNA sequences that we allow to synthesize, it is possible to optimize the number of information bits that can be synthesized per synthesis cycle, improving the cost-efficiency of DNA-based data storage systems. We have shown how to relate the synthesis problem to costly constrained channels, allowing to compute the maximum achievable information rate using the results presented in Chapter 7. We further have shown how to compute maximal achievable synthesis information rates when one restricts to synthesizing constraint sequences. Our results have proven that there are constraints for which the optimal synthesis sequence for the unconstrained case, i.e., the alternating sequence, is not optimal. Finally, our results give the solution to a prominent problem in information theory, coding theory and bioinformatics, i.e., that of efficiently enumerating and counting all subsequences of a given supersequence. We have shown that with our methods, we can characterize the exact asymptotic behavior of the number of subsequences by a comprehensible analysis of the Perron root of the cost-enumerator matrix.

Chapter 9

Further Publications by the Author

We supplement this manuscript with a short presentation of further publications from the author that resulted from his work as a doctoral candidate.

9.1 Concatenated Codes for the Probabilistic DNA Storage Channel

Motivated by the employment in multiple current DNA storage experiments, we investigate the performance of concatenated codes over the probabilistic DNA storage channel. Concatenated codes are the natural choice for DNA storage experiments, since typically the data is stored on many parallel sequences, motivating the employment of an inner code for each sequences, enhanced by an outer code over all strands. In [LWP20] we show that the achievable rate of a decoder that decodes each strand with a hard decision using the inner code, followed by a decoding of the outer code, exhibits a gap with respect to the capacity of the probabilistic DNA storage channel. This is because the inner code has a fixed rate, while the nature of the inner channel is unknown due to the randomness of the sequencing coverage of each DNA strand. We show that combining several strands to joint inner codewords, it is possible to narrow the gap between achievable rates and capacity, under the cost of additional decoding complexity.

The author's main contribution to this study was the derivation and proof of the achievable information rates of the investigated concatenated coding schemes.

9.2 Covering Codes for Insertions and Deletions

A covering code is a set of codewords with the property that the union of balls, suitably defined, around these codewords covers an entire space. Covering codes are a core object of study in coding theory and discrete mathematics. They have found applications in diverse areas such as data compression, football pools, circuit complexity, lattice problems, and approximate nearest neighbor search. Generally, the goal is to find the covering code with the minimum size codebook. While most prior work on covering codes has focused on the Hamming metric, in [Len+20b; Len+21e] we consider the problem of designing covering codes defined in terms of either insertions or deletions. First, we provide new sphere-covering lower bounds on the minimum possible size of such codes. Then, we provide new existential upper bounds on the size of optimal covering codes for a single insertion or a single deletion that are tight up to a constant factor. Finally, we derive improved upper bounds for covering codes using $R \geq 2$ insertions or deletions. We prove

that codes exist with density that is only a factor $O(R \log R)$ larger than the lower bounds for all fixed R. In particular, our upper bounds have an optimal dependence on the word length, and we achieve an asymptotic density that matches the best known bounds for Hamming distance covering codes.

9.3 Codes for Reconstruction of Multiple Reads of a DNA Sequence

Decoding sequences that stem from multiple transmissions of a codeword over an insertion, deletion, and substitution channel is a critical component of efficient DNA data storage systems. In [Len+21c], we consider a concatenated coding scheme with an outer low-density parity-check code and either an inner convolutional code or a block code. We propose two new decoding algorithms for inference from multiple received sequences, both combining the inner code and channel to a joint hidden Markov model to infer symbolwise a posteriori probabilities (APPs). The first decoder computes the exact APPs by jointly decoding the received sequences, whereas the second decoder approximates the APPs by combining the results of separately decoded received sequences. Using the proposed algorithms, we evaluate the performance of decoding multiple received sequences by means of achievable information rates and Monte-Carlo simulations. We show that decoding multiple sequences at once can lead to significant performance gains as compared to a single received sequence.

In this joint project, the author's main contributions comprised the development of the joint decoder and the numerical evaluations of achievable information rates and error rates of the schemes with inner convolutional codes.

9.4 Function-Correcting Codes

In standard communication systems, a sender desires to convey a digital message to a receiver via an erroneous channel. To protect this message from errors, it is first encoded using an error-correcting code and then it is transmitted over the channel to the receiver, which decodes the received word to obtain the original message. Within this setup, the goal is to recover the message correctly. Motivated by applications in machine learning and archival data storage, in [Len+21a; Len+21b] we introduce function-correcting codes, a new class of codes designed to protect a function evaluation of the data against errors. We show that function-correcting codes are equivalent to irregular-distance codes, i.e., codes that obey some given distance requirement between each pair of codewords. Using these connections, we study irregular-distance codes and derive general upper and lower bounds on their optimal redundancy. Since these bounds heavily depend on the specific function, we provide simplified, suboptimal bounds that are easier to evaluate. We further employ our general results to specific functions of interest and compare our results to standard error-correcting codes which protect the whole data.

9.5 Codes Correcting a Burst of Deletions

Burst deletions and insertions are a class of errors that can be found in a variety of applications, ranging from modern data storage systems, e.g., DNA-based data storage over communication systems to file synchronization. In contrast to classical deletion and insertions errors, that delete and insert symbols into a string at arbitrary positions, burst errors occur at consecutive

positions. In [LP20], we present an efficiently encodable and decodable code construction that is capable of correcting a burst of deletions of length at most k. The redundancy of this code is $\log n + O_k \log\log n$, which is optimal in terms of scaling with n. The code can be split into two main components. First, we impose a constraint that allows us to locate the burst of deletions up to an interval of size roughly $\log n$. Then, with the knowledge of the approximate location of the burst, we use several shifted Varshamov-Tenengolts codes to correct the burst of deletions, which only requires a small amount of redundancy since the location is already known up to an interval of small size. Finally, we show how to efficiently encode and decode the code.

9.6 Multi-Symbol Duplication-Correcting Codes

During replication of DNA it is common that multiple consecutive nucleotides are repeated and duplicated within a DNA strand. These errors are known as tandem duplications, where a sequence of symbols is repeated; respectively as palindromic duplications, where a sequence is repeated in reversed order. In our works, [LWY17; LWY19] we investigate error-correcting codes over channels that suffer from single tandem or palindromic duplication errors. In particular, we derive upper bounds on the cardinality of tandem duplication and palindromic deletion-correcting codes by deriving sphere packing bounds for these error types. Our upper bounds on the cardinality directly imply lower bounds on the redundancy which we compare with the redundancy of the best known construction correcting arbitrary burst errors. Our results indicate that the correction of palindromic deletions requires more redundancy than the correction of tandem duplications. Further, there is a significant gap between the minimum redundancy of duplication correcting codes and burst insertion-correcting codes. In [LJW18], we generalize our results on single tandem duplication errors to the case of multiple errors. We propose explicit constructions that correct duplications of multiple consecutive symbols. Finally, we discuss the asymptotic behavior of the derived codes and bounds, exposing fundamental insights about the tandem duplication channel.

9.7 Clustering-Correcting Codes

Clustering DNA sequences according to their similarity is a vital aspect of many DNA-based storage systems. Accurate clustering results however depend on the dissimilarity of the original strands, which usually depends on the user data to be stored and is not necessarily the case for all possible messages. In [Shi+19; Shi+22], we propose a new class of codes, called *Clustering-Correcting Codes* that encode user data into DNA strands that are guaranteed to be dissimilar. We further design the codes such that a clustering algorithm based on the (possibly erroneous) indices of the sequences, followed by a simple decision rule within the clusters can be used to achieve accurate clustering, even in the presence of errors. We present efficient encoding and decoding algorithms for these codes and derive converse bounds on the minimum redundancy required to impose the desired constraint on the DNA sequences.

The main contributors to this study were Tal Shinkar and Eitan Yaakobi, while the author assisted with the derivation of the converse and achievability bounds.

9.8 Error Correction for Physically Unclonable Functions

Physically unclonable functions are hardware devices that serve as unique identifiers or key storage devices. Our publications [Imm+17; Imm+19] propose a novel variable-length mapping scheme from physical readouts to binary messages. While such a mapping comes at the advantage of a reduced bias, it introduces new errors in form of insertions and deletions, when the reads are perturbed by errors. To combat these errors, we use error correction schemes based on Varshamov-Tenengolts codes and prove the feasibility of the scheme using simulations with an empirical read distribution. This scheme promises a high effective number of secrecy bits, while providing sensitivity with respect to tampering attacks.

Within this project, the author designed the codes for insertion and deletion correction.

Chapter 10

Concluding Remarks

The topic of reliability and efficiency in digital memories is a popular candidate for information-theoretic and coding-theoretic studies. As part of this research branch, this dissertation is focused around the abstraction, modeling and analysis of the core processes involved in next-generation memories, such as DNA-based storage systems. Large parts of the research are concerned with the reliable communication of information over novel channels that abstract central aspects of modern memories, such as the appearance of synchronization errors in the form of insertions or deletions or the disorder of data stored in different parts of the memory. Another part deals with the optimization of array-based synthesis, establishing novel connections between the theory of analytical combinatorics in several variables and noiseless information theory.

Being a research object for only a short period of time, "*DNA-based storage systems are new and uncharted territory for coding theorists*" [Mil+18] and the list of open problems within these fascinating areas is long, demanding the ceaseless interest of researchers from diverse fields. Here, for brevity, we only focus on those open questions that are most related to our studies.

Regarding zero-error codes over unordered sets, we remark that there are several interesting problems that have not been addressed in the literature yet. One is to find the comprehensive codes that perform well in all or at least in many parameter regimes. Especially for the combination of a loss of sequences together with errors within the sequences, a general approach to and understanding of constructive solutions remains elusive to date. From an algorithmic point of view, efficient encoders and decoders are also of interest due to very large code dimensions that arise from the two-dimensional nature of the codewords.

Within the field of information transmission via a probabilistic noisy drawing of input sequences, the derivation of the capacity for the case of insertion and deletion errors is an intriguing question. The methods presented in this thesis suggest that our results on the probabilistic DNA storage channel may generalize to a broader class of constituent channels. That is, it is conceivable that a replacement of the q-ary symmetric channel with an insertion and deletion channel yields a channel whose capacity is obtained by replacing the capacity of the multinomial channel with the capacity of a multiple draw insertion and deletion channel in Theorem 6.2. This result remains unproven to date and is hindered due to the lack of a comprehensive understanding of even the conventional insertion and deletion channel with a single received sequence. However, it might still be possible to derive the capacity without the precise knowledge of the individual capacity expressions, provided that specific properties of the insertion and deletion channels are known, such as the capacity-achieving input distributions.

Appendix A

Auxiliary and Supplementary Results

A.1 Auxiliary Lemmas

For the readers convenience, we summarize auxiliary lemmas that are used throughout the dissertation.

Lemma A.1. *Let $f(n), g(n) : \mathbb{N} \mapsto \mathbb{R}$ be two arbitrary functions with $f(n) = o(1)$ for $n \to \infty$. Then,*

$$g(n) \log\left(1 + f(n)\right) = g(n) f(n) \log \mathrm{e} + O\left(g(n) f^2(n)\right).$$

Proof. We use the standard bound on the natural logarithm

$$\frac{x}{x+1} \leq \frac{\log(1+x)}{\log \mathrm{e}} \leq x,$$

for all $x > -1$. Since $f(n) = o(1)$, there exists $n_0 \in \mathbb{N}$, such that $|f(n)| < 1$ for all $n \geq n_0$ and therefore

$$g(n) \frac{f(n)}{f(n)+1} \leq g(n) \frac{\log\left(1 + f(n)\right)}{\log \mathrm{e}} \leq g(n) f(n),$$

for all $n \geq n_0$. This allows to find an upper bound to the following limit of the first order approximation

$$\lim_{n \to \infty} \left| \frac{g(n) \log\left(1 + f(n)\right) - g(n) f(n) \log \mathrm{e}}{g(n) f^2(n)} \right| \leq 1,$$

by plugging in the lower and upper bound on $g(n) \log(1 + f(n))$, which proves the statement. □

Lemma A.2. *Fix $k \in \mathbb{N}_0$ and let $f(n) : \mathbb{N} \mapsto \mathbb{R}$ be a function with $f(n) = o(n)$ as $n \to \infty$. When $n \to \infty$, the binomial coefficient satisfies*

$$\log \binom{n - f(n)}{k} = k \log n - \log k! + o(1), \tag{A.1}$$

$$\log \sum_{i=0}^{k} \binom{n}{i} = k \log n - \log k! + o(1). \tag{A.2}$$

Proof. We start by proving statement (A.1). From the definition of the binomial coefficient, for all $n \geq f(n) + k$ it directly follows that

$$\frac{(n - f(n) - k)^k}{k!} \leq \binom{n - f(n)}{k} \leq \frac{(n - f(n))^k}{k!}.$$

Thus, we can bound the binomial coefficient from below by

$$\begin{aligned}\log \frac{(n - f(n) - k)^k}{k!} &= k \log(n - f(n) - k) - \log k! = k \log n + k \log(1 - (f(n) + k)/n) - \log k! \\ &\overset{(a)}{=} k \log n - \log k! - o(1),\end{aligned}$$

where in equality (a) we used Lemma A.1 to show that $k \log(1 - (f(n) + k)/n) = o(1)$ for fixed k, m as $n \to \infty$. The upper bound exhibits the same asymptotic behavior, which can be shown using analogous steps. Therefore, the statement (A.1) follows.

We now proceed with proving statement (A.2). On the one hand, we can bound the binomial sum from below by $\sum_{i=0}^{k} \binom{n}{i} \geq \binom{n}{k}$ and thus the binomial sum is also asymptotically bounded from below by the right hand side of statement (A.1). On the other hand, we can prove that

$$\begin{aligned}\sum_{i=0}^{k} \binom{n}{i} &= \binom{n}{k} \left(1 + \sum_{i=0}^{k-1} \frac{\binom{n}{i}}{\binom{n}{k}}\right) = \binom{n}{k} \left(1 + \sum_{i=0}^{k-1} \frac{(n-k)!k!}{(n-i)!i!}\right) \\ &\overset{(a)}{\leq} \binom{n}{k} \left(1 + k \frac{k!}{(n-k+1)}\right),\end{aligned}$$

where in inequality (a) we used that $(n-i)!i! \geq (n-k+1)!$ for all $i \leq k-1$.[1] Now, the second factor approaches 1 as $n \to \infty$, since k is fixed. Therefore, taking logarithms, this upper bound has the same asymptotic behavior as (A.1) and the statement follows. □

Lemma A.3. *Let $f(n), g(n) : \mathbb{N} \mapsto \mathbb{R}$ be two arbitrary functions with $g(n) = o(f(n))$ and $g(n) = \omega(1)$, when $n \to \infty$. The binomial coefficient satisfies*

$$\log \binom{f(n)}{g(n)} = g(n) \log \frac{\mathrm{e} f(n)}{g(n)} + o(g(n)),$$

when $n \to \infty$.

Proof. Note that $g(n) = o(f(n))$ and $g(n) = \omega(1)$ automatically implies $f(n) = \omega(1)$. The binomial coefficient satisfies

$$\begin{aligned}\log \binom{f(n)}{g(n)} &= \log \frac{f(n)!}{(f(n) - g(n))! g(n)!} \\ &= g(n) \log \frac{f(n)}{g(n)} - \frac{1}{2} \log g(n) - \left(f(n) - g(n) + \frac{1}{2}\right) \log \left(1 - \frac{g(n)}{f(n)}\right) + \gamma,\end{aligned}$$

[1] While this bound is rather crude, it suffices for the statement of the lemma.

where $\gamma = -\log\sqrt{2\pi} + O(\frac{1}{g(n)})$. Here we used a refinement [Rob55] of Stirling's approximation, which states that

$$\sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n \mathrm{e}^{\frac{1}{12n+1}} \leq n! \leq \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n \mathrm{e}^{\frac{1}{12n}},$$

for any $n \in \mathbb{N}$. Using Lemma A.1, we obtain

$$\begin{aligned} -\left(f(n) - g(n) + \frac{1}{2}\right)\log\left(1 - \frac{g(n)}{f(n)}\right) &= \log\mathrm{e}\left(g(n) - \frac{g^2(n)}{f(n)} + \frac{g(n)}{2f(n)}\right) + O\left(\frac{g^2(n)}{f(n)}\right) \\ &= g(n)\log\mathrm{e} + O\left(\frac{g^2(n)}{f(n)}\right), \end{aligned}$$

where we used that $\frac{g(n)}{f(n)} = o(1)$. Plugging this result into the expression of the binomial coefficient and using further $\log g(n) = o(g(n))$ and $\gamma = o(g(n))$ proves the lemma. □

Lemma A.4. *Let $n \in \mathbb{N}$, $0 < p < 1$ and $k \in \mathbb{N}$, $k \geq np$. Then, the binomial tail distribution can be bounded from above by*

$$\begin{aligned} \sum_{i=k}^{n}\binom{n}{i}p^i(1-p)^{n-i} &\leq 2^{-nD\left(\frac{k}{n}||p\right)}, \\ \sum_{i=k}^{n}\binom{n}{i}p^i(1-p)^{n-i} &\leq \mathrm{e}^{-2n\left(\frac{k}{n}-p\right)^2} \end{aligned}$$

where $D(p_1||p_2) = p_1\log(p_1/p_2) + (1-p_1)\log((1-p_1)/(1-p_2))$ is the Kullback-Leibler divergence between two Bernoulli distributions with probabilities p_1 and p_2.

Proof. The first inequality is a well-known upper bound on the binomial tail, which can be found in, e.g., [Ash90, Lemma 4.7.2]. The second inequality can directly be proven using Hoeffding inequality [Ver18, Thm. 2.2.6], by using that the expected value of the binomial distribution is equal to np. □

Lemma A.5. *For any events $\mathcal{A}, \mathcal{B}$, the conditional probability of $\mathcal{A}$ given $\mathcal{B}$ satisfies*

$$\mathsf{Pr}\left(\mathcal{A}\right) + \mathsf{Pr}\left(\mathcal{B}\right) - 1 \leq \mathsf{Pr}\left(\mathcal{A}|\mathcal{B}\right) \leq \frac{\mathsf{Pr}\left(\mathcal{A}\right)}{\mathsf{Pr}\left(\mathcal{B}\right)}.$$

Proof. The proof follows directly from basic stochastic principles. On the one hand, we have

$$\mathsf{Pr}\left(\mathcal{A}|\mathcal{B}\right) = \frac{\mathsf{Pr}\left(\mathcal{A}\cap\mathcal{B}\right)}{\mathsf{Pr}\left(\mathcal{B}\right)} \leq \frac{\mathsf{Pr}\left(\mathcal{A}\right)}{\mathsf{Pr}\left(\mathcal{B}\right)}.$$

For the lower bound, we denote by $\mathcal{B}^{\mathsf{c}}$ the *complement* event of $\mathcal{B}$ with $\mathsf{Pr}\left(\mathcal{B}^{\mathsf{c}}\right) = 1 - \mathsf{Pr}\left(\mathcal{B}\right)$. The lower bound then follows from the following series of inequalities

$$\mathsf{Pr}\left(\mathcal{A}|\mathcal{B}\right) = \frac{\mathsf{Pr}\left(\mathcal{A}\cap\mathcal{B}\right)}{\mathsf{Pr}\left(\mathcal{B}\right)} \geq \mathsf{Pr}\left(\mathcal{A}\cap\mathcal{B}\right) = \mathsf{Pr}\left(\mathcal{A}\right) - \mathsf{Pr}\left(\mathcal{A}\cap\mathcal{B}^{\mathsf{c}}\right) \geq \mathsf{Pr}\left(\mathcal{A}\right) - \mathsf{Pr}\left(\mathcal{B}^{\mathsf{c}}\right).$$

□

A.2 Bound on the Fraction of Clustered Sets

Lemma A.6. *For any fixed $0 < \beta < 1$, fixed integer $\delta \in \mathbb{N}_0$, and any integer functions $y(M) \leq M$ and $z(L)$ with $z(L) \leq 2^L/y(M)$ for large enough M, the following asymptotic property holds*

$$\log \frac{\binom{2^L}{M-y(M)}\binom{2^L/z(L)}{y(M)}}{\binom{2^L}{M-\delta}} \leq -y(M) \log \frac{z(L)y(M)}{\mathrm{e}M} + O\left(\frac{My(M)}{2^L}\right) + O(L),$$

when $M \to \infty$ and $M = 2^{\beta L}$.

Proof. The lemma can be shown directly by inserting the factorial expression for the binomial coefficient. Denoting by $n^{\underline{m}} = n \cdot (n-1) \dots (n-m+1)$ the falling factorial for arbitrary $n, m \in \mathbb{N}_0$ with $n \geq m$, we obtain

$$\begin{aligned}\log \frac{\binom{2^L}{M-y(M)}\binom{2^L/z(L)}{y(M)}}{\binom{2^L}{M-\delta}} &= \log \frac{(2^L/z(L))^{\underline{y(M)}}(2^L - M + \delta)^{\underline{\delta}}}{(2^L - M + y(M))^{\underline{y(M)}} M^{\underline{\delta}}} + \log \binom{M}{y(M)} \\ &\leq y(M) \log \frac{2^L/z(L)}{2^L - M} + \log \binom{M}{y(M)} + O(L).\end{aligned}$$

Using $\binom{n}{k} \leq \left(\frac{\mathrm{e}n}{k}\right)^k$ as an upper bound for the binomial coefficient and rearranging the term $z(L)$, we further obtain

$$\begin{aligned}\log \frac{\binom{2^L}{M-y(M)}\binom{2^L/z(L)}{y(M)}}{\binom{2^L}{M-\delta}} &\leq -y(M) \log\left(1 - M/2^L\right) + y(M) \log \frac{\mathrm{e}M}{z(L)y(M)} + O(L) \\ &\overset{(a)}{\leq} y(M) \log \frac{\mathrm{e}M}{z(L)y(M)} + O\left(\frac{My(M)}{2^L}\right) + O(L).\end{aligned}$$

In inequality (a), we used Lemma A.1 for the approximation of the logarithm. □

A.3 Capacity of the Ordered Multinomial Channel

Lemma A.7. *Fix $0 < p < 1$, $q \in \mathbb{N}$ and let the distribution $\mathsf{Pr}(\boldsymbol{d})$ be a given distribution that has a bounded number of draws and converges in frequency to $\boldsymbol{\nu}$. Then, the capacity of the ordered parallel multinomial channel is given by*

$$C_{\mathsf{OPM}}(\boldsymbol{\nu}, p, q) = \sum_{d \geq 0} \nu_d C_{\mathsf{Mul}}(d, p, q).$$

Proof. For ease of notation, we abbreviate $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_M)$ as the length-ML vector comprising all input sequences and $\boldsymbol{Z} = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_M)$ as the length-ML vector that contains all output clusters. We first bound the capacity from above using Fano's inequality, which implies that for any code $\mathcal{C} \subseteq \Sigma_q^{M \times L}$ of rate R, we have [CT06]

$$R \leq \mathsf{Pr}(\mathsf{Err}|\mathcal{C})\, R + \frac{1 + I(\boldsymbol{X}; \boldsymbol{Z})}{ML}.$$

By the definition of an achievable code rate we have that $\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}\right) \to 0$ as $M \to \infty$. Further, $\frac{1}{ML} \to 0$ and we therefore obtain a valid bound on achievable code rates, if we bound the mutual information $I(\boldsymbol{X};\boldsymbol{Z}) = H(\boldsymbol{Z}) - H(\boldsymbol{Z}|\boldsymbol{X})$ from above. Using the chain rule of entropy [CT06, Thm. 2.5.1], we can write the output entropy as

$$H(\boldsymbol{Z}) = H(\boldsymbol{Z},\boldsymbol{d}) - H(\boldsymbol{d}|\boldsymbol{Z}) = H(\boldsymbol{Z}|\boldsymbol{d}) + H(\boldsymbol{d}) - H(\boldsymbol{d}|\boldsymbol{Z}).$$

Similarly, we can express the conditional output entropy as

$$H(\boldsymbol{Z}|\boldsymbol{X}) = H(\boldsymbol{Z},\boldsymbol{d}|\boldsymbol{X}) - H(\boldsymbol{d}|\boldsymbol{Z},\boldsymbol{X}) = H(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{d}) + H(\boldsymbol{d}|\boldsymbol{X}) - H(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{Z}).$$

We can simplify both expressions due to the fact that $\boldsymbol{d}$ is a function of $\boldsymbol{Z}$, since for each $i \in [M]$ it is possible to infer the number of draws d_i from $\boldsymbol{z}_i$, by identifying the symbol alphabet of $\boldsymbol{z}_i$. This is in particular due to the fact that for the multinomial channel, it holds that the output alphabets $\mathcal{Z}_d$ are distinct. It follows that $H(\boldsymbol{d}|\boldsymbol{Z}) = H(\boldsymbol{d}|\boldsymbol{X},\boldsymbol{Z}) = 0$. Further, since $\boldsymbol{X}$ and $\boldsymbol{d}$ are independent by the definition of the channel, we have $H(\boldsymbol{d}|\boldsymbol{X}) = H(\boldsymbol{d})$ and we obtain

$$I(\boldsymbol{X};\boldsymbol{Z}) = H(\boldsymbol{Z}|\boldsymbol{d}) - H(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{d}) = \sum_{\mathbf{d}} \mathsf{Pr}\left(\mathbf{d}\right)\left(H(\boldsymbol{Z}|\boldsymbol{d}=\mathbf{d}) - H(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{d}=\mathbf{d})\right).$$

By the chain rule of entropy together with the fact that conditioning reduces entropy [CT06, Thm. 2.6.5], the joint entropy can be bounded from above by the sum of the individual marginal entropies and we obtain

$$H(\boldsymbol{Z}|\boldsymbol{d}=\mathbf{d}) \leq \sum_{i=1}^{M} H(\boldsymbol{z}_i|\boldsymbol{d}=\mathbf{d}) = \sum_{i=1}^{M} H(\boldsymbol{z}_i|d_i=\mathtt{d}_i),$$

where the last equality is due to the fact that given d_i, $\boldsymbol{z}_i$ is independent of all d_j with $j \neq i$. On the other hand, the conditional output entropy can be simplified to

$$H(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{d}=\mathbf{d}) = \sum_{i=1}^{M} H(\boldsymbol{z}_i|\boldsymbol{X},\boldsymbol{d}=\mathbf{d}) = \sum_{i=1}^{M} H(\boldsymbol{z}_i|\boldsymbol{x}_i,d_i=\mathtt{d}_i).$$

The first equality is because we can write $\boldsymbol{z}_i = \boldsymbol{x}_i + \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is a random vector over the symbols $\Sigma_q^{\mathtt{d}_i}$ that is independent of $\boldsymbol{X}$. Thus, given $\boldsymbol{X}$ and $\boldsymbol{d}$, the only randomness in $\boldsymbol{z}_i$ is $\boldsymbol{e}_i$ and henceforth, the $\boldsymbol{z}_i$'s are independent given $\boldsymbol{X}$ and $\boldsymbol{d}$. Consequently the joint entropy is equal to the sum of marginal entropies. The second equality follows from the fact that given $\boldsymbol{x}_i$ and d_i, $\boldsymbol{z}_i$ is independent of $\boldsymbol{x}_j$ and d_j for all $j \neq i$. It follows that

$$\begin{aligned} I(\boldsymbol{X};\boldsymbol{Z}) &\leq \sum_{\mathbf{d}} \mathsf{Pr}\left(\mathbf{d}\right) \sum_{i=1}^{M} (H(\boldsymbol{z}_i|d_i=\mathtt{d}_i) - H(\boldsymbol{z}_i|\boldsymbol{x}_i,d_i=\mathtt{d}_i)) = \sum_{\mathbf{d}} \mathsf{Pr}\left(\mathbf{d}\right) \sum_{i=1}^{M} I(\boldsymbol{x}_i,\boldsymbol{z}_i|d_i=\mathtt{d}_i) \\ &\overset{(a)}{\leq} L \sum_{\mathbf{d}} \mathsf{Pr}\left(\mathbf{d}\right) \sum_{i=1}^{M} C_{\mathsf{Mul}}(\mathtt{d}_i,p,q) \overset{(b)}{=} L \sum_{\mathbf{n}} \mathsf{Pr}\left(\mathbf{n}\right) \sum_{d\geq 0} \mathtt{n}_d C_{\mathsf{Mul}}(d,p,q), \end{aligned}$$

where inequality (a) is due to the fact that each of the individual mutual information terms is maximized by the capacity as shown in Lemma 5.2. Equality (b) follows from the fact that in the sum over i, it only matters how often some d occurs in $\mathbf{d}$. By definition, this is equal to

the drawing frequency $\boldsymbol{n}$, which is a function of $\boldsymbol{d}$, which allows to replace the sum over $\mathbf{d}$ by a sum over $\mathbf{n}$. For an arbitrary $\epsilon > 0$ denote now by $\mathcal{N}_\epsilon$ the event on the random variable $\boldsymbol{n}$ that $\sum_{d\geq 0} \left|\frac{\mathtt{n}_d}{M} - \nu_d\right| \leq \epsilon/4$. We can thus split the sum over $\mathbf{n}$ into two parts

$$\begin{aligned} I(\boldsymbol{X};\boldsymbol{Z}) &\leq L\left(\sum_{\mathbf{n}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{n}\right) \sum_{d\geq 0} \mathtt{n}_d C_{\mathsf{Mul}}(d,p,q) + \sum_{\mathbf{n}\notin\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{n}\right) \sum_{d\geq 0} \mathtt{n}_d C_{\mathsf{Mul}}(d,p,q)\right) \\ &\overset{(a)}{\leq} L\left(\frac{M\epsilon}{4} + M\sum_{d\geq 0} \nu_d C_{\mathsf{Mul}}(d,p,q) + \sum_{\mathbf{n}\notin\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{n}\right) \sum_{d\geq 0} \mathtt{n}_d C_{\mathsf{Mul}}(d,p,q)\right) \\ &\overset{(b)}{\leq} ML\left(\frac{\epsilon}{4} + \sum_{d\geq 0} \nu_d C_{\mathsf{Mul}}(d,p,q) + \mathsf{Pr}\left(\boldsymbol{n}\notin\mathcal{N}_\epsilon\right)\right), \end{aligned}$$

where (a) is due to the fact that for all $\mathbf{n}\in\mathcal{N}_\epsilon$, we have

$$\sum_{d\geq 0} (\mathtt{n}_d - M\nu_d) C_{\mathsf{Mul}}(d,p,q) \leq \sum_{d\geq 0} |\mathtt{n}_d - M\nu_d| \leq \frac{M\epsilon}{4}.$$

Inequality (b) follows from the fact that $C_{\mathsf{Mul}}(d,p,q) \leq 1$ and $\sum_{d\geq 0}\mathtt{n}_d = M$. By definition of frequency convergence $\mathsf{Pr}\left(\boldsymbol{n}\notin\mathcal{N}_\epsilon\right)\to 0$ as $M\to\infty$ for all $\epsilon > 0$, any achievable rate R satisfies $R \leq C_{\mathsf{OPM}}(\boldsymbol{\nu},p,q)$, which proves the converse bound on the capacity.

On the other hand, we prove achievability using a random coding argument as in [CT06; Sha48]. Let $\mathcal{C} = \{\boldsymbol{X}(1),\ldots,\boldsymbol{X}(q^{MLR})\} \subseteq \Sigma_q^{M\times L}$ be a random codebook with code rate R, where each $\boldsymbol{X}(i)\in\Sigma_q^{M\times L}$ is chosen independently and uniform over all possible words in $\Sigma_q^{M\times L}$. We denote the individual sequences of $\boldsymbol{X}(i)$ by $\boldsymbol{X}(i) = (\boldsymbol{x}_1(i),\ldots,\boldsymbol{x}_M(i))$ For a given codebook, we associate an encoder $\mathsf{enc}_{\mathcal{C}}(W) = \boldsymbol{X}(W)$ and decoder $\widehat{W} = \mathsf{dec}_{\mathcal{C}}(\boldsymbol{Z})$ that we will describe in the following. To define the decoder, we fix an $\epsilon > 0$ and introduce the notion of typical sequences over the ordered multinomial channel as follows.

$$\mathcal{T}_{\mathsf{OPM}}^{M,L,\epsilon}(p,q) \triangleq \left\{ \begin{array}{l} ((\mathbf{x}_1,\ldots,\mathbf{x}_M),(\mathbf{z}_1,\ldots,\mathbf{z}_M)) \in \Sigma_q^{M\times L} \times \left(\Sigma_q^{d_1\times L}\times\cdots\times\Sigma_q^{d_M\times L}\right) : \\ \quad d_1,\ldots,d_M\in\mathbb{N}_0 \wedge \left|\left\{i\in[M] : (\mathbf{x}_i,\mathbf{z}_i)\in\mathcal{T}_{\mathsf{Mul}}^{L,\epsilon}(d_i,p,q)\right\}\right| \geq (1-\epsilon)M \end{array} \right\}.$$

In other words, a pair of input sequences and output clusters $((\mathbf{x}_1,\ldots,\mathbf{x}_M),(\mathbf{z}_1,\ldots,\mathbf{z}_M))$ is jointly typical, if almost all of the individual sequences $\mathbf{x}_i,\mathbf{z}_i$ are jointly typical with respect to the multinomial channel. The decoder $\mathsf{dec}_{\mathcal{C}}(\boldsymbol{Z})$ then decodes to $\widehat{W}$, if $\boldsymbol{X}(\widehat{W})$ is the unique codeword that is jointly typical with $\boldsymbol{Z} = (\boldsymbol{z}_1,\ldots,\boldsymbol{z}_M)$ with respect to the ordered multinomial channel. If there is none or more than two codewords that are jointly typical with $\boldsymbol{Z}$, than the decoder outputs a failure, resulting in a decoding error. The probability of error, averaged over all codebooks is given by

$$\mathsf{Pr}\left(\mathsf{Err}\right) = \sum_{\mathcal{C}} \mathsf{Pr}\left(\mathcal{C}\right)\mathsf{Pr}\left(\mathsf{Err}|\mathcal{C}\right) = \mathsf{Pr}\left(\mathsf{Err}|W=1\right),$$

where the last equality is due to the symmetry of the choice of random codebooks, see, e.g., [CT06, Ch. 7.7]. The two possible error events are that $\boldsymbol{X}(1)$ is not jointly typical with $\boldsymbol{Z}$ or that one of the other codewords is jointly typical with respect to $\boldsymbol{Z}$. Denote by $\mathcal{J}_i$ the event that the i-th

codeword $\boldsymbol{X}(i)$ is jointly typical with $\boldsymbol{Z}$, i.e., $(\boldsymbol{X}(i), \boldsymbol{Z}) \in \mathcal{T}_{\mathsf{OPM}}^{M,L,\epsilon}(p,q)$ and by $\mathcal{J}_i^{\mathsf{c}}$ the complement event. By the union bound we obtain

$$\mathsf{Pr}\left(\mathsf{Err}|W=1\right) \leq \mathsf{Pr}\left(\mathcal{J}_1^{\mathsf{c}} \cup \bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right) \leq 1 - \mathsf{Pr}\left(\mathcal{J}_1|W=1\right) + \mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right).$$

We first bound $\mathsf{Pr}\left(\mathcal{J}_1|W=1\right)$ from below. We demarginalize with respect to the drawing distribution $\boldsymbol{d}$ and obtain

$$\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) = \sum_{\mathbf{d}} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\mathcal{J}_1|W=1, \boldsymbol{d}=\mathbf{d}\right) \geq \sum_{\mathbf{d}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\mathcal{J}_1|W=1, \boldsymbol{d}=\mathbf{d}\right),$$

where we used that $\boldsymbol{d}$ is independent of W. Note that the event $\mathcal{N}_\epsilon$ was originally defined as an event on the drawing frequency $\boldsymbol{n}$, however since $\boldsymbol{n}$ is a function of $\boldsymbol{d}$, one can also define it on the drawing composition $\boldsymbol{d}$. We can use the fact that, given $\boldsymbol{d}=\mathbf{d}$, the number of jointly typical pairs over the multinomial channel $\left|\left\{i\in[M] : (\boldsymbol{x}_i(1), \boldsymbol{z}_i) \in \mathcal{T}_{\mathsf{Mul}}^{L,\epsilon}(\mathtt{d}_i, p, q)\right\}\right|$ is the sum of M independent random Bernoulli random variables with success probabilities $\pi_i \triangleq \mathsf{Pr}\left((\boldsymbol{x}_i(1), \boldsymbol{z}_i) \in \mathcal{T}_{\mathsf{Mul}}^{L,\epsilon}(\mathtt{d}_i, p, q) | W=1, d_i=\mathtt{d}_i\right)$. From the results about jointly typical sequences [CT06, Thm. 7.6.1] we know that for all $\epsilon>0$ and $i\in[M]$, it holds that $\pi_i > 1-\epsilon/2$ for all $L \geq L_{\mathtt{d}_i}$, as $\boldsymbol{z}_i$ is the result of transmitting $\boldsymbol{x}_i(1)$ over the multinomial channel. As $\max_{i\in[M]} L_{\mathtt{d}_i}$ might increase with M, we focus our attention to a subset of multinomial channels whose number of draws is bounded from above by a large, but finite quantity. To this end, let D_ϵ be chosen such that $\sum_{d\geq D_\epsilon} \nu_d < \epsilon/4$. We have that for all $\mathbf{d}\in\mathcal{N}_\epsilon$, the number of $i\in[M]$ with $\mathtt{d}_i < D_\epsilon$ is at least

$$\sum_{d=0}^{D_\epsilon-1} \mathtt{n}_d \geq M \sum_{d=0}^{D_\epsilon-1} \nu_d - \frac{M\epsilon}{4} > M\left(1-\frac{\epsilon}{2}\right).$$

Thus, at least $M(1-\epsilon/2)$ Bernoulli variables have success probability at least $\pi_i > 1-\epsilon/2$ for all $L \geq \max_{0\leq d<D_\epsilon} L_d$ (which is finite, i.e., not a function of L) and, we obtain

$$\begin{aligned}
&\mathsf{Pr}\left(\mathcal{J}_1|W=1, \boldsymbol{d}=\mathbf{d}\right) \\
&\geq \sum_{i=M-M\epsilon}^{M-\frac{M\epsilon}{2}} \binom{M-\frac{M\epsilon}{2}}{i} \left(1-\frac{\epsilon}{2}\right)^i \left(\frac{\epsilon}{2}\right)^{M-\frac{M\epsilon}{2}-i} = \sum_{i=0}^{\frac{M\epsilon}{2}} \binom{M-\frac{M\epsilon}{2}}{i} \left(1-\frac{\epsilon}{2}\right)^{M-\frac{M\epsilon}{2}-i} \left(\frac{\epsilon}{2}\right)^i \\
&= 1 - \sum_{i=\frac{M\epsilon}{2}+1}^{M-\frac{M\epsilon}{2}} \binom{M-\frac{M\epsilon}{2}}{i} \left(1-\frac{\epsilon}{2}\right)^{M-\frac{M\epsilon}{2}-i} \left(\frac{\epsilon}{2}\right)^i \overset{(c)}{\geq} 1 - \mathrm{e}^{-2\left(M-\frac{M\epsilon}{2}\right)\left(\frac{\epsilon^2}{4-2\epsilon}\right)^2},
\end{aligned}$$

for all $0<\epsilon<1$ and large enough L. Here we used Lemma A.4 to bound the binomial tail in inequality (c). Thus, finally, for any $\epsilon>0$ and large enough L,

$$\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) \geq \left(1 - \mathrm{e}^{-2\left(M-\frac{M\epsilon}{2}\right)\left(\frac{\epsilon^2}{4-2\epsilon}\right)^2}\right) \mathsf{Pr}\left(\boldsymbol{d}\in\mathcal{N}_\epsilon\right),$$

where the first term approaches 1 as $M\to\infty$ for any $\epsilon>0$ and the second term approaches 1 as well by assumption of convergence of the drawing frequency. It follows that $\mathsf{Pr}\left(\mathcal{J}_1|W=1\right) \to 1$.

We now bound $\mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i | W=1\right)$ from above and perform the same demarginalization with respect to the drawing composition $\boldsymbol{d}$ to obtain

$$\begin{aligned}\mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right) &\leq \mathsf{Pr}\left(\boldsymbol{d} \notin \mathcal{N}_\epsilon\right) + \sum_{\mathbf{d}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1, \boldsymbol{d}=\mathbf{d}\right) \\ &\overset{(d)}{\leq} \mathsf{Pr}\left(\boldsymbol{d} \notin \mathcal{N}_\epsilon\right) + q^{MLR} \sum_{\mathbf{d}\in\mathcal{N}_\epsilon} \mathsf{Pr}\left(\mathbf{d}\right) \mathsf{Pr}\left(\mathcal{J}_2 | W=1, \boldsymbol{d}=\mathbf{d}\right),\end{aligned}$$

where we used the union bound to prove inequality (d), together with the fact that due to the identical and independent choice of codewords, $\mathsf{Pr}\left(\mathcal{J}_2 | W=1, \boldsymbol{d}=\mathbf{d}\right) = \ldots = \mathsf{Pr}\left(\mathcal{J}_{q^{MLR}} | W=1, \boldsymbol{d}=\mathbf{d}\right)$. Again, the number of jointly typical pairs of the multinomial channel is distributed as the sum of independent Bernoulli variables with success probability π_i. For the case of the event $\mathcal{J}_2$, the $\boldsymbol{x}_i(2)$ are independent of $\boldsymbol{z}_i$ and thus for all $i \in [M]$, $\pi_i < q^{-L(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon)}$ for $L \geq L_{\mathtt{d}_i}$ [CT06, Thm. 7.6.1]. It follows that

$$\begin{aligned}\mathsf{Pr}\left(\mathcal{J}_2 | W=1, \boldsymbol{d}=\mathbf{d}\right) &= \sum_{\mathcal{I}\subseteq[M]:|\mathcal{I}|\geq M(1-\epsilon)} \prod_{i\in\mathcal{I}} \pi_i \prod_{j\notin\mathcal{I}} (1-\pi_j) \leq \sum_{\mathcal{I}\subseteq[M]:|\mathcal{I}|=M(1-\epsilon)} \prod_{i\in\mathcal{I}} \pi_i \\ &\leq \binom{M}{M(1-\epsilon)} \max_{\mathcal{I}\subseteq[M]:|\mathcal{I}|=M(1-\epsilon)} \prod_{i\in\mathcal{I}} \pi_i.\end{aligned}$$

Now abbreviate $\mathcal{I}(\epsilon) = \{i \in \mathcal{I} : \mathtt{d}_i < D_\epsilon\}$. We can bound the product over π_i by

$$\prod_{i\in\mathcal{I}} \pi_i \leq \prod_{i\in\mathcal{I}(\epsilon)} \pi_i < \prod_{i\in\mathcal{I}(\epsilon)} q^{-L(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon)} = q^{-L\sum_{i\in\mathcal{I}(\epsilon)}(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon)}$$

for all $L \geq \max_{0\leq d<D_\epsilon} L_d$. Recall from earlier that for all $\mathbf{d} \in \mathcal{N}_\epsilon$, the number of $i \in [M]$ with $\mathtt{d}_i < D_\epsilon$ is at least $M\left(1-\frac{\epsilon}{2}\right)$ and thus $|\mathcal{I}(\epsilon)| \geq M(1-\frac{3\epsilon}{2})$ for any choice of $\mathcal{I}$. Analyzing the exponent of the error probability expression above, we find that

$$\begin{aligned}L\sum_{i\in\mathcal{I}(\epsilon)} (C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon) &= L\left(\sum_{i=1}^{M}(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon) - \sum_{i\notin\mathcal{I}(\epsilon)} (C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon)\right) \\ &\geq L\left(\sum_{i=1}^{M}(C_{\mathsf{Mul}}(\mathtt{d}_i,p,q)-\epsilon) - \frac{3M\epsilon}{2}\right) = L\sum_{d\geq 0} \mathtt{n}_d C_{\mathsf{Mul}}(d,p,q) - \frac{5ML\epsilon}{2} \\ &\geq ML\sum_{d\geq 0} \nu_d C_{\mathsf{Mul}}(d,p,q) - \frac{7ML\epsilon}{2} = MLC_{\mathsf{OPM}}(\boldsymbol{\nu},\boldsymbol{p},\boldsymbol{q}) - \frac{7ML\epsilon}{2}.\end{aligned}$$

Using further that $\binom{M}{M(1-\epsilon)} \leq 2^M$, it follows that for any $0 < \epsilon < 1$ and large enough M the error probability of the second error event is at most

$$\mathsf{Pr}\left(\bigcup_{i=2}^{q^{MLR}} \mathcal{J}_i \middle| W=1\right) \leq \mathsf{Pr}\left(\boldsymbol{d} \notin \mathcal{N}_\epsilon\right) + 2^M q^{-ML(C_{\mathsf{OPM}}(\boldsymbol{\nu},p,q)-R-7\epsilon/2)}.$$

As we can choose ϵ as small as desired, for any $R < C_{\mathsf{OPM}}(\boldsymbol{\nu},p,q)$, the above error probability vanishes and thus the average error probability of the random code ensemble $\mathsf{Pr}\left(\mathsf{Err} | W=1\right)$ goes to 0 as $M \to \infty$. Henceforth, we can conclude that there exists at least one code of rate R that has vanishing error probability. □

A.4 Alternative Proofs for Results on Cost-Uniform Graphs

We provide an alternative proof of Corollary 7.37 that makes use of Wielandt's theorem and avoids the employment of Lemma 7.35. This proof is of purely academical nature and is meant to provide the reader with more background on irreducible matrices.

Alternative proof of Corollary 7.37. Wielandt's theorem (Theorem 7.21) states that $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) \leq \rho_G(x)$ with equality if and only if there exist $\theta, \theta_1, \theta_2, \ldots, \theta_{|\mathcal{V}|}$ such that

$$\boldsymbol{P}_G(x\mathrm{e}^{\mathrm{i}\phi}) = \mathrm{e}^{\mathrm{i}\theta}\boldsymbol{D}^{-1}\boldsymbol{P}_G(x)\boldsymbol{D},$$

where $\boldsymbol{D}$ is a diagonal matrix with entries $[\boldsymbol{D}]_{jj} = \mathrm{e}^{\mathrm{i}\theta_j}$. If G is cost-uniform, there can be at most one cost on edges connecting any two states, so each nonzero entry $[\boldsymbol{P}_G(x)]_{ij}$ of $\boldsymbol{P}_G(x)$ is a monomial of the form $N_{ij}x^{\tau_{ij}}$, where τ_{ij} is the cost of an edge from v_i to v_j, and N_{ij} is the number of such edges. Due to the coboundary condition, τ_{ij} can be written as

$$\tau_{ij} = b + B(v_j) - B(v_i).$$

For any $x > 0$ and any $0 \leq \phi < 2\pi$, we can write

$$\begin{aligned}[\boldsymbol{P}_G(x\mathrm{e}^{\mathrm{i}\phi})]_{ij} &= N_{ij}x^{\tau_{ij}}\mathrm{e}^{\mathrm{i}\phi\tau_{ij}} = N_{ij}x^{\tau_{ij}}\mathrm{e}^{\mathrm{i}\phi(b+B(v_j)-B(v_i))} = \mathrm{e}^{\mathrm{i}\phi(b+B(v_j)-B(v_i))}[\boldsymbol{P}_G(x)]_{ij} \\ &= \mathrm{e}^{\mathrm{i}(\theta+\theta_i-\theta_j)}[\boldsymbol{P}_G(x)]_{ij}\end{aligned}$$

where $\theta = \phi b$ and $\theta_k = \phi B(v_k)$ for all k. This confirms that the condition in Wielandt's theorem holds and, therefore, $\rho_G(x\mathrm{e}^{\mathrm{i}\phi}) = \rho_G(x)$ for all $0 \leq \phi < 2\pi$. □

We also provide a standalone proof of Corollary 7.38 on the log-log-linearity of the Perron root.

Alternative proof of Corollary 7.38. Let $0 < x_1 < x_2 < 1$ and, for $0 \leq s \leq 1$, define the matrix $\Pi(s) \triangleq \boldsymbol{P}_G(\mathrm{e}^{s\log(x_2/x_1)+\log x_1})$. Notice that $\Pi(0) = \boldsymbol{P}_G(x_1)$ and $\Pi(1) = \boldsymbol{P}_G(x_2)$. We will apply [Nus86, Corollary 1.2] to prove log-linearity of $\Pi(s)$. To start with, as cost-uniformity implies the coboundary condition, each entry $[\boldsymbol{P}_G(\mathrm{e}^s)]_{ij}$ has the form $[\boldsymbol{P}_G(\mathrm{e}^s)]_{ij} = N_{ij}\mathrm{e}^{s\tau_{ij}}$, where N_{ij} is the number of edges from v_i to v_j and $\tau_{ij} = b + B(v_j) - B(v_i)$. We directly see that $[\boldsymbol{P}_G(\mathrm{e}^s)]_{ij}$ is log-convex in s. Since scaling does not change convexity, so are the entries of $\Pi(s)$. We can thus apply [Nus86, Corollary 1.2], which implies that

$$\lambda(\Pi(s)) \leq \lambda(\Pi(0))^{1-s}\lambda(\Pi(1))^s$$

with equality for all $0 < s < 1$ if and only if there exist constants $c \in \mathbb{R}$ and $C(1), \ldots, C(|\mathcal{V}|) > 0$ such that $[\Pi(0)]_{ij} = c \cdot C^{-1}(i)C(j)[\Pi(1)]_{ij}$ for all i, j. That is, these constants need to satisfy

$$N_{ij}x_1^{\tau_{ij}} = N_{ij}c \cdot C^{-1}(i)C(j)x_2^{\tau_{ij}},$$

or, equivalently,

$$\tau_{ij}\log\left(\frac{x_1}{x_2}\right) - \log c + \log C(i) - \log C(j) = 0.$$

This condition can directly be identified as the coboundary condition, which is fulfilled by assumption of cost-diversity. It follows that $\lambda(\Pi(s)) = \rho_G(x_1)\left(\rho_G(x_2)/\rho_G(x_1)\right)^s$ for all $0 \leq s \leq 1$ and $0 < x_1 < x_2 < 1$. Or, reversing the linear scaling in s, $\rho_G(\mathrm{e}^s) = \alpha\beta^s$ for some $\alpha, \beta \in \mathbb{R}$, proving that $\log\rho_G(\mathrm{e}^s)$ is linear and, henceforth, $\rho_G(x)$ is log-log-linear.

□

A.5 Periodicity of Strongly Connected Graphs

We first prove that the definition of periodicity from [MRS01] follows from Definition 7.3.

Lemma A.8. *Let $G = (\mathcal{V}, \mathcal{E}, \sigma, \tau)$ be a strongly connected graph with largest period d. Then, the greatest common divisor of all cycle lengths is d.*

Proof. To start with, we note that the length m of each cycle must be divisible by d, since otherwise the twice repetition of this cycle would not have a length congruent to that of the single cycle modulo d. Analogously to the proof of Lemma 7.33, we can prove the existence of two cycles at the same state whose length differs in precisely d. This implies that the the greatest common divisor of the path lengths is d, which proves the statement. □

We proceed with a variation of the Chinese Remainder Theorem. The original Chinese Remainder Theorem can be found in, e.g., [IR90, Section 3.4].

Lemma A.9. *Let $c \in \mathbb{N}$ and $(m_1, \tau_1), (m_2, \tau_2), \ldots,$ be pairs of integers $(m_i, \tau_i) \in \mathbb{N}^2$. Denote by d the greatest common divisor of all m_i. If these pairs satisfy*

$$m_i \tau_j \equiv m_j \tau_i \pmod{(cd)}$$

for all i, j, then there exists $b \in \mathbb{Z}$ such that for all i,

$$d\tau_i \equiv m_i b \pmod{(cd)}.$$

Further, any $b' \in \mathbb{Z}$ with $b' \equiv b \pmod{c}$ has the same property.

Proof. We prove the statement by a direct construction. Assume without loss of generality that $\gcd(m_1, \ldots, m_n) = d$ for some $n \in \mathbb{N}$. This is possible, since there exist finitely many m_i such that their greatest common divisor is equal to d. By Bézout's identity, there exist $z_1, \ldots, z_n \in \mathbb{Z}$ with $z_1 m_1 + \cdots + z_n m_n = d$. Choosing $b = z_1\tau_1 + \ldots z_n\tau_n$, we obtain for any $1 \leq i \leq n$,

$$m_i b = z_i m_i \tau_i + \sum_{j \neq i} z_j m_i \tau_j = \tau_i \left(d - \sum_{j \neq i} z_j m_j \right) + \sum_{j \neq i} z_j m_i \tau_j = \tau_i d + \sum_{j \neq i} z_j (m_i \tau_j - m_j \tau_i).$$

By assumption $m_i\tau_j - m_j\tau_i \equiv 0 \pmod{(cd)}$ and thus $m_i b \equiv \tau_i d \pmod{(cd)}$. On the other hand, for any $i > n$, we set $z_i = 0$ and obtain via a similar argument

$$m_i b = z_i m_i \tau_i + \sum_{j=1}^{n} z_j m_i \tau_j = \tau_i d + \sum_{j=1}^{n} z_j (m_i \tau_j - m_j \tau_i),$$

which implies that $m_i b \equiv \tau_i d \pmod{(cd)}$. This concludes the proof.

□

A.6 Concavity and Maximality of Fixed-Length Capacity

Proposition A.10. *Let G be a strongly connected, deterministic, cost-diverse graph. Then, $C_G(\alpha)$ is a concave function in α and its maximum is equal to $C_G(\alpha^*) = C_G$, where $\alpha^* = 2^{C_G} / \rho'_G(2^{-C_G})$.*

Proof. To start with, $C_G(\alpha)$ is linear in the interval $0 \leq \alpha < \alpha_G^{\mathsf{lo}}$. In the interval $\alpha_G^{\mathsf{lo}} \leq \alpha < \alpha_G^{\mathsf{up}}$, $C_G(\alpha) = -\log x_0(\alpha) + \alpha \log \rho_G(x_0(\alpha))$, where $x_0(\alpha)$ is the unique positive solution to $f(x) = \alpha^{-1}$ with $f(x) \triangleq x\rho_G'(x)/\rho_G(x)$. Notice that $\rho_G(x) > 0$ for all $x > 0$ and thus, by Lemma 7.24, $f(x)$ is analytic for all $x > 0$. Further, as in the proof of Lemma 7.50, we can show that $\frac{\partial}{\partial x} f(x) > 0$, which means that $x_0(\alpha)$ is analytic in α and also strictly monotonically decreasing in α. Therefore, for $\alpha_G^{\mathsf{lo}} \leq \alpha < \alpha_G^{\mathsf{up}}$,

$$\begin{aligned}
\frac{\partial}{\partial \alpha} C_G(\alpha) &= -\frac{x_0'(\alpha)}{x_0(\alpha)} + \log \rho_G(x_0(\alpha)) + \alpha \frac{x_0'(\alpha)\rho_G'(x_0(\alpha))}{\rho_G(x_0(\alpha))} \\
&= -\frac{x_0'(\alpha)}{x_0(\alpha)} + \log \rho_G(x_0(\alpha)) + \alpha f(x_0(\alpha)) \frac{x_0'(\alpha)}{x_0(\alpha)} \\
&\overset{(a)}{=} \log \rho_G(x_0(\alpha)),
\end{aligned}$$

where we used in (a) that $f(x_0(\alpha)) = \alpha^{-1}$ by definition. Since $x_0(\alpha)$ is strictly monotonically decreasing in α and also $\rho_G(x)$ and the logarithm are strictly monotone functions (see Lemma 7.26), $C_G(\alpha)$ is strictly concave in the considered interval. By definition of α_G^{lo}, $C_G(\alpha) \to \alpha_G^{\mathsf{lo}} \log \rho_G(1)$, as α approaches α_G^{lo} from both the left and right, proving continuity of $C_G(\alpha)$. Therefore, $C_G(\alpha)$ is a concave function on the full interval $0 \leq \alpha \leq \alpha_G^{\mathsf{up}}$. From the above derivation of the derivative of $C_G(\alpha)$, we further see that α^* with $\rho_G(x_0(\alpha^*)) = 1$ is a unique stationary point of $C_G(\alpha)$ with capacity $C_G(\alpha^*) = -\log x_0(\alpha^*) = C_G$. It follows that α^* is the unique solution for α to the system of two equations $f(x) = \alpha^{-1}$ and $\rho_G(x) = 1$, $x > 0$. The above exposition proves that the solution to these equations are α^* and $x_0(\alpha^*)$, where α^* is as given in the statement. □

Appendix B

Glossary

Abbreviations

Notation	Description
BCH	Bose-Chaudhuri-Hocquenghem
DNA	Deoxyribonucleic acid
GV	Gilbert-Varshamov
MDS	Maximum-distance-separable
OPM	Ordered parallel multinomial channel
QSC	q-ary symmetric channel
SP	Sphere-packing
TPC	Tensor product code
UPM	Unordered parallel multinomial channel
VT	Varshamov-Tenengolts

Global Notation

Notation	Description
$\mathbb{N}$	Set of natural numbers
$\mathbb{Z}$	Set of positive and negative integer numbers
$\mathbb{Q}$	Set of rational numbers
$\mathbb{R}$	Set of real numbers, $\mathbb{R}^+$ for positive real numbers
$\mathbb{C}$	Set of complex numbers
$[n]$	Set of integers up to n
$\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}$	DNA bases: Adenine (A), Cytosine (C), Guanine (G), Thymine (T)
Σ_q	Alphabet of size q with letters $\Sigma_q = \{0, 1, \ldots, q-1\}$.
$\mathbb{F}_q$	Finite field of size q
$\mathcal{C}$	Code over some space
$\mathsf{wt}_\mathsf{H}(\boldsymbol{x})$	Hamming weight of the vector $\boldsymbol{x}$
$d_\mathsf{H}(\boldsymbol{x}, \boldsymbol{y})$	Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$

Glossary for Part 1

Notation	Description
L	Length of DNA strand, measured in number of consecutive nucleotides
M	Number of DNA strands that are stored in the archive
β	Archive density
$\mathcal{S}$	Data set of M DNA strands, each of length L
$\mathcal{S}_{\mathsf{C}}$	Set of sequences, which have been reconstructed correctly
$\mathcal{S}_{\mathsf{L}}$	Set of sequences, which have been lost after reconstruction
$\mathcal{S}_{\mathsf{E}}$	Set of sequences, which have been reconstructed with errors
$\mathcal{X}_M^L$	Set of all possible data sets with M sequences, each of length L
$r(\mathcal{C})$	Redundancy of a code $\mathcal{C} \subseteq \mathcal{X}_M^L$
$\mathbb{T}$	Error type
$(s,t,u)_{\mathbb{T}}$	Parameters of the adversarial DNA storage channel
$B^{\mathbb{T}}(\boldsymbol{x},u)$	Error ball of possible outcomes after at most u errors of type $\mathbb{T}$ in $\boldsymbol{x}$
$S^{\mathbb{T}}(\boldsymbol{x},u)$	Error sphere of possible outcomes after exactly u errors of type $\mathbb{T}$ in $\boldsymbol{x}$
$B^{\mathbb{T}}(\mathcal{S},s,t,u)$	Error ball of possible outcomes from the $(s,t,u)_{\mathbb{T}}$ DNA storage channel
$V^{\mathbb{T}}(\mathcal{S},s,t,u)$	Set of all words which have intersection error balls with $\mathcal{S}$
$\overline{V}^{\mathbb{T}}(s,t,u)$	Average of the number of sets with intersecting DNA error balls
$V^{\mathbb{T}}(u)$	Maximum of the number of sequences with intersecting $\mathbb{T}$ error balls
$\overline{S}^{\mathbb{D},t}(u)$	t-th moment of the deletion sphere size distribution
$\|\|\boldsymbol{x}\|\|$	Number of runs in the vector $\boldsymbol{x}$
$\mathcal{I}_M^L$	Code of indexed sequences
$\boldsymbol{I}(i)$	Binary representation of the index $i-1$
$\boldsymbol{I}(\mathcal{S})$	Set of indices of sequences in the set $\mathcal{S}$
$\mathsf{pref}_k(\boldsymbol{x})$	Prefix of length k of the vector $\boldsymbol{x}$
$\mathsf{suff}_k(\boldsymbol{x})$	Suffix of length k of the vector $\boldsymbol{x}$
$\mathsf{MDS}[M,k,]$	MDS code of length M and dimension k
$\boldsymbol{v}(\mathcal{S})$	Set-indicator vector of length 2^L
$s_{\mathsf{VT}}(\boldsymbol{x})$	Varshamov-Tenengolts checksum

Glossary for Part 2

Notation	Description
L	Length of DNA strand, measured in number of consecutive nucleotides
M	Number of DNA strands that are stored in the archive
β	Archive density, $\beta = \log_q(M)/L$
N	Total number of drawn sequences in the DNA storage channel
c	Sequencing depth: average number of times, a sequence is read
$\boldsymbol{X} = (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M)$	Input sequences
$\boldsymbol{Y} = (\boldsymbol{y}_1,\ldots,\boldsymbol{y}_M)$	Output clusters
$\boldsymbol{R} = (\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N)$	Output sequences
$\mathsf{Pr}\,(x = \mathsf{x})$	Probability of an event

$I(x;y)$	Mutual information of the random variables x and y
$H(x)$	Entropy of the random variable x
$\mathsf{E}[x]$	Expected value of the random variable x
$\mathsf{V}[x]$	Variance of the random variable x
$\{\{\bullet\}\}$	Multiset
$C_{\mathsf{Mul}}(d,p,q)$	Capacity of the multinomial channel with d draws
$C_{\mathsf{OPM}}(\boldsymbol{\nu},\beta,p,q)$	Capacity of the ordered parallel multinomial channel
$C_{\mathsf{UPM}}(\boldsymbol{\nu},\beta,p,q)$	Capacity of the unordered parallel multinomial channel
$C_{\mathsf{DNA}}(c,\beta,p,q)$	Capacity of the DNA storage channel
$\mathcal{T}_{\mathsf{Mul}}^{L,\epsilon}(d,p,q)$	Typical sequences over the multinomial channel
$\mathcal{T}_{\mathsf{OPM}}^{M,L,\epsilon}(p,q)$	Typical sequences over the ordered parallel multinomial channel
$\mathcal{T}_{\mathsf{UPM}}^{M,L,\epsilon}(p,q)$	Typical sequences over the unordered parallel multinomial channel

Glossary for Part 3

Notation	Description
G	Labeled and directed graph
$\mathcal{V}$	Vertex set of a graph
$\mathcal{E}$	Set of all directed edges of a graph
σ	Labels of graph edges
τ	Costs or weights of graph edges
$\boldsymbol{P}_G(x)$	Cost-enumerator matrix of the graph G
$\rho_G(x)$	Spectral radius of the cost-enumerator matrix $\boldsymbol{P}_G(x)$
$\mathcal{L}_{G,v}(t)$	Cost-t follower set of the vertex v in the graph G
$\mathcal{L}_{G,v}(t,n)$	Length-n and cost-t follower set of the vertex v in the graph G
$N_{G,v}(t)$	Size of the cost-t follower set of the vertex v in the graph G
$N_{G,v}(t,n)$	Size of the length-n and cost-t follower set of the vertex v in the graph G
C_G	Variable-length capacity
$C_G(\alpha)$	Fixed-length capacity
$\boldsymbol{F}_G(x,y)$	Generating function of the length-n and cost-t follower set sizes
$\boldsymbol{I}$	Identity matrix
$\mathrm{rank}(\boldsymbol{P})$	Rank of the matrix $\boldsymbol{P}$
$\mathrm{adj}(\boldsymbol{P})$	Adjoint (or adjugate) of the matrix $\boldsymbol{P}$
$\mathrm{tr}(\boldsymbol{P})$	Trace of the matrix $\boldsymbol{P}$
gcd	Greatest common divisor
mod	Modulo operator
i	Imaginary unit

Bibliography

[Abr+19] Mahed Abroshan, Ramji Venkataramanan, Lara Dolecek, and Albert Guillen i Fabregas. "Coding for Deletion Channels with Multiple Traces". In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 1372–1376. DOI: 10.1109/ISIT.2019.8849647.

[Ach+15] Jayadev Acharya, Hirakendu Das, Olgica Milenkovic, Alon Orlitsky, and Shengjun Pan. "String Reconstruction from Substring Compositions". In: *SIAM Journal on Discrete Mathematics* 29.3 (Jan. 2015), pp. 1340–1371. DOI: 10.1137/140962486.

[Aga+20] Abhishek Agarwal, Olgica Milenkovic, Srilakshmi Pattabiraman, and Joao Ribeiro. "Group Testing with Runlength Constraints for Topological Molecular Storage". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 132–137. DOI: 10.1109/ISIT44484.2020.9174502.

[Ahl86] R. Ahlswede. "Arbitrarily Varying Channels with States Sequence Known to the Sender". In: *IEEE Transactions on Information Theory* 32.5 (Sept. 1986), pp. 621–629. DOI: 10.1109/TIT.1986.1057222.

[AKN14] Angela Angeleska, Sabrina Kleessen, and Zoran Nikoloski. "The Sequence Reconstruction Problem". In: *Discrete and Topological Models in Molecular Biology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 23–43. DOI: 10.1007/978-3-642-40193-0_2.

[All+12] Morten E. Allentoft, Matthew Collins, David Harker, James Haile, Charlotte L. Oskam, Marie L. Hale, Paula F. Campos, Jose A. Samaniego, M. Thomas P. Gilbert, Eske Willerslev, Guojie Zhang, R. Paul Scofield, Richard N. Holdaway, and Michael Bunce. "The Half-Life of DNA in Bone: Measuring Decay Kinetics in 158 Dated Fossils". In: *Proceedings of the Royal Society B: Biological Sciences* 279.1748 (Dec. 2012), pp. 4724–4733. DOI: 10.1098/rspb.2012.1745.

[Ana+19] Leon Anavy, Inbal Vaknin, Orna Atar, Roee Amit, and Zohar Yakhini. "Data Storage in DNA with Fewer Synthesis Cycles Using Composite DNA Letters". In: *Nature Biotechnology* 37.10 (Oct. 2019), pp. 1229–1236. DOI: 10.1038/s41587-019-0240-x.

[Ant+20] Philipp L. Antkowiak, Jory Lietard, Mohammad Zalbagi Darestani, Mark M. Somoza, Wendelin J. Stark, Reinhard Heckel, and Robert N. Grass. "Low Cost DNA Data Storage Using Photolithographic Synthesis and Advanced Information Reconstruction and Error Correction". In: *Nature Communications* 11.1 (Dec. 2020), p. 5345. DOI: 10.1038/s41467-020-19148-3.

[Ash90] Robert B. Ash. *Information Theory*. New York: Dover Publications, 1990.

[AVF18] Mahed Abroshan, Ramji Venkataramanan, and Albert Guillen i Fabregas. "Coding for Segmented Edit Channels". In: *IEEE Transactions on Information Theory* 64.4 (Apr. 2018), pp. 3086–3098. DOI: 10.1109/TIT.2017.2788143. arXiv: 1701.06341.

[Ban+01] Carter Bancroft, Timothy Bowler, Brian Bloom, and Catherine Taylor Clelland. "Long-Term Storage of Information in DNA". In: *Science* 293.5536 (Sept. 2001), pp. 1763–1765. DOI: 10.1126/science.293.5536.1763c.

[Bau95] E. Baum. "Building an Associative Memory Vastly Larger than the Brain". In: *Science* 268.5210 (Apr. 1995), pp. 583–585. DOI: 10.1126/science.7725109.

[BBT59] David Blackwell, Leo Breiman, and A. J. Thomasian. "The Capacity of a Class of Channels". In: *The Annals of Mathematical Statistics* 30.4 (Dec. 1959), pp. 1229–1241. DOI: 10.1214/aoms/1177706106.

[Bea13] Brian Beach. *How Long Do Disk Drives Last.* Nov. 2013. URL: https://www.backblaze.com/blog/how-long-do-disk-drives-last/ (visited on 06/22/2021).

[BF15] Victor Buttigieg and Noel Farrugia. "Improved Bit Error Rate Performance of Convolutional Codes with Synchronization Errors". In: *Proc. Int. Conf. Comm.* London, June 2015, pp. 4077–4082. DOI: 10.1109/ICC.2015.7248962.

[BGH17] Boris Bukh, Venkatesan Guruswami, and Johan Hastad. "An Improved Bound on the Fraction of Correctable Deletions". In: *IEEE Transactions on Information Theory* 63.1 (Jan. 2017), pp. 93–103. DOI: 10.1109/TIT.2016.2621044.

[BGZ18] Joshua Brakensiek, Venkatesan Guruswami, and Samuel Zbarsky. "Efficient Low-Redundancy Codes for Correcting Multiple Deletions". In: *IEEE Transactions on Information Theory* 64.5 (May 2018), pp. 3403–3410. DOI: 10.1109/TIT.2017.2746566.

[BJP10] Georg Böcherer, Valdemar Cardoso da Rocha Junior, and Cecilio Pimentel. "Capacity of General Discrete Noiseless Channels". In: *arXiv:0802.2451 [cs, math]* (June 2010). arXiv: 0802.2451 [cs, math].

[Bla+16] Meinolf Blawat, Klaus Gaedke, Ingo Hütter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W. Pruitt, and George M. Church. "Forward Error Correction for DNA Data Storage". In: *Procedia Computer Science* 80 (2016), pp. 1011–1022. DOI: 10.1016/j.procs.2016.05.398.

[BLS20] Joshua Brakensiek, Ray Li, and Bruce Spang. "Coded Trace Reconstruction in a Constant Number of Traces". In: *Proc. Annu. Symp. Found. Comput. Sci.* Nov. 2020, pp. 482–493. DOI: 10.1109/FOCS46700.2020.00052.

[Böc+10] G. Böcherer, R. Mathar, V.C. da Rocha, and C. Pimentel. "On the Capacity of Constrained Systems". In: *Proc. Int. ITG Conf. Source Channel Coding.* Jan. 2010, pp. 1–6.

[Bor+16] James Bornholt, Randolph Lopez, Douglas M. Carmean, Luis Ceze, Georg Seelig, and Karin Strauss. "A DNA-Based Archival Storage System". In: *Proc. Int. Conf. Architectural Support Program. Lang. Operating Syst.* Atlanta, Georgia, USA, 2016, pp. 637–649. DOI: 10.1145/2872362.2872397.

[Bos14] Martin Bossert. *Kanalcodierung.* Wiesbaden: Vieweg & Teubner, 2014. DOI: 10.1524/9783486755169.

[Bro+90] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith. "A New Table of Constant Weight Codes". In: *IEEE Transactions on Information Theory* 36.6 (Nov. 1990), pp. 1334–1380. DOI: 10.1109/18.59932.

[BSW10] J. A. Briffa, H. G. Schaathun, and S. Wesemeyer. "An Improved Decoding Algorithm for the Davey-MacKay Construction". In: *Proc. Int. Conf. Comm.* Cape Town, South Africa, May 2010. DOI: 10.1109/ICC.2010.5502293.

[Cai+21] Kui Cai, Yeow Meng Chee, Ryan Gabrys, Han Mao Kiah, and Tuan Thanh Nguyen. "Correcting a Single Indel/Edit for DNA-Based Data Storage: Linear-Time Encoders and Order-Optimality". In: *IEEE Transactions on Information Theory* 67.6 (June 2021), pp. 3438–3451. DOI: 10.1109/TIT.2021.3049627.

[Car13] Marvin H. Caruthers. "The Chemical Synthesis of DNA/RNA: Our Gift to Science". In: *Journal of Biological Chemistry* 288.2 (Jan. 2013), pp. 1420–1427. DOI: 10.1074/jbc.X112.442855.

[Car+19] Douglas Carmean, Luis Ceze, Georg Seelig, Kendall Stewart, Karin Strauss, and Max Willsey. "DNA Data Storage and Hybrid Molecular–Electronic Computing". In: *Proceedings of the IEEE* 107.1 (Jan. 2019), pp. 63–72. DOI: 10.1109/JPROC.2018.2875386.

[CGK12] G. M. Church, Y. Gao, and S. Kosuri. "Next-Generation Digital Information Storage in DNA". In: *Science* 337.6102 (Sept. 2012), pp. 1628–1628. DOI: 10.1126/science.1226355.

[Cha+19] Shubham Chandak, Kedar Tatwawadi, Billy Lau, Jay Mardia, Matthew Kubit, Joachim Neu, Peter Griffin, Mary Wootters, Tsachy Weissman, and Hanlee Ji. "Improved Read/Write Cost Tradeoff in DNA-Based Data Storage Using LDPC Codes". In: *Proc. Annu. Allerton Conf. Commun. Control Comp.* Monticello, IL, Sept. 2019, pp. 147–156. DOI: 10.1109/ALLERTON.2019.8919890.

[Cha+20] Shubham Chandak, Joachim Neu, Kedar Tatwawadi, Jay Mardia, Billy Lau, Matthew Kubit, Reyna Hulett, Peter Griffin, Mary Wootters, Tsachy Weissman, and Hanlee Ji. "Overcoming High Nanopore Basecaller Error Rates for DNA Storage via Basecaller-Decoder Integration and Convolutional Codes". In: *Proc. Int. Conf. Acoust., Speech, Sig. Process.* Barcelona, Spain, May 2020, pp. 8822–8826. DOI: 10.1109/ICASSP40776.2020.9053441.

[Che+14] Ling Cheng, Theo G. Swart, Hendrik C. Ferreira, and Khaled A. S. Abdel-Ghaffar. "Codes for Correcting Three or More Adjacent Deletions or Insertions". In: *Proc. Int. Symp. Inf. Theory*. Honolulu, HI, USA, June 2014, pp. 1246–1250. DOI: 10.1109/ISIT.2014.6875032.

[Che+18] Yeow Meng Chee, Johan Chrisnata, Han Mao Kiah, and Tuan Thanh Nguyen. "Efficient Encoding/Decoding of Irreducible Words for Codes Correcting Tandem Duplications". In: *Proc. Int. Symp. Inf. Theory*. Vail, CO, USA, June 2018, pp. 2406–2410. DOI: 10.1109/ISIT.2018.8437789.

[Che+20] Mahdi Cheraghchi, Ryan Gabrys, Olgica Milenkovic, and João Ribeiro. "Coded Trace Reconstruction". In: *IEEE Transactions on Information Theory* 66.10 (Oct. 2020), pp. 6084–6103. DOI: 10.1109/TIT.2020.2996377.

[Che+21] Kuan Cheng, Venkatesan Guruswami, Bernhard Haeupler, and Xin Li. "Efficient Linear and Affine Codes for Correcting Insertions/Deletions". In: *Proc. Annu. Symp. Discrete Algorithms*. Philadelphia, PA, Jan. 2021. DOI: 10.1137/1.9781611976465.

[Cho+20] Yeongjae Choi, Hyung Jong Bae, Amos C. Lee, Hansol Choi, Daewon Lee, Taehoon Ryu, Jinwoo Hyun, Seojoo Kim, Hyeli Kim, Suk-Heung Song, Kibeom Kim, Wook Park, and Sunghoon Kwon. "DNA Micro-Disks for the Management of DNA-Based Data Storage with Index and Write-Once–Read-Many (WORM) Memory Features". In: *Advanced Materials* 32.37 (2020), p. 2001249. DOI: 10.1002/adma.202001249.

[CK11] Imre Csiszár and János Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Second. Cambridge: Cambridge University Press, 2011. DOI: 10.1017/CBO9780511921889.

[CNS19] Luis Ceze, Jeff Nivala, and Karin Strauss. "Molecular Digital Data Storage Using DNA". In: *Nature Reviews Genetics* 20.8 (Aug. 2019), pp. 456–466. DOI: 10.1038/s41576-019-0125-3.

[Coh78] Joel E. Cohen. "Derivatives of the Spectral Radius as a Function of Non-Negative Matrix Elements". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 83.2 (Mar. 1978), pp. 183–190. DOI: 10.1017/S0305004100054438.

[Coh81] Joel E. Cohen. "Convexity of the Dominant Eigenvalue of an Essentially Nonnegative Matrix". In: *Proceedings of the American Mathematical Society* 81.4 (Apr. 1981), pp. 657–657. DOI: 10.1090/S0002-9939-1981-0601750-2.

[Cou19] Thomas Coughlin. "Nine Years of Media and Entertainment Digital Storage Surveys". In: *SMPTE Motion Imaging Journal* 128.7 (Aug. 2019), pp. 1–10. DOI: 10.5594/JMI.2019.2918035.

[CRB99] Catherine Taylor Clelland, Viviana Risca, and Carter Bancroft. "Hiding Messages in DNA Microdots". In: *Nature* 399.6736 (June 1999), pp. 533–534. DOI: 10.1038/21092.

[CS99] G. Caire and S. Shamai. "On the Capacity of Some Channels with Channel State Information". In: *IEEE Transactions on Information Theory* 45.6 (Sept. 1999), pp. 2007–2019. DOI: 10.1109/18.782125.

[CST21] Roni Con, Amir Shpilka, and Itzhak Tamo. "Linear and Reed Solomon Codes against Adversarial Insertions and Deletions". In: *arXiv:2107.05699 [cs, math]* (July 2021). arXiv: 2107.05699 [cs, math].

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Second. Hoboken, NJ: Wiley-Interscience, 2006. DOI: 10.1002/047174882X.

[DA10] Lara Dolecek and Venkat Anantharam. "Repetition Error Correcting Sets: Explicit Constructions and Prefixing Methods". In: *SIAM Journal on Discrete Mathematics* 23.4 (Jan. 2010), pp. 2120–2146. DOI: 10.1137/080730093.

[Dav96] Joe Davis. "Microvenus". In: *Art Journal* 55.1 (1996), pp. 70–74. DOI: 10.2307/777811.

[DM01] Matthew C. Davey and David J. C. Mackay. "Reliable Communication over Channels with Insertions, Deletions, and Substitutions". In: *IEEE Transactions on Information Theory* 47.2 (Feb. 2001), pp. 687–698. DOI: 10.1109/18.910582.

[Dob67] R. L. Dobrushin. “Shannon’s Theorems for Channels with Synchronization Errors”. In: *Problems of Information Transmission* 3.4 (1967), pp. 11–26.

[EZ16] Yaniv Erlich and Dina Zielinski. *Capacity-Approaching DNA Storage*. Preprint. Synthetic Biology, Sept. 2016. DOI: 10.1101/074237.

[EZ17] Yaniv Erlich and Dina Zielinski. “DNA Fountain Enables a Robust and Efficient Storage Architecture”. In: *Science (New York, N.Y.)* 355.6328 (Mar. 2017), pp. 950–954. DOI: 10.1126/science.aaj2038.

[Fey59] Richard Feynman. *There’s Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics*. Pasadena, CA, Dec. 1959.

[FH61] Robert M. Fano and David Hawkins. “Transmission of Information: A Statistical Theory of Communications”. In: *American Journal of Physics* 29.11 (Nov. 1961), pp. 793–794. DOI: 10.1119/1.1937609.

[FL18] Fahim Farzadfard and Timothy K. Lu. “Emerging Applications for DNA Writers and Molecular Recorders”. In: *Science* 361.6405 (Aug. 2018), pp. 870–875. DOI: 10.1126/science.aat9249.

[Fro12] Georg Frobenius. *Über Matrizen aus nicht negativen Elementen*. Königliche Gesellschaft der Wissenschaften, 1912. DOI: 10.3931/E-RARA-18865.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge ; New York: Cambridge University Press, 2009.

[FVY15] Arman Fazeli, Alexander Vardy, and Eitan Yaakobi. “Generalized Sphere Packing Bound”. In: *IEEE Transactions on Information Theory* 61.5 (May 2015), pp. 2313–2334. DOI: 10.1109/TIT.2015.2413418.

[Gab+20] R. Gabrys, H. S. Dau, C. J. Colbourn, and O. Milenkovic. “Set-Codes with Small Intersections and Small Discrepancies”. In: *SIAM Journal on Discrete Mathematics* 34.2 (Jan. 2020), pp. 1148–1171. DOI: 10.1137/19M1241106.

[Gal61] Robert G. Gallager. *Sequential Decoding for Binary Channel with Noise and Synchronization Errors*. Tech. rep. Arlington, VA, USA: Lincoln Lab Group, Sept. 1961.

[Gal72] Robert Gallager. *Information Theory and Reliable Communication*. Vienna: Springer Vienna, 1972. DOI: 10.1007/978-3-7091-2945-6.

[GF93] J. Gu and T. Fuja. “A Generalized Gilbert-Varshamov Bound Derived via Analysis of a Code-Search Algorithm”. In: *IEEE Transactions on Information Theory* 39.3 (May 1993), pp. 1089–1093. DOI: 10.1109/18.256522.

[GHS20] Venkatesan Guruswami, Bernhard Haeupler, and Amirbehshad Shahrasbi. “Optimally Resilient Codes for List-Decoding from Insertions and Deletions”. In: *Proc. Annu. Symp. Theory Comput.* New York, NY, USA, June 2020, pp. 524–537. DOI: 10.1145/3357713.3384262.

[Gib+10] Daniel G. Gibson, John I. Glass, Carole Lartigue, Vladimir N. Noskov, Ray-Yuan Chuang, Mikkel A. Algire, Gwynedd A. Benders, Michael G. Montague, Li Ma, Monzia M. Moodie, Chuck Merryman, Sanjay Vashee, Radha Krishnakumar, Nacyra Assad-Garcia, Cynthia Andrews-Pfannkoch, Evgeniya A. Denisova, Lei Young, Zhi-Qing Qi, Thomas H. Segall-Shapiro, Christopher H. Calvey, Prashanth P. Parmar, Clyde A. Hutchison, Hamilton O. Smith, and J. Craig Venter. "Creation of a Bacterial Cell Controlled by a Chemically Synthesized Genome". In: *Science* 329.5987 (July 2010), pp. 52–56. DOI: 10.1126/science.1190719.

[Gil52] E. N. Gilbert. "A Comparison of Signalling Alphabets". In: *Bell System Technical Journal* 31.3 (May 1952), pp. 504–522. DOI: 10.1002/j.1538-7305.1952.tb01393.x.

[Gol+13] Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M. LeProust, Botond Sipos, and Ewan Birney. "Towards Practical, High-Capacity, Low-Maintenance Information Storage in Synthesized DNA". In: *Nature* 494.7435 (Feb. 2013), pp. 77–80. DOI: 10.1038/nature11875.

[GP80] S. I. Gelfand and M. S. Pinsker. "Coding for Channels with Random Parameters". In: *Problem of Control and Information Theory* 9.1 (1980), pp. 19–31.

[GPM20] Ryan Gabrys, Srilakshmi Pattabiraman, and Olgica Milenkovic. "Mass Error-Correction Codes for Polymer-Based Data Storage". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 25–30. DOI: 10.1109/ISIT44484.2020.9174404.

[GPM21] Ryan Gabrys, Srilakshmi Pattabiraman, and Olgica Milenkovic. "Reconstructing Mixtures of Coded Strings from Prefix and Suffix Compositions". In: *Proc. Inf. Theory Workshop*. Riva del Garda, Italy, Apr. 2021, pp. 1–5. DOI: 10.1109/ITW46852.2021.9457660.

[Gra+15] Robert N. Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J. Stark. "Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes". In: *Angewandte Chemie International Edition* 54.8 (Feb. 2015), pp. 2552–2555. DOI: 10.1002/anie.201411378.

[GS19] Ryan Gabrys and Frederic Sala. "Codes Correcting Two Deletions". In: *IEEE Transactions on Information Theory* 65.2 (Feb. 2019), pp. 965–974. DOI: 10.1109/TIT.2018.2876281.

[Gus09] Claes Gustafsson. "For Anyone Who Ever Said There's No Such Thing as a Poetic Gene". In: *Nature* 458.7239 (Apr. 2009), pp. 703–703. DOI: 10.1038/458703a.

[GW17] Venkatesan Guruswami and Carol Wang. "Deletion Codes in the High-Noise and High-Rate Regimes". In: *IEEE Transactions on Information Theory* 63.4 (Apr. 2017), pp. 1961–1970. DOI: 10.1109/TIT.2017.2659765.

[GY18] Ryan Gabrys and Eitan Yaakobi. "Sequence Reconstruction over the Deletion Channel". In: *IEEE Transactions on Information Theory* 64.4 (Apr. 2018), pp. 2924–2931. DOI: 10.1109/TIT.2018.2800044.

[GYM18] Ryan Gabrys, Eitan Yaakobi, and Olgica Milenkovic. "Codes in the Damerau Distance for Deletion and Adjacent Transposition Correction". In: *IEEE Transactions on Information Theory* 64.4 (Apr. 2018), pp. 2550–2570. DOI: 10.1109/TIT.2017.2778143.

[HA21] Thomas Heinis and Jamie J. Alnasir. “Survey of Information Encoding Techniques for DNA”. In: *arXiv:1906.11062 [cs, math, q-bio]* (Aug. 2021). arXiv: 1906.11062 [cs, math, q-bio].

[Hae19] Bernhard Haeupler. “Optimal Document Exchange and New Codes for Insertions and Deletions”. In: *Proc. Annu. Symp. Found. Comput. Sci.* Baltimore, MD, USA, Nov. 2019, pp. 334–347. DOI: 10.1109/FOCS.2019.00029.

[Ham50] R. W. Hamming. “Error Detecting and Error Correcting Codes”. In: *The Bell System Technical Journal* 29.2 (Apr. 1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[Hao+21] Yaya Hao, Qian Li, Chunhai Fan, and Fei Wang. “Data Storage Based on DNA”. In: *Small Structures* 2.2 (2021), p. 2000046. DOI: 10.1002/sstr.202000046.

[HB21] Serge Kas Hanna and Rawad Bitar. “Detecting Deletions and Insertions in Concatenated Strings with Optimal Redundancy”. In: *arXiv:2105.00212 [cs, math]* (May 2021). arXiv: 2105.00212 [cs, math].

[Hec+17] Reinhard Heckel, Ilan Shomorony, Kannan Ramchandran, and David N. C. Tse. “Fundamental Limits of DNA Storage Systems”. In: *Proc. Int. Symp. Inf. Theory*. Aachen, June 2017, pp. 3130–3134. DOI: 10.1109/ISIT.2017.8007106.

[Hec18] Reinhard Heckel. “An Archive Written in DNA”. In: *Nature Biotechnology* 36.3 (Mar. 2018), pp. 236–237. DOI: 10.1038/nbt.4093.

[HF02] A.S. J. Helberg and Hendrik C. Ferreira. “On Multiple Insertion/Deletion Correcting Codes”. In: *IEEE Transactions on Information Theory* 48.1 (Jan. 2002), pp. 305–308. DOI: 10.1109/18.971760.

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. 2nd ed. Cambridge ; New York: Cambridge University Press, 2012.

[HMG19] Reinhard Heckel, Gediminas Mikutis, and Robert N. Grass. “A Characterization of the DNA Data Storage Channel”. In: *Scientific Reports* 9.9663 (July 2019). DOI: 10.1038/s41598-019-45832-6.

[Hoe63] Wassily Hoeffding. “Probability Inequalities for Sums of Bounded Random Variables”. In: *Journal of the American Statistical Association* 58.301 (Mar. 1963), pp. 13–30. DOI: 10.1080/01621459.1963.10500830.

[Hof+13] Eran Hof, Igal Sason, Shlomo Shamai, and Chao Tian. “Capacity-Achieving Polar Codes for Arbitrarily Permuted Parallel Channels”. In: *IEEE Transactions on Information Theory* 59.3 (Mar. 2013), pp. 1505–1516. DOI: 10.1109/TIT.2012.2236971.

[HR00] Daniel S. Hirschberg and Mireille Regnier. “Tight Bounds on the Number of String Subsequences”. In: *Journal of Discrete Algorithms* 1.1 (2000), pp. 123–132.

[HSS18] Bernhard Haeupler, Amirbehshad Shahrasbi, and Madhu Sudan. “Synchronization Strings: List Decoding for Insertions and Deletions”. In: *Proc. Int. Colloq. Automata, Languages, and Programming*. Dagstuhl, Germany, 2018, 76:1–76:14. DOI: 10.4230/LIPICS.ICALP.2018.76.

[HY20] Tomohiro Hayashi and Kenji Yasunaga. “On the List Decodability of Insertions and Deletions”. In: *IEEE Transactions on Information Theory* 66.9 (Sept. 2020), pp. 5335–5343. DOI: 10.1109/TIT.2020.2981321.

[IR90] Kenneth Ireland and Michael Rosen. *A Classical Introduction to Modern Number Theory*. Vol. 84. Graduate Texts in Mathematics. New York, NY: Springer New York, 1990. DOI: 10.1007/978-1-4757-2103-4.

[Jai+17a] Siddharth Jain, Farzad Farnoud Hassanzadeh, Moshe Schwartz, and Jehoshua Bruck. "Duplication-Correcting Codes for Data Storage in the DNA of Living Organisms". In: *IEEE Transactions on Information Theory* 63.8 (Aug. 2017), pp. 4996–5010. DOI: 10.1109/TIT.2017.2688361.

[Jai+17b] Siddharth Jain, Farzad Farnoud Hassanzadeh, Moshe Schwartz, and Jehoshua Bruck. "Noise and Uncertainty in String-Duplication Systems". In: *Proc. Int. Symp. Inf. Theory*. Aachen, Germany, June 2017, pp. 3120–3124. DOI: 10.1109/ISIT.2017.8007104.

[Jai+20] Siddharth Jain, Farzad Farnoud, Moshe Schwartz, and Jehoshua Bruck. "Coding for Optimized Writing Rate in DNA Storage". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 711–716. DOI: 10.1109/ISIT44484.2020.9174253.

[Jen06] Johan L. W. V. Jensen. "Sur Les Fonctions Convexes et Les Inégalités Entre Les Valeurs Moyennes". In: *Acta Mathematica* 30.0 (1906), pp. 175–193. DOI: 10.1007/BF02418571.

[JH84] J. Justesen and T. Hoholdt. "Maxentropic Markov Chains". In: *IEEE Transactions on Information Theory* 30.4 (July 1984), pp. 665–667. DOI: 10.1109/TIT.1984.1056939.

[JLR00] Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random Graphs*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Feb. 2000. DOI: 10.1002/9781118032718.

[Joh72] Sehner Johnson. "Upper Bounds for Constant Weight Error Correcting Codes". In: *Discrete Mathematics* 3.1-3 (1972), pp. 109–124. DOI: 10.1016/0012-365X(72)90027-1.

[KC14] Sriram Kosuri and George M Church. "Large-Scale de Novo DNA Synthesis: Technologies and Applications". In: *Nature Methods* 11.5 (May 2014), pp. 499–507. DOI: 10.1038/nmeth.2918.

[Kin61] J. F. C. Kingman. "A Convexity Property of Positive Matrices". In: *The Quarterly Journal of Mathematics* 12.1 (1961), pp. 283–284. DOI: 10.1093/qmath/12.1.283.

[KK13] Ankur A. Kulkarni and Negar Kiyavash. "Nonasymptotic Upper Bounds for Deletion Correcting Codes". In: *IEEE Transactions on Information Theory* 59.8 (Aug. 2013), pp. 5115–5130. DOI: 10.1109/TIT.2013.2257917.

[KK19] Hikari Koremura and Haruhiko Kaneko. "Successive Cancellation Decoding of Polar Codes for Insertion/Deletion Error Correction". In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 1357–1361. DOI: 10.1109/ISIT.2019.8849482.

[KM05] S. Kannan and A. McGregor. "More on Reconstructing Strings from Random Traces: Insertions and Deletions". In: *Proc. Int. Symp. Inf. Theory*. Adelaide, Australia, 2005, pp. 297–301. DOI: 10.1109/ISIT.2005.1523342.

[KMC17] Reza Kalhor, Prashant Mali, and George M Church. "Rapidly Evolving Homing CRISPR Barcodes". In: *Nature Methods* 14.2 (Feb. 2017), pp. 195–200. DOI: 10.1038/nmeth.4108.

[KMR00] A. Khandekar, Robert J. McEliece, and Eugene R. Rodemich. "The Discrete Noiseless Channel Revisited". In: *Coding, Communications, and Broadcasting*. Badlock: Research Studies Series Ltd., UK, 2000, pp. 115–137.

[Kno08] Andreas Knoblauch. "Closed-Form Expressions for the Moments of the Binomial Probability Distribution". In: *SIAM Journal on Applied Mathematics* 69.1 (Jan. 2008), pp. 197–204. DOI: 10.1137/070700024.

[Kov19a] Mladen Kovačević. "Runlength-Limited Sequences and Shift-Correcting Codes: Asymptotic Analysis". In: *IEEE Transactions on Information Theory* 65.8 (Aug. 2019), pp. 4804–4814. DOI: 10.1109/TIT.2019.2907979.

[Kov19b] Mladen Kovačević. "Zero-Error Capacity of Duplication Channels". In: *IEEE Transactions on Communications* 67.10 (Oct. 2019), pp. 6735–6742. DOI: 10.1109/TCOMM.2019.2931342. arXiv: 1902.06275.

[KPM16] Han Mao Kiah, Gregory J. Puleo, and Olgica Milenkovic. "Codes for DNA Sequence Profiles". In: *IEEE Transactions on Information Theory* 62.6 (June 2016), pp. 3125–3146. DOI: 10.1109/TIT.2016.2555321.

[KSC78] V. F. Kolchin, B. A. Sevast'yanov, and V. P. Chistyakov. *Random Allocations*. Scripta Series in Mathematics. V. H. Winston, 1978.

[KT18a] Mladen Kovacevic and Vincent Y. F. Tan. "Asymptotically Optimal Codes Correcting Fixed-Length Duplication Errors in DNA Storage Systems". In: *IEEE Communications Letters* 22.11 (Nov. 2018), pp. 2194–2197. DOI: 10.1109/LCOMM.2018.2868666.

[KT18b] Mladen Kovačević and Vincent Y. F. Tan. "Codes in the Space of Multisets—Coding for Permutation Channels with Impairments". In: *IEEE Transactions on Information Theory* 64.7 (July 2018), pp. 5156–5169. DOI: 10.1109/TIT.2017.2789292. arXiv: 1612.08837.

[LC04] Shu Lin and Daniel J. Costello. *Error Control Coding: Fundamentals and Applications*. 2. ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2004. DOI: 10.1002/sat.4600020214.

[Lev01] Vladimir I. Levenshtein. "Efficient Reconstruction of Sequences from Their Subsequences or Supersequences". In: *Journal of Combinatorial Theory, Series A* 93.2 (Feb. 2001), pp. 310–332. DOI: 10.1006/jcta.2000.3081.

[Lev65] Vladimir I. Levenshtein. "Binary Codes Capable of Correcting Spurious Insertions and Deletions of Ones". In: *Problemy Peredachi Informatsii* 1.1 (1965), pp. 8–17.

[Lev66] Vladimir I. Levenshtein. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals". In: *Soviet Physics Doklady* 10.8 (Feb. 1966), pp. 707–7710.

[Lev67] Vladimir I. Levenshtein. "Asymptotically Optimum Binary Code with Correcting for Losses of One or Two Adjacent Bits". In: *Problemy Kibernetiki* 19 (1967), pp. 293–298.

[Lev74] Vladimir I. Levenshtein. "Elements of coding theory". In: *Discrete Mathematics and Math. Problems of Cybernetics*. Moscow: Nauka, 1974, pp. 207–305. DOI: 10.1007/978-3-0348-5543-3.

[Lin99] J. H. van Lint. *Introduction to Coding Theory*. 1999. DOI: 10.1007/978-3-642-58575-3.

[Liu20] Yi Liu. "Coding Techniques to Extend the Lifetime of Flash Memories". PhD thesis. San Diego: University of California, San Diego, 2020.

[Liu+20] Yi Liu, Pengfei Huang, Alexander W. Bergman, and Paul H. Siegel. "Rate-Constrained Shaping Codes for Structured Sources". In: *arXiv:2001.02748 [cs, math]* (Jan. 2020). arXiv: 2001.02748 [cs, math].

[LK21] Xiaozhou Lu and Sunghwan Kim. "Design of Nonbinary Error Correction Codes With a Maximum Run-Length Constraint to Correct a Single Insertion or Deletion Error for DNA Storage". In: *IEEE Access* 9 (2021), pp. 135354–135363. DOI: 10.1109/ACCESS.2021.3116245.

[LL17] Tero Laihonen and Tuomo Lehtila. "Improved Codes for List Decoding in the Levenshtein's Channel and Information Retrieval". In: *Proc. Int. Symp. Inf. Theory*. Aachen, Germany, June 2017, pp. 2643–2647. DOI: 10.1109/ISIT.2017.8007008.

[LM10] Zhenming Liu and Michael Mitzenmacher. "Codes for Deletion and Insertion Channels with Segmented Errors". In: *IEEE Transactions on Information Theory* 56.1 (Jan. 2010), pp. 224–232. DOI: 10.1109/TIT.2009.2034886.

[LM95] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. First. Cambridge University Press, Nov. 1995. DOI: 10.1017/CBO9780511626302.

[Lop+19] Randolph Lopez, Yuan-Jyue Chen, Siena Dumas Ang, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Georg Seelig, Karin Strauss, and Luis Ceze. "DNA Assembly for Nanopore Data Storage Readout". In: *Nature Communications* 10.1 (Dec. 2019), p. 2933. DOI: 10.1038/s41467-019-10978-4.

[LSY17] Michael Langberg, Moshe Schwartz, and Eitan Yaakobi. "Coding for the ℓ_∞-Limited Permutation Channel". In: *IEEE Transactions on Information Theory* 63.12 (Dec. 2017), pp. 7676–7686. DOI: 10.1109/TIT.2017.2762676.

[Mac15] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. 15th print. Cambridge: Cambridge University Press, 2015. DOI: 10.5555/971143.

[Mai78] David Maier. "The Complexity of Some Problems on Subsequences and Supersequences". In: *Journal of the ACM* 25.2 (Apr. 1978), pp. 322–336. DOI: 10.1145/322063.322075.

[Mak18] Anuran Makur. "Information Capacity of BSC and BEC Permutation Channels". In: *Proc. Annu. Allerton Conf. Commun. Control Comp.* Monticello, IL, USA, Oct. 2018, pp. 1112–1119. DOI: 10.1109/ALLERTON.2018.8636070.

[Mak+21] Konstantin Makarychev, Miklós Z. Rácz, Cyrus Rashtchian, and Sergey Yekhanin. "Batch Optimization for DNA Synthesis". In: *Proc. Int. Symp. Inf. Theory*. Melbourne, Australia, July 2021, pp. 1949–1954. DOI: 10.1109/ISIT45174.2021.9517820.

[Mar96] S. Martirosyan. "Single-Error Correcting Close-Packed and Perfect Codes". In: *Proc. 1st INTAS Int. Seminar Coding Theory Comb.* Thakhadzor, Armenia, 1996, pp. 90–115.

[Maz17] Kayvon Mazooji. "On Unique Decoding from Insertion Errors". In: *Proc. Int. Symp. Inf. Theory*. Aachen, Germany, June 2017, pp. 2698–2702. DOI: 10.1109/ISIT.2017.8007019.

[MB09] Hugues Mercier and Vijay K. Bhargava. “Convolutional Codes for Channels with Deletion Errors”. In: *Proc. Canadian Workshop Inf. Theory*. Ottawa, ON, Canada, May 2009, pp. 136–139. DOI: 10.1109/CWIT.2009.5069539.

[MBT13] Abolfazl S. Motahari, Guy Bresler, and David N. C. Tse. “Information Theory of DNA Shotgun Sequencing”. In: *IEEE Transactions on Information Theory* 59.10 (Oct. 2013), pp. 6273–6289. DOI: 10.1109/TIT.2013.2270273.

[Mel21] Stephen Melczer. *An Invitation to Analytic Combinatorics: From One to Several Variables*. Texts & Monographs in Symbolic Computation. Cham: Springer International Publishing, 2021. DOI: 10.1007/978-3-030-67080-1.

[Mey00] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics, 2000.

[Mil+18] Olgica Milenkovic, Ryan Gabrys, Han Mao Kiah, and S.M. Hossein Tabatabaei Yazdi. “Exabytes in a Test Tube”. In: *IEEE Spectrum* 55.5 (May 2018), pp. 40–45. DOI: 10.1109/MSPEC.2018.8352574.

[Mit06] Michael Mitzenmacher. “On the Theory and Practice of Data Recovery with Multiple Versions”. In: *Proc. Int. Symp. Inf. Theory*. Seattle, WA, July 2006, pp. 982–986. DOI: 10.1109/ISIT.2006.261874.

[Mit08] Michael Mitzenmacher. “Capacity Bounds for Sticky Channels”. In: *IEEE Transactions on Information Theory* 54.1 (Jan. 2008), pp. 72–77. DOI: 10.1109/TIT.2007.911291.

[Mit09] Michael Mitzenmacher. “A Survey of Results for Deletion Channels and Related Synchronization Channels”. In: *Probability Surveys* 6.none (Jan. 2009). DOI: 10.1214/08-PS141.

[Mit96] M. Mitzenmacher. “The Power of Two Choices in Randomized Load Balancing”. PhD thesis. University of California, Berkeley, 1996.

[MKB08] Hugues Mercier, Majid Khabbazian, and Vijay K. Bhargava. “On the Number of Subsequences When Deleting Symbols from a String”. In: *IEEE Transactions on Information Theory* 54.7 (July 2008), pp. 3279–3285. DOI: 10.1109/TIT.2008.924730.

[MPV14] Andrew McGregor, Eric Price, and Sofya Vorotnikova. “Trace Reconstruction Revisited”. In: *Algorithms - ESA 2014*. Vol. 8737. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 689–700. DOI: 10.1007/978-3-662-44777-2_57.

[MR83] Robert J. McEliece and Eugene R. Rodemich. “A Maximum Entropy Markov Chain”. In: *Proc. Conf. Inform. Sciences and Systems*. Johns Hopkins University, Mar. 1983, pp. 245–248.

[MRS01] B. H. Marcus, Ron M. Roth, and Paul H. Siegel. *An Introduction to Coding for Constrained Systems*. 2001.

[MSG15] David J. C. Mackay, Jossy Sayir, and Nick Goldman. *Near-Capacity Codes for Fountain Channels with Insertions, Deletions, and Substitutions, with Applications to DNA Archives*. Unpublished Manuscript. June 2015.

[MY20] Sagi Marcovich and Eitan Yaakobi. "Reconstruction of Strings from Their Substrings Spectrum". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 658–663. DOI: 10.1109/ISIT44484.2020.9174113.

[Nei65] M. S. Neiman. "On the Molecular Memory Systems and the Directed Mutations". In: *Radiotekhnika* 6 (1965), pp. 1–8.

[New+19] Sharon Newman, Ashley P. Stephenson, Max Willsey, Bichlien H. Nguyen, Christopher N. Takahashi, Karin Strauss, and Luis Ceze. "High Density DNA Data Storage Library via Dehydration with Digital Microfluidic Retrieval". In: *Nature Communications* 10.1 (Dec. 2019), p. 1706. DOI: 10.1038/s41467-019-09517-y.

[Ngu+21] Tuan Thanh Nguyen, Kui Cai, Kees A. Schouhamer Immink, and Han Mao Kiah. "Capacity-Approaching Constrained Codes With Error Correction for DNA-Based Data Storage". In: *IEEE Transactions on Information Theory* 67.8 (Aug. 2021), pp. 5602–5613. DOI: 10.1109/TIT.2021.3066430.

[NL06] Kang Ning and Hon Wai Leong. "Towards a Better Solution to the Shortest Common Supersequence Problem: The Deposition and Reduction Algorithm". In: *BMC Bioinformatics* 7.S4 (Dec. 2006), S12. DOI: 10.1186/1471-2105-7-S4-S12.

[Nus86] Roger D. Nussbaum. "Convexity and Log Convexity for the Spectral Radius". In: *Linear Algebra and its Applications* 73 (Jan. 1986), pp. 59–122. DOI: 10.1016/0024-3795(86)90233-8.

[Org+18] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, Christopher N Takahashi, Sharon Newman, Hsing-Yeh Parker, Cyrus Rashtchian, Kendall Stewart, Gagan Gupta, Robert Carlson, John Mulligan, Douglas Carmean, Georg Seelig, Luis Ceze, and Karin Strauss. "Random Access in Large-Scale DNA Data Storage". In: *Nature Biotechnology* 36.3 (Mar. 2018), pp. 242–248. DOI: 10.1038/nbt.4079.

[Pan+20] Chao Pan, S. M. Hossein Tabatabaei Yazdi, S Kasra Tabatabaei, Alvaro G. Hernandez, Charles Schroeder, and Olgica Milenkovic. "Image Processing in DNA". In: *Proc. Int. Conf. Acoust., Speech, Sig. Process.* May 2020, pp. 8831–8835. DOI: 10.1109/ICASSP40776.2020.9054262.

[Pan+21] Chao Pan, S. Kasra Tabatabaei, SM Hossein Tabatabaei Yazdi, Alvaro G. Hernandez, Charles M. Schroeder, and Olgica Milenkovic. "Rewritable Two-Dimensional DNA-Based Data Storage with Machine Learning Reconstruction". In: *bioRxiv* (Feb. 2021). DOI: 10.1101/2021.02.22.432304.

[Per07] Oskar Perron. "Zur Theorie der Matrices". In: *Mathematische Annalen* 64.2 (June 1907), pp. 248–263. DOI: 10.1007/BF01449896.

[PGM19] Srilakshmi Pattabiraman, Ryan Gabrys, and Olgica Milenkovic. "Reconstruction and Error-Correction Codes for Polymer-Based Data Storage". In: *Proc. Inf. Theory Workshop*. Visby, Sweden: IEEE, Aug. 2019, pp. 1–5. DOI: 10.1109/ITW44776.2019.8989171.

[PW04] Robin Pemantle and Mark C. Wilson. "Asymptotics of Multivariate Sequences II: Multiple Points of the Singular Variety". In: *Combinatorics, Probability and Computing* 13.4-5 (July 2004), pp. 735–761. DOI: 10.1017/S0963548304006248.

[PW08] Robin Pemantle and Mark C. Wilson. “Twenty Combinatorial Examples of Asymptotics Derived from Multivariate Generating Functions”. In: *SIAM Review* 50.2 (Jan. 2008), pp. 199–272. DOI: 10.1137/050643866.

[PW13] Robin Pemantle and Mark C. Wilson. *Analytic Combinatorics in Several Variables*. Cambridge Studies in Advanced Mathematics 140. Cambridge: Cambridge University Press, 2013.

[PYA21] Inbal Preuss, Zohar Yakhini, and Leon Anavy. *Data Storage Based on Combinatorial Synthesis of DNA Shortmers*. Preprint. Synthetic Biology, Aug. 2021. DOI: 10.1101/2021.08.01.454622.

[RA13] M. Ramezani and M. Ardakani. “On the Capacity of Duplication Channels”. In: *IEEE Transactions on Communications* 61.3 (Mar. 2013), pp. 1020–1027. DOI: 10.1109/TCOMM.2013.020413.120070.

[Ras+17] Cyrus Rashtchian, Konstantin Makarychev, Miklos Racz, Siena Ang, Djordje Jevdjic, Sergey Yekhanin, Luis Ceze, and Karin Strauss. “Clustering Billions of Reads for DNA Data Storage”. In: *Proc. Conf. Neural Inf. Process. Sys.* Long Beach, CA, Dec. 2017, p. 12.

[Rob55] Herbert Robbins. “A Remark on Stirling’s Formula”. In: *The American Mathematical Monthly* 62.1 (Jan. 1955), p. 26. DOI: 10.2307/2308012.

[Ros+13] Michael G. Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J. Lennon, Ryan Hegarty, Chad Nusbaum, and David B. Jaffe. “Characterizing and Measuring Bias in Sequence Data”. In: *Genome Biology* 14.5 (May 2013), R51. DOI: 10.1186/gb-2013-14-5-r51.

[Rot06] Ron Roth. *Introduction to Coding Theory*. Cambridge: Cambridge University Press, 2006. DOI: 10.1017/CBO9780511808968.

[Sab+20] Omer Sabary, Alexander Yucovich, Guy Shapira, and Eitan Yaakobi. “Reconstruction Algorithms for DNA-Storage Systems”. In: *Proc. Int. Conf. DNA Comput. Molecular Program.* Oxford, UK, Sept. 2020. DOI: 10.1101/2020.09.16.300186.

[Sal+17] Frederic Sala, Ryan Gabrys, Clayton Schoeny, and Lara Dolecek. “Exact Reconstruction from Insertions in Synchronization Codes”. In: *IEEE Transactions on Information Theory* 63.4 (Apr. 2017), pp. 2428–2445. DOI: 10.1109/TIT.2017.2649493.

[SB19] Jin Sima and Jehoshua Bruck. “Optimal K-Deletion Correcting Codes”. In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 847–851. DOI: 10.1109/ISIT.2019.8849750.

[SC64] B. A. Sevast’yanov and V. P. Chistyakov. “Asymptotic Normality in the Classical Ball Problem”. In: *Theory of Probability & Its Applications* 9.2 (Jan. 1964), pp. 198–211. DOI: 10.1137/1109034.

[Sch+17] Clayton Schoeny, Antonia Wachter-Zeh, Ryan Gabrys, and Eitan Yaakobi. “Codes Correcting a Burst of Deletions or Insertions”. In: *IEEE Transactions on Information Theory* 63.4 (Apr. 2017), pp. 1971–1985. DOI: 10.1109/TIT.2017.2661747.

[SCSI19] Wentu Song, Kui Cai, and Kees A. Schouhamer Immink. “Sequence-Subset Distance and Coding for Error Control for DNA-Based Data Storage”. In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 86–90. DOI: 10.1109/ISIT.2019.8849687.

[SH19] Ilan Shomorony and Reinhard Heckel. "Capacity Results for the Noisy Shuffling Channel". In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 762–766. DOI: 10.1109/ISIT.2019.8849789.

[SH21] Ilan Shomorony and Reinhard Heckel. "DNA-Based Storage: Models and Fundamental Limits". In: *IEEE Transactions on Information Theory* 67.6 (June 2021), pp. 3675–3689. DOI: 10.1109/TIT.2021.3058966.

[Sha48] Claude E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27 (Oct. 1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[Shi+17] Seth L. Shipman, Jeff Nivala, Jeffrey D. Macklis, and George M. Church. "CRISPR–Cas Encoding of a Digital Movie into the Genomes of a Population of Living Bacteria". In: *Nature* 547.7663 (July 2017), pp. 345–349. DOI: 10.1038/nature23017.

[SHS20] Seiyun Shin, Reinhard Heckel, and Ilan Shomorony. "Capacity of the Erasure Shuffling Channel". In: *Proc. Int. Conf. Acoust., Speech, Sig. Process.* May 2020, pp. 8841–8845. DOI: 10.1109/ICASSP40776.2020.9053486.

[SHY19] Ryo Shibata, Gou Hosoya, and Hiroyuki Yashima. "Design of Irregular LDPC Codes without Markers for Insertion/Deletion Channels". In: *Proc. Gobal Commun. Conf.* Waikoloa, HI, USA, Dec. 2019. DOI: 10.1109/GLOBECOM38437.2019.9014209.

[SIC18] Kees A. Schouhamer Immink and Kui Cai. "Design of Capacity-Approaching Constrained Codes for DNA-Based Storage Systems". In: *IEEE Communications Letters* 22.2 (Feb. 2018), pp. 224–227. DOI: 10.1109/LCOMM.2017.2775608.

[SIC19] Kees A. Schouhamer Immink and Kui Cai. "Efficient Balanced and Maximum Homopolymer-Run Restricted Block Codes for DNA-Based Data Storage". In: *IEEE Communications Letters* 23.10 (Oct. 2019), pp. 1676–1679. DOI: 10.1109/LCOMM.2019.2930970.

[Slo00] N. J. A. Sloane. "On Single-Deletion-Correcting Codes". In: *Codes and Designs: Proceedings of a Conference Honoring Professor Dijen K. Ray-Chaudhurion the Occasion of His 65th Birthday*. De Gruyter, 2000, pp. 273–292. DOI: 10.1515/9783110198119.273.

[Sma+20] Ilia Smagloy, Lorenz Welter, Antonia Wachter-Zeh, and Eitan Yaakobi. "Single-Deletion Single-Substitution Correcting Codes". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 775–780. DOI: 10.1109/ISIT44484.2020.9174213.

[Son+18] Wentu Song, Kui Cai, Mu Zhang, and Chau Yuen. "Codes with Run-Length and GC-Content Constraints for DNA-Based Data Storage". In: *IEEE Communications Letters* 22.10 (Oct. 2018), pp. 2004–2007. DOI: 10.1109/LCOMM.2018.2866566.

[Son+21] Wentu Song, Nikita Polyanskii, Kui Cai, and Xuan He. "On Multiple-Deletion Multiple-Substitution Correcting Codes". In: *2021 IEEE International Symposium on Information Theory (ISIT)*. Melbourne, Australia: IEEE, July 2021, pp. 2655–2660. DOI: 10.1109/ISIT45174.2021.9517878.

[Sor05] J. B. Soriaga. "On Near-Capacity Code Design for Partial-Response Channels". PhD thesis. University of California, San Diego, 2005.

[SRB20] Jin Sima, Netanel Raviv, and Jehoshua Bruck. “Robust Indexing - Optimal Codes for DNA Storage”. In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, June 2020, pp. 717–722. DOI: 10.1109/ISIT44484.2020.9174447.

[SRB21] Jin Sima, Netanel Raviv, and Jehoshua Bruck. “On Coding over Sliced Information”. In: *IEEE Transactions on Information Theory* 67.5 (May 2021), pp. 2793–2807. DOI: 10.1109/TIT.2021.3063709.

[Sri+19] Sundara R. Srinivasavaradhan, Michelle Du, Suhas Diggavi, and Christina Fragouli. “Symbolwise MAP for Multiple Deletion Channels”. In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 181–185. DOI: 10.1109/ISIT.2019.8849567.

[Sri+20] Sundara Rajan Srinivasavaradhan, Michelle Du, Suhas Diggavi, and Christina Fragouli. “Algorithms for Reconstruction over Single and Multiple Deletion Channels”. In: *arXiv:2005.14388 [cs, math]* (May 2020). arXiv: 2005.14388 [cs, math].

[SS06] J. B. Soriaga and Paul H. Siegel. “On the Design of Finite-State Shaping Encoders for Partial-Response Channels”. In: *Proc. Inf. Theory Appl. Workshop*. San Diego, CA, USA, Feb. 2006.

[SWB06] Slawomir Stanczak, Marcin Wiczanowski, and Holger Boche. *Resource Allocation in Wireless Networks*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. DOI: 10.1007/11818762.

[SY19] Maria Abu Sini and Eitan Yaakobi. “Reconstruction of Sequences in DNA Storage”. In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 290–294. DOI: 10.1109/ISIT.2019.8849740.

[SYY20] Omer Sabary, Eitan Yaakobi, and Alexander Yucovich. “The Error Probability of Maximum-Likelihood Decoding over Two Deletion/Insertion Channels”. In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, June 2020, pp. 763–768. DOI: 10.1109/ISIT44484.2020.9174488.

[Tab+20] S. Kasra Tabatabaei, Boya Wang, Nagendra Bala Murali Athreya, Behnam Enghiad, Alvaro Gonzalo Hernandez, Christopher J. Fields, Jean-Pierre Leburton, David Soloveichik, Huimin Zhao, and Olgica Milenkovic. “DNA Punch Cards for Storing Data on Native DNA Sequences via Enzymatic Nicking”. In: *Nature Communications* 11.1 (Dec. 2020), p. 1742. DOI: 10.1038/s41467-020-15588-z.

[Tab+21] S. Kasra Tabatabaei, Bach Pham, Chao Pan, Jingqian Liu, Shubham Chandak, Spencer A. Shorkey, Alvaro G. Hernandez, Aleksei Aksimentiev, Min Chen, Charles M. Schroeder, and Olgica Milenkovic. *Expanding the Molecular Alphabet of DNA-Based Data Storage Systems with Neural Network Nanopore Readout Processing*. Preprint. Synthetic Biology, Sept. 2021. DOI: 10.1101/2021.09.27.462049.

[Tak+19] Christopher N. Takahashi, Bichlien H. Nguyen, Karin Strauss, and Luis Ceze. “Demonstration of End-to-End Automation of DNA Data Storage”. In: *Scientific Reports* 9.1 (Mar. 2019), p. 4998. DOI: 10.1038/s41598-019-41228-8.

[Tal+19] Ido Tal, Henry D. Pfister, Arman Fazeli, and Alexander Vardy. “Polar Codes for the Deletion Channel: Weak and Strong Polarization”. In: *Proc. Int. Symp. Inf. Theory*. July 2019, pp. 1362–1366. DOI: 10.1109/ISIT.2019.8849705.

[Tan+19] Yuanyuan Tang, Yonatan Yehezkeally, Moshe Schwartz, and Farzad Farnoud. "Single-Error Detection and Correction for Duplication and Substitution Channels". In: *arXiv:1911.05413 [cs, math]* (Nov. 2019). arXiv: 1911.05413 [cs, math].

[TBK20] Shubham Taluja, Jagrit Bhupal, and Siva Rama Krishnan. "A Survey Paper on DNA-Based Data Storage". In: *Int. Conf. Emerging Trends Inf. Technol. Eng.* Feb. 2020, pp. 1–4. DOI: 10.1109/ic-ETITE47903.2020.62.

[Ten84] Grigory M. Tenengolts. "Nonbinary Codes, Correcting Single Deletion or Insertion". In: *IEEE Transactions on Information Theory* 30.5 (Sept. 1984), pp. 766–769. DOI: 10.1109/TIT.1984.1056962.

[TL18] Weixin Tang and David R. Liu. "Rewritable Multi-Event Analog Recording in Bacterial and Mammalian Cells". In: *Science* 360.6385 (Apr. 2018), eaap8992. DOI: 10.1126/science.aap8992.

[Tol97] Ludo M. G. M. Tolhuizen. "The Generalized Gilbert-Varshamov Bound Is Implied by Turan's Theorem". In: *IEEE Transactions on Information Theory* 43.5 (Sept. 1997), pp. 1605–1606. DOI: 10.1109/18.623158.

[Var57] R. Varshamov. "Estimate of the Number of Signals in Error Correcting Codes". In: *Dokl. Akad. Nauk SSSR* 117 (1957), pp. 739–741.

[VB95] John W. C. Van Bogart. *Magnetic Tape Storage and Handling: A Guide for Libraries and Archives*. Washington, DC : St. Paul, MN: Commission on Preservation and Access; National Media Laboratory, 1995.

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. First. Cambridge University Press, Sept. 2018. DOI: 10.1017/9781108231596.

[VT65] R. Varshamov and Grigory M. Tenengolts. "Code correcting single asymmetric errors". In: *Avtomatika i Telemekhanika* 26.2 (1965), pp. 288–292.

[Wac18] Antonia Wachter-Zeh. "List Decoding of Insertions and Deletions". In: *IEEE Transactions on Information Theory* 64.9 (Sept. 2018), pp. 6297–6304. DOI: 10.1109/TIT.2017.2777471.

[Wan+19] Yixin Wang, Md. Noor-A-Rahim, Erry Gunawan, Yong Liang Guan, and Chueh Loo Poh. "Construction of Bio-Constrained Code for DNA Data Storage". In: *IEEE Communications Letters* 23.6 (June 2019), pp. 963–966. DOI: 10.1109/LCOMM.2019.2912572.

[Wet] Kris A. Wetterstrand. *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)*. URL: http://www.genome.gov/sequencingcostsdata (visited on 08/04/2021).

[WG08] Frans M. J. Willems and Alexei Gorokhov. "Signaling over Arbitrarily Permuted Parallel Channels". In: *IEEE Transactions on Information Theory* 54.3 (Mar. 2008), pp. 1374–1382. DOI: 10.1109/TIT.2007.915912.

[Wie50] Helmut Wielandt. "Unzerlegbare, nicht negative Matrizen". In: *Mathematische Zeitschrift* 52.1 (Dec. 1950), pp. 642–648. DOI: 10.1007/BF02230720.

[Wil88] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Monographs on Numerical Analysis. Oxford : Oxford ; New York: Clarendon Press ; Oxford University Press, 1988.

[WM21] Nir Weinberger and Neri Merhav. "The DNA Storage Channel: Capacity and Error Probability". In: *arXiv:2109.12549 [cs, math]* (Sept. 2021). arXiv: 2109.12549 [cs, math].

[Wol06] Jack Keil Wolf. "An Introduction to Tensor Product Codes and Applications to Digital Storage Systems". In: *Proc. Inf. Theory Workshop*. Chengdu, China, Oct. 2006, pp. 6–10. DOI: 10.1109/ITW2.2006.323741.

[Wol59] J. Wolfowitz. "Simultaneous Channels". In: *Archive for Rational Mechanics and Analysis* 4 (Jan. 1959), pp. 371–386. DOI: 10.1007/BF00281397.

[WS21] Hengjia Wei and Moshe Schwartz. "Improved Coding over Sets for DNA-Based Data Storage". In: *IEEE Transactions on Information Theory*. Early Access (2021). DOI: 10.1109/TIT.2021.3119584.

[Xu+21] Chengtao Xu, Chao Zhao, Biao Ma, and Hong Liu. "Uncertainties in Synthetic DNA-Based Data Storage". In: *Nucleic Acids Research* 49.10 (June 2021), pp. 5451–5469. DOI: 10.1093/nar/gkab230.

[Yaz+15a] S. M. Hossein Tabatabaei Yazdi, Han Mao Kiah, Eva Garcia-Ruiz, Jian Ma, Huimin Zhao, and Olgica Milenkovic. "DNA-Based Storage: Trends and Methods". In: *IEEE Transactions on Molecular, Biological and Multi-Scale Communications* 1.3 (Sept. 2015), pp. 230–248. DOI: 10.1109/TMBMC.2016.2537305.

[Yaz+15b] S. M. Hossein Tabatabaei Yazdi, Yongbo Yuan, Jian Ma, Huimin Zhao, and Olgica Milenkovic. "A Rewritable, Random-Access DNA-Based Storage System". In: *Scientific Reports* 5.1 (Nov. 2015), p. 14138. DOI: 10.1038/srep14138.

[Yaz+18] Seyed M. T. Yazdi, Han Mao Kiah, Ryan Gabrys, and Olgica Milenkovic. "Mutually Uncorrelated Primers for DNA-Based Data Storage". In: *IEEE Transactions on Information Theory* 64.9 (Sept. 2018), pp. 6283–6296. DOI: 10.1109/TIT.2018.2792488.

[YGM17] S. M. Hossein Tabatabaei Yazdi, Ryan Gabrys, and Olgica Milenkovic. "Portable and Error-Free DNA-Based Data Storage". In: *Scientific Reports* 7.1 (Dec. 2017), p. 5011. DOI: 10.1038/s41598-017-05188-1.

[YS18] Yonatan Yehezkeally and Moshe Schwartz. "Reconstruction Codes for DNA Sequences with Uniform Tandem-Duplication Errors". In: *Proc. Int. Symp. Inf. Theory*. Vail, CO, USA, June 2018, pp. 2535–2539. DOI: 10.1109/ISIT.2018.8437731.

[YS21] Yonatan Yehezkeally and Moshe Schwartz. "Uncertainty of Reconstruction with List-Decoding from Uniform-Tandem-Duplication Noise". In: *IEEE Transactions on Information Theory* 67.7 (July 2021), pp. 4276–4287. DOI: 10.1109/TIT.2021.3070466.

Publications Containing Parts of This Thesis

[Len+18] Andreas Lenz, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "Coding over Sets for DNA Storage". In: *Proc. Int. Symp. Inf. Theory*. Vail, CO, USA, June 2018, pp. 2411–2415. DOI: 10.1109/ISIT.2018.8437544.

[Len+19a] Andreas Lenz, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "An Upper Bound on the Capacity of the DNA Storage Channel". In: *Proc. Inf. Theory Workshop*. Visby, Sweden, Aug. 2019, pp. 1–5. DOI: 10.1109/ITW44776.2019.8989388.

[Len+19b] Andreas Lenz, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "Anchor-Based Correction of Substitutions in Indexed Sets". In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 757–761. DOI: 10.1109/ISIT.2019.8849523.

[Len+20a] Andreas Lenz, Yi Liu, Cyrus Rashtchian, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "Coding for Efficient DNA Synthesis". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 2885–2890. DOI: 10.1109/ISIT44484.2020.9174272.

[Len+20c] Andreas Lenz, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "Achieving the Capacity of the DNA Storage Channel". In: *Proc. Int. Conf. Acoust., Speech, Sig. Process.* Barcelona, Spain, May 2020, pp. 8846–8850. DOI: 10.1109/ICASSP40776.2020.9053049.

[Len+20d] Andreas Lenz, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "Coding over Sets for DNA Storage". In: *IEEE Transactions on Information Theory* 66.4 (Apr. 2020), pp. 2331–2351. DOI: 10.1109/TIT.2019.2961265.

[Len+21f] Andreas Lenz, Paul H. Siegel, Antonia Wachter-Zeh, and Eitan Yaakobi. "On the Capacity of DNA-Based Data Storage under Substitution Errors". In: *Proc. Visual Commun. Image Process.* Munich, Germany, Dec. 2021.

Preprints Containing Parts of This Thesis

[Len+21d] Andreas Lenz, Stephen Melczer, Cyrus Rashtchian, and Paul H. Siegel. "Multivariate Analytic Combinatorics for Cost Constrained Channels and Subsequence Enumeration". In: *submitted to Proc. Annu. Symp. Theory Comput.* (2021). arXiv: 2111.06105.

Other Publications by the Author

[Imm+17] Vincent Immler, Matthias Hiller, Qinzhi Liu, Andreas Lenz, and Antonia Wachter-Zeh. "Variable-Length Bit Mapping and Error-Correcting Codes for Higher-Order Alphabet PUFs". In: *Security, Privacy, and Applied Cryptography Engineering*. Vol. 10662. Cham, Switzerland: Springer International Publishing, 2017, pp. 190–209. DOI: 10.1007/978-3-319-71501-8_11.

[Imm+19] Vincent Immler, Matthias Hiller, Qinzhi Liu, Andreas Lenz, and Antonia Wachter-Zeh. "Variable-Length Bit Mapping and Error-Correcting Codes for Higher-Order Alphabet PUFs". In: *Journal of Hardware and Systems Security* 3.1 (Mar. 2019), pp. 78–93. DOI: 10.1007/s41635-018-0056-z.

[Len+20b] Andreas Lenz, Cyrus Rashtchian, Paul H. Siegel, and Eitan Yaakobi. "Covering Codes for Insertions and Deletions". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 723–728. DOI: 10.1109/ISIT44484.2020.9174295.

[Len+21a] Andreas Lenz, Rawad Bitar, Antonia Wachter-Zeh, and Eitan Yaakobi. "Function-Correcting Codes". In: *Proc. Int. Symp. Inf. Theory*. Melbourne, Australia, July 2021, pp. 1290–1295. DOI: 10.1109/ISIT45174.2021.9517976.

[Len+21b] Andreas Lenz, Rawad Bitar, Antonia Wachter-Zeh, and Eitan Yaakobi. "Function-Correcting Codes". In: *submitted to IEEE Transaction on Information Theory* (2021). arXiv: 2102.03094.

[Len+21c] Andreas Lenz, Issam Maarouf, Lorenz Welter, Antonia Wachter-Zeh, Eirik Rosnes, and Alexandre Graell i Amat. "Concatenated Codes for Recovery From Multiple Reads of DNA Sequences". In: *Proc. Inf. Theory Workshop*. Riva del Garda, Italy, Apr. 2021, pp. 1–5. DOI: 10.1109/ITW46852.2021.9457675.

[Len+21e] Andreas Lenz, Cyrus Rashtchian, Paul H. Siegel, and Eitan Yaakobi. "Covering Codes Using Insertions or Deletions". In: *IEEE Transactions on Information Theory* 67.6 (June 2021), pp. 3376–3388. DOI: 10.1109/TIT.2020.2985691.

[LJW18] Andreas Lenz, Niklas Jünger, and Antonia Wachter-Zeh. "Bounds and Constructions for Multi-Symbol Duplication Error Correcting Codes". In: *Proc. Workshop Algebraic Comb. Coding Theory*. Svetlogorsk, Russia, Sept. 2018, pp. 129–133.

[LP20] Andreas Lenz and Nikita Polyanskii. "Optimal Codes Correcting a Burst of Deletions of Variable Length". In: *Proc. Int. Symp. Inf. Theory*. Los Angeles, CA, USA, June 2020, pp. 757–762. DOI: 10.1109/ISIT44484.2020.9174288.

[LWP20] Andreas Lenz, Lorenz Welter, and Sven Puchinger. "Achievable Rates of Concatenated Codes in DNA Storage under Substitution Errors". In: *Proc. Int. Symp. Inf. Theory Appl.* Kapolei, HI, USA, Oct. 2020, pp. 269–273.

[LWY17] Andreas Lenz, Antonia Wachter-Zeh, and Eitan Yaakobi. "Bounds on Codes Correcting Tandem and Palindromic Duplications". In: *Proc. Int. Workshop Coding Cryptography*. St. Petersburg, Russia, Sept. 2017, p. 11.

[LWY19] Andreas Lenz, Antonia Wachter-Zeh, and Eitan Yaakobi. "Duplication-Correcting Codes". In: *Designs, Codes and Cryptography* 87.2-3 (Mar. 2019), pp. 277–298. DOI: 10.1007/s10623-018-0523-0.

[Shi+19] Tal Shinkar, Eitan Yaakobi, Andreas Lenz, and Antonia Wachter-Zeh. "Clustering-Correcting Codes". In: *Proc. Int. Symp. Inf. Theory*. Paris, France, July 2019, pp. 81–85. DOI: 10.1109/ISIT.2019.8849737.

[Shi+22] Tal Shinkar, Eitan Yaakobi, Andreas Lenz, and Antonia Wachter-Zeh. "Clustering-Correcting Codes". In: *IEEE Transactions on Information Theory* 68.3 (Mar. 2022), pp. 1560–1580. DOI: 10.1109/TIT.2021.3127174.