

https://dspace.org/

Sequence alignment using the seeding heuristics method

Oyekanmi, Adedoyin

2022
https://hdl.handle.net/10539/37026

Downloaded from a DSpace repository. https://dspace.org/

Master's Dissertation

Sequence alignment using the seeding heuristics method

Author:

Adedoyin Oyekanmi

(2260391)

Supervisor:

Prof. Margaret Archibald

submitted to

the Faculty of Science, in fulfilment of the requirements for a

Master's degree

in the

School of Mathematics



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

Declaration of Authorship

I, Adedoyin OYEKANMI (2260391), declare that this Master's Dissertation titled, "Sequence alignment using the seeding heuristics method" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this Master's Dissertation has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this Master's Dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the Master's Dissertation is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 15/08/2022

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Abstract

Faculty of Science

School of Mathematics

Masters

Sequence alignment using the seeding heuristics method

by Adedoyin OYEKANMI (2260391)

In this dissertation, we studied Guillaume J. Filion's paper on utilizing analytic combinatorics to calculate the probability that a read of size $(s = k)$ has no seed $(P(S))$. First, we computed the weighted generating functions of all reads and reads that have no seed by constructing sequences of combinatorial objects using transfer graphs and transfer matrices as formulated by Filion. Then, we extended his logic to calculate the probability that a seed is present in a read $(1 - P(S))$ by singularity analysis, after obtaining the probability that a read has no seed. This involved extracting coefficients of weighted generating functions and solving complex polynomials which is the analytic combinatorics approach. Finally, we use recurrence approach (on Mathematica) to find more precise probabilities than what was provided by Filion for some practical examples.

Acknowledgements

First and foremost, thanks to God, the Almighty for his guidance and wisdom as I worked on my dissertation. He granted me the strength and help I needed in completing this dissertation.

I would like to express my deep and sincere gratitude to my Supervisor, Professor Margaret Archibald for giving me the opportunity to do my research with her and providing invaluable guidance and advice throughout this dissertation. I am waohed at how she takes time to read my dissertation and give comments on how to improve on my writing. I appreciate her kind words when I solve problems I struggle with. She was never tired of my questions from time to time. I appreciate you ma for your efforts. It was a great privilege and honor to work and study under your guidance.

I am extremely grateful to my Husband, Jeremiah, for his financial support and words of encouragement from time to time. And to my parents, siblings and friends for their love and encouragements during this research.

Contents

Declaration of Authorship i

Acknowledgements iii

1 Introduction 1

2 Generating Functions 5

2.1 Operations on Ordinary Generating Functions 6

2.2 Symbolic approach involving generating functions 8

2.3 Extracting the coefficients of generating functions 12

3 Transfer Graphs and Transfer Matrices 15

3.1 Representing sequences of symbols in a transfer graph 15

3.2 Transfer Matrix 17

3.3 Obtaining the coefficients of weighted generating functions 21

4 Analytic Combinatorial Approach 28

4.1 Sequencing 28

4.2 Read mapping problem 30

4.3 Substitution Error Model 32

4.4 Substitution and deletion 38

4.5 Substitutions, Deletions and Insertions 46

4.5.1 Approximation 54

5 Recurrence Approach 55

5.1 Recurrence approach (substitution error model) 55

5.2 Recurrence approach (substitution and deletion error model) 58
5.3 Recurrence approach (substitution, deletion and insertion error model) . . . 61

6 Conclusion **69**

A Appendix **71**
A.1 Numerical bisection using Python 71
A.1.1 Substitution only, Example 4.1 71
A.1.2 Substitution and deletion only, Example 4.2 71
A.1.3 Substitution, deletion, insertion, Example 4.2 72
A.2 Recurrence Plots 73
A.2.1 Recurrence plots for substitution only 73
A.2.2 Recurrence plots for substitution and deletion 74
A.2.3 Recurrence plots for substitution, deletion and insertion 75

Bibliography **77**

List of Figures

3.1 A walk on the graph above represents a sequence of nucleotides without the edge CG [13, Page 8]. 16
3.2 *The matrix represents the adjacency matrix of Figure 3.1* [14, Page 8]. . . 16
3.3 *A generic transfer graph [13, Page 9].* 17
3.4 A walk on the graph above represents a sequence of nucleotides with labelled edges except the edge CG 18

4.1 *The diagram describes what DNA is made up of.* 28
4.2 *This is a sequence of symbols (read) where the white boxes, thick horizontal bar, light grey boxes, and dark grey boxes represent the correct nucleotides, deletions, insertions, and substitutions respectively [14, Page 8]* 30
4.3 *This represents a sequence of error-free intervals together with their sizes [14, Page 8].* 30
4.4 *The figure shows how reads are being mapped to a reference genome [32]* 31
4.5 *The transfer graph representing reads with substitutions (see [14, Page 9]).* 32
4.6 *The transfer graph representing the reads with substitutions and deletions [14, Page 12].* 39
4.7 *The graph reflects the transfer graph of the substitutions, deletions, and insertions error model.* 46
4.8 *The graphs depict the substitutions, deletions, and insertions transfer graph, with the subgraph (B) detached from (A) [14, Page 14].* 47

5.1 *The solid lines represent* $P(S)$ *while the dashed lines represent* $1 - P(S)$*. The* y*-axis represents the probability plotted against the read size on the* x*-axis at* $p = 0.08,\ 0.1$ *and* 0.12*.* 57

5.2 *The solid lines represent $P(S)$ while the dashed lines represent $1 - P(S)$. The y-axis represents the probability plotted against the read size on the x-axis at $\delta = 0.14,\ 0.15$ and 0.16.* 60

5.3 *The solid lines represent $P(S)$ while the dashed lines represent $1 - P(S)$. The y-axis represents the probability plotted against the read size on the x-axis at $r = 0.04,\ 0.05$ and 0.06.* 68

List of Special Symbols

Here are some symbols used in the dissertation.

$G(z)$	weighted generating function
$\mathcal{G}^+$	the set of non-empty sequences of the element $\mathcal{G}$
$[z^k]G(z)$	the total weight of all combinatorial objects with k occurrences
$\mathcal{G}^*$	is the set of sequences of the element $\mathcal{G}$
$M_*(z)$	weighted generating function represented on transfer matrix
$\psi(z)$	weighted generating function of any other sequences
S	is the statement that a read of size $s = k$ has no seed
$P(S)$	probability that a read of size $s = k$ has no seed
$1 - P(S)$	probability that a read of size $s = k$ has a seed
p	probability that a substitution error occurs
pz	weighted generating function of substitution
$F_\gamma(z)$	truncated weighted generating function of error-free intervals
$F(z)$	weighted generating function of error-free intervals
$R(z)$	weighted generating function of all reads
$S_\gamma(z)$	weighted generating function of a read of size k with no seed
$[z^k]R(z)$	total weights of all reads of length k
$[z^k]S_\gamma(z)$	total weights of reads of length k with no seed

Chapter 1

Introduction

Sequencing technologies are methods such as data analysis, sequencing, imaging and preparation of templates which in turn give insights on biological functions and differences of sequences [24]. The study of the genome's methylation pattern is one biological application that has been enhanced by sequencing technologies [10]. For many of these biological applications, alignment of sequences comes first, as reads (sequences) need to be aligned with the reference genome to know the location of each read and find the similarities [28]. Due to the amount of data produced by new sequencing technologies, many alignment tools have been introduced to help in aligning sequences.

"Sequence alignment is a way of arranging sequences of DNA, RNA or proteins to identify regions of similarity, that can be a consequence of functional, structural, or evolutionary relationships between the sequences" [4]. In a simple form, sequence alignment in bioinformatics involves comparing and detecting similarities in biological sequences [4]. The two types of sequence alignment methods are pairwise sequence alignment and multiple sequence alignment (MSA). Pairwise sequence alignment involves aligning two sequences, while multiple sequence alignment allows aligning many sequences [8], [17].

Sequence alignment can be classified into two approaches, namely: local and global alignments [20]. A global alignment approach is an expression of global optimization which forces an alignment to cover the length of the entire sequence. This type of alignment is suitable for protein-coding genes and genome sequences with a short length. On the other hand, local alignment is necessary for long genomic DNA. In local alignment, a similar

region between two long sequences is identified by taking account of the medium, large-scale rearrangement and large sequence insertions and deletions [20]. The problem of finding seeds is a local alignment problem, while the mapping problem is a global alignment problem. The sequences involved in local alignment can be DNA, RNA, or protein sequences, and databases containing a considerable number of sequences (e.g., patented protein sequences and protein databases (PDB)) [30]. Optimal and heuristic sequence alignment are two categories of sequence alignment that have been presented over the years. The goal of optimal sequence alignment is to get the best alignment, while heuristic sequence alignment aims for a close match to the best alignment [18]. The size of DNA and protein sequences have grown and have become important in analysis of sequenced genes and proteins because they give room for finding similarities between sequences. Algorithms that have been used in the optimal sequence alignment include the Needleman-Wunsch algorithm [25], and the Smith-Waterman algorithm [29]. The Needleman-Wunsch algorithm is a type of optimal global alignment while the Smith-Waterman algorithm is a type of optimal local alignment [19]. The Needleman-Wunch algorithm and Smith-Waterman algorithm are dynamic programming algorithms for sequence alignment that employ a scoring function to determine the similarity between sequences [8]. The *score* applied to determine the similarity between sequences is a numerical value that represents the performance of an alignment and based on this score, a particular alignment is preferred to another. A gap penalty (which involves inserting or extending gaps) is used to discover the optimal alignment [8]. The effects of insertion, deletion, and substitution errors over the years have produced sequences that have more or fewer gaps during alignment [20].

The sequence alignment problem has been addressed using a variety of computational approaches. These computational approaches include dynamic programming, which is slow but assures a correct solution, and heuristic algorithms, which are built for large-scale database searches and are fast but can produce inaccurate sequence alignment solutions. Dynamic programming involves simplifying complex problems by breaking them down into simpler ones in a recursive way [12]. The dynamic programming method is quadratic in time and the memory required is proportional to the total length of the two sequences that are to be aligned [12].

Over the years, in bioinformatics, the seeding heuristic approach has been the most widely used strategy in speeding up sequence alignment [14]. The seeding heuristic approach is the most used because alignment solutions are obtained faster compared to other alignment methods that have been used. It has to do with searching regions where sequences are similar. A *seed* is an exact local alignment of the read with the genome. The advantage of the seeding method is that it locates the seed quickly; nevertheless, it is possible that no seed may exist in the sequences being aligned. Heuristics sequence alignment does not produce an optimal solution but is preferred to the optimal sequence alignment as it reduces the time needed in obtaining the best alignment. We have heuristic local alignments and heuristic global alignments. The Basic Local Alignment Search Tool (BLAST) [1] and Fast-All (FASTA) [19] are examples of heuristic local alignments while LAGAN [7] and MUMMER [11] are examples of heuristic global alignments.

FASTA was first used to find similarities between sequences before BLAST [2], [26]. The FASTA method includes the use of a scoring matrix and classifying sequences into fragments of k-tuples of specific sizes to obtain a region of similarity [19]. FASTA aligns sequences faster than optimal alignment algorithms but does not produce an optimal solution because scores lower than a certain cutoff are not regarded for alignment [19].

The use of seeds in sequence alignment is the most used approach for comparing/aligning large genome sequences. Seeds have been employed for large-scale local alignment in the whole-genome sequence and multiple sequence alignment algorithms [3], [6]. In [31], seeds were grouped into three namely, basic seeds (seeds that determine a match or mismatch between aligned sequences), transition seeds (seeds that allow transition mutations at certain positions) and score seeds (seeds that involve computing scores to obtain alignment). Seeding heuristics were used in the BLAST algorithm [1], to compare protein sequences and DNA sequences to determine the similarity between them. Altschul et al [1], estimated BLAST heuristics by arriving at a reasonably accurate score identity for the seeds to cut down on the amount of time used in alignment. The discovery of the indexing by the Burrows-Wheeler transform made it possible to map short reads. The Burrows-Wheeler Aligner and Bowtie make use of the Burrows-Wheeler transform to map short reads. The indexing method helps in determining the number of times the substring

occurs more quickly, see, for instance, [21], [22].

The problems discussed earlier regarding seeding heuristics can be addressed with the help of concepts from *analytic combinatorics*. In [14], Guillaume J. Filion used analytic combinatorics in studying the read mapping problem and calculated the probability that a read of size k has no seed. In this dissertation, we denote the probability that a read of size k has no seed and probability that a read of size k has a seed as $P(S)$ and $1 - P(S)$ respectively. The seed used in [14] is the basic seed which is concerned with the finding an exact match between aligned sequences. Analytic combinatorics is also applied to estimating the number of reads where seeding heuristics fail. The solution of the heuristics approach is greatly influenced by the kind of errors, the probability of errors occurring, the size of the read and the length of the seed. The paper which we are studying, [14], aims to show how instrumental analytic combinatorics is in solving sequence alignment problems in bioinformatics. Some of the tools of analytic combinatorics that are used in solving some alignment problems are generating functions, transfer matrices, transfer graphs, singularity analysis and asymptotic approximation.

In this paper, [14], Filion obtained the weighted generating function of the reads represented as a sequence of objects and approximated the probabilities, which involved solving polynomial equations. I computed the weighted generating functions of all reads and reads that have no seed by constructing sequences of combinatorial objects using transfer graphs and transfer matrices as formulated by Filion in more detail. I consider a read mapping problem and calculate $1 - P(S)$ by first obtaining $P(S)$ under different error models (substitution only; substitution and deletion; and substitution, deletion and insertion). In this dissertation, I have used Filion's suggestion that the recurrence approach is more accurate for smaller read sizes. I have elaborated on finding $P(S)$ using the recurrence approach to obtain more precise probabilities than provided by Filion with some examples. I also plotted the graph of $P(S)$ and $1 - P(S)$ using Mathematica using the values given in the examples. When we compared the probabilities obtained using analytic combinatorics and the recurrence, we arrived at the same numerical value correct to four decimal places for read sizes $(50, 100,$ and $200)$ and three decimal places for read size 25.

Chapter 2

Generating Functions

In this section, we discuss generating functions and how we relate them to sequences of combinatorial symbols (see [16], [27]).

Definition 2.1. Given a sequence $(g_0, g_1, g_2, ..., g_k, ...)$, the function

$$G(z) = g_0 + g_1 z + g_2 z^2 + ... = \sum_{k \geq 0} g_k z^k, \tag{2.1}$$

is called the *ordinary generating function (OGF)* of the sequence [27].

Example 2.1. The generating function associated with $(1, 1, 1, 1, ...)$ is

$$G(z) = 1 + z + z^2 + z^3 + ... = \frac{1}{1-z} = \sum_{k \geq 0} z^k.$$

Example 2.2. The generating function associated with the sequence $(1, 2^1, 2^2, 2^3 ...)$ is

$$G(z) = 1 + 2z + 2^2 z^2 + ... + 2^k z^k + ... = \frac{1}{1-2z} = \sum_{k \geq 0} 2^k z^k.$$

The basic operations of the generating functions discussed below allow one to solve more complex generating function problems.

2.1 Operations on Ordinary Generating Functions

1. Scaling: Given the sequence $(g_0, g_1, g_2, ..., g_n, ...)$, the generating function of the sequence $(g_0, \alpha^1 g_1, \alpha^2 g_2, ..., \alpha^n g_n, ...)$ is

$$G(\alpha z) = \sum_{n \geq 0} g_n (\alpha z)^n = g_0 + g_1 \alpha^1 z + g_2 \alpha^2 z^2 + ... + g_n \alpha^n z^n + ...$$

2. Addition: Given two sequences $(g_0, g_1, g_2, g_3, ..)$ and $(h_0, h_1, h_2, h_3, ...)$ with generating functions $G(z)$ and $H(z)$ respectively, $G(z) + H(z)$ is the generating function for the sequence $(g_0 + h_0, g_1 + h_1, g_2 + h_2, g_3 + h_3, ...)$ written as

$$G(z) + H(z) = \sum_{n \geq 0} (g_n + h_n) z^n.$$

3. Right Shifting : The generating function of the sequence $(\underbrace{0, 0, ..., 0}_{k}, g_0, g_1, g_2, g_3, ...)$, is

$$\begin{aligned} g_0 z^k + g_1 z^{k+1} + g_2 z^{k+2} + g_3 z^{k+3} + ... &= z^k (g_0 + g_1 z + g_2 z^2 + g_3 z^3 + ...) \\ &= z^k G(z) \\ &= z^k \sum_{n \geq 0} g_n z^n \\ &= \sum_{n \geq 0} g_n z^{n+k}. \end{aligned}$$

   We change the limit of summation by setting $n = n - k$ and obtain the generating function of the sequence as

$$\sum_{n \geq k} g_{n-k} z^n.$$

4. Differentiation: If $G(z) = \sum_{n \geq 0} g_n z^n$, then the generating function for the sequence $(g_1, 2g_2, 3g_3, ...)$ is $G'(z)$. So,

$$g_1 + 2g_2 z + 3g_3 z^2 + ... = \sum_{n \geq 1} n g_n z^{n-1}.$$

We change the limit of summation by setting as $n = n + 1$ and obtain

$$g_1 + 2g_2z + 3g_3z^2 + ... = \sum_{n\geq 0}(n+1)g_{n+1}z^n.$$

5. Convolution: Given the sequences $(g_0, g_1, g_2, g_3, ...)$ and $(h_0, h_1, h_2, h_3, ...)$, the generating function of the convolution of the given sequences is given as

$$\begin{aligned} G(z)\cdot H(z) &= \sum_{n\geq 0}\Big(\sum_{0\leq k\leq n} g_k h_{n-k}\Big)z^n \\ &= g_0h_0 + (g_1h_0 + g_0h_1)z + (g_2h_0 + g_1h_1 + g_0h_2)z^2 + ... \end{aligned} \tag{2.2}$$

6. Difference: The generating function for the sequence $(g_0, g_1 - g_0, g_2 - g_1, ..., g_n - g_{n-1}, ...)$, is given by

$$(1-z)G(z) = g_0 + \sum_{n\geq 1}(g_n - g_{n-1})z^n.$$

7. Integration: The generating function of the sequence $(0, g_0, \frac{g_1}{2}, \frac{g_2}{3}, \frac{g_3}{4}, ..., \frac{g_{n-1}}{n}, ...)$, is

$$\begin{aligned} \int_0^z G(t)dt &= \int_0^z \Big(\sum_{n\geq 0} g_n t^n\Big)dt \\ &= \sum_{n\geq 0}\Big(\int_0^z g_n t^n dt\Big) \\ &= \sum_{n\geq 0}\Big[g_n\frac{t^{n+1}}{n+1}\Big]_0^z \\ &= \sum_{n\geq 0} g_n\frac{z^{n+1}}{n+1} \\ &= \sum_{n\geq 1}\frac{g_{n-1}}{n}z^n. \end{aligned}$$

8. Partial sums: Partial sums are a special case of convolution. The generating function of the sequence $(g_0,\ g_0 + g_1,\ g_0 + g_1 + g_2,\ ...,\ \sum_{0\leq k\leq n} g_k,\ ...)$, is given as

$$\frac{G(z)}{1-z} = \sum_{n\geq 0}\Big(\sum_{0\leq k\leq n} g_k\Big)z^n.$$

2.2 Symbolic approach involving generating functions

We translate combinatorial objects to generating functions to have an understanding of the relationship between them. In simple terms, this has to do with describing the sequence of objects using mathematical symbols [27]. The sequences of objects have size s and a weight denoted as ω. In general combinatorial examples, the weight of the sequences of objects could be the length of the sequence or size of a subset of a set. For example, if $g = (g_1, g_2, g_3, ...g_k)$, the weight of g is the sum of the elements in g, $\omega(g) = g_1 + g_2 + g_3 + ... + g_k$ and the size $s(g) = k$ is the number of elements in g.

Definition 2.2. A *weighted generating function*, $G(z)$ of a set $\mathcal{G}$ where $\mathcal{G}$ is a set of combinatorial objects, is defined as

$$G(z) = \sum_{g \in \mathcal{G}} \omega(g) z^{|g|}, \tag{2.3}$$

where $|g|$ denotes the size and $\omega(g)$ refers to the weight of an object g [27].

The operations on combinatorial objects can be described in terms of operations on their weighted generating function. Examples of combinatorial objects include permutations, subsets of sets, and binary trees but our focus will be on sequences of symbols with finite alphabets.

Example 2.3. Let $\mathcal{G} = \{g\}$ and $\mathcal{H} = \{h\}$ where $|g| = 1$ and $|h| = 1$, then we assume that $\omega(g) = m$ and $\omega(h) = n$, so $G(z) = mz$ and $H(z) = nz$ are the weighted generating functions of $\mathcal{G}$ and $\mathcal{H}$ respectively [13].

We will calculate the weighted generating functions of sequences of symbols by breaking them down into simpler weighted generating functions which are already known. Below are some examples where we applied two operations on weighted generating functions.

1. Addition: Given that the weighted generating function of a pair of disjoint sets $\mathcal{G}$ and $\mathcal{H}$, are $G(z)$ and $H(z)$, the weighted generating function of the union of the sets $(\mathcal{G} \cup \mathcal{H})$ is $G(z) + H(z) = mz + nz$ [27], considering Example 2.3.

2. Multiplication: The elements in the cartesian product of $\mathcal{G}$ and $\mathcal{H}$ are defined as (g,h). So, the size and weight of the Cartesian product of the sets $\mathcal{G}$ and $\mathcal{H}$ are defined as $|(g,h)| = |g| + |h|$ and $\omega(g,h) = \omega(g)\omega(h)$ respectively. The Cartesian product of the sets $\mathcal{G}$ and $\mathcal{H}$ has a weighted generating function $G(z)H(z)$ [27]. Recall the convolution operation from the section above (see Equation (2.2)).

$$G(z)H(z) = \sum_{g\in\mathcal{G}} \omega(g)z^{|g|} \sum_{g\in\mathcal{H}} \omega(h)z^{|h|} = \sum_{(g,h)\in\mathcal{G}\times\mathcal{H}} \omega(g)\omega(H)z^{|g|+|h|}.$$

In the following two examples, we assume the weight of element g is m, and the weight of element h is n.

Example 2.4. Let $\mathcal{G}^4 = \{(g,g,g,g)\}$ be a tuple containing four elements. The tuple $\mathcal{G}$ has size s =4 and weight $\omega = m^4$. So, $G(z)^4 = m^4z^4$ is the weighted generating function of $\mathcal{G}^4$. [13].

Example 2.5. The set $(\mathcal{G}\cup\mathcal{H})^2 = \{\{g,g\},\{g,h\},\{h,g\},\{h,h\}\}$ has size $s = 4$ with each pair having size $s = 2$ and the weight of each of the pairs is denoted as m^2, mn, nm, and n^2 respectively. The weighted generating function of $(\mathcal{G}\cup\mathcal{H})^2$ can be written as

$$(mz+nz)^2 = (m^2+2mn+n^2)z^2 = (G(z)+H(z))^2,$$

by applying the addition and multiplication operations.

Remark. We use the results obtained from the addition and multiplication operations to construct a finite sequence of objects. This is referred to as the *Sequence principle*.

Proposition 2.1. Given a weighted generating function $G(z)$ of a set $\mathcal{G} = (g_0, g_1, g_2, ..., g_k, ...)$, and $g_0 = 0$, the weighted generating function of $\mathcal{G}^+ = \cup_{k=1}^{\infty}\mathcal{G}^k$, is

$$\frac{G(z)}{1-G(z)}.$$

Thus, the set which contains non-empty sequences of the elements of $\mathcal{G}$ is referred to as $\mathcal{G}^+$.

Proof. The weighted generating function of a set $\mathcal{G}^k$ is $G(z)^k$, for a fixed $k \geq 1$. Since the sets of $\mathcal{G}^k$ are disjoint, the weighted generating function of the union of the sets $\mathcal{G}^k$ is written as

$$\begin{aligned} G(z) + G(z)^2 + G(z)^3 + ... &= G(z)[1 + G(z) + G(z)^2 + ...] \\ &= G(z)\Big[\frac{1}{1 - G(z)}\Big] \end{aligned}$$

(see [13]).

Example 2.6. Let $\mathcal{G} = \{g\}$ and $\mathcal{G}^k = \{g, g, g, ..., g\}$ with size $s = k$ and weight $\omega = m^k$, then the weighted generating function is $G(z)^k = m^k z^k$. The union of $\mathcal{G}, \mathcal{G}^2, \mathcal{G}^3, ..., \mathcal{G}^k$ has its weighted generating function as

$$mz + m^2z^2 + m^3z^3 + ... + m^kz^k.$$

Note that

$$\frac{1}{1 - mz} = 1 + mz + m^2z^2 + m^3z^3 + ... + m^kz^k + ... = \sum_{k=0}^{\infty} m^kz^k,$$

and

$$(1 + mz + m^2z^2 + m^3z^3 + ... + m^kz^k + ...)(1 - mz) = 1$$

(see [27]). Then for $k > 1$, we have

$$(mz + m^2z^2 + m^3z^3 + ... + m^kz^k)(1 - mz) = mz - (mz)^{k+1}. \tag{2.4}$$

The expression $(mz)^{k+1}$ becomes zero as k increases where $|z| < \frac{1}{m}$, so $\cup_{k=1}^{\infty}\mathcal{G}^k$ has its weighted generating function as

$$mz + m^2z^2 + m^3z^3 + ... = \lim_{k\to\infty} \frac{mz(1 - (mz)^k)}{1 - mz} = \frac{mz}{1 - mz},$$

(see [13]).

□

Definition 2.3. The weighted generating function of an empty object ϵ, having a weight of 1 and size of 0 is 1 [27].

Proposition 2.2. Given a weighted generating function $G(z)$ of a set $\mathcal{G}$, the weighted generating function of the set $\mathcal{G}^*$ given by $\mathcal{G}^* = \cup_{k=0}^{\infty}\mathcal{G}^k$, is defined as

$$\frac{1}{1-G(z)}.$$

$\mathcal{G}^*$ refers to the set of sequences of the elements of $\mathcal{G}$ [27].

Proof. Let $\mathcal{G}^* = \{\epsilon\} \cup \mathcal{G}^+$. By Proposition 2.1, since the weighted generating functions of $\mathcal{G}^+$ and ϵ are $\frac{G(z)}{1-G(z)}$ and 1 respectively. So we have the weighted generating function of $\mathcal{G}^*$ as

$$\begin{aligned} 1 + \frac{G(z)}{1-G(z)} &= \frac{1-G(z)+G(z)}{1-G(z)} \\ &= \frac{1}{1-G(z)}. \end{aligned}$$

□

Example 2.7. From Proposition 2.2 and the addition operation we have that the weighted generating functions of $\mathcal{G}^*$ and $\mathcal{G} \cup \mathcal{H}$ are $\frac{1}{1-G(z)}$ and $mz + nz$ respectively. So, the weighted generating function of $(\mathcal{G} \cup \mathcal{H})^*$ is

$$\frac{1}{1-(mz+nz)} = 1 + (m+n)z + (m+n)^2z^2 + (m+n)^3z^3 + ... \tag{2.5}$$

(see [13]).

The equality in Equation 2.5 implies that the weighted generating function for a set $\{\{g,h\},\{g,h\},...,\{g,h\}\}$ containing k elements is $(mz+nz)^k$.

2.3 Extracting the coefficients of generating functions

The coefficient g_k of z^k in $G(z)$ is denoted as $[z^k]G(z)$. The expression $[z^k]G(z)$ is the total weight of all combinatorial objects with k occurrences (or of size k). We suppose that the coefficient g_k of z^k in $G(z)$ represents a count related to k objects with each object labelled for identification.

The different ways of computing the coefficients of generating functions include:

1. Applying Taylor's theorem which results in a more difficult problem to solve. We write Taylor's theorem as
$$[z^n]G(z) = \frac{G^{(n)}(0)}{n!},$$
to extract the coefficients given that the derivatives exist [27].

2. Expanding the generating function by breaking it into known simpler ones. The expansion can be done using convolutions and partial fraction decomposition.

Example 2.8. Find the coefficient of z^n in the ordinary generating function

$$(1-z)^2 \ln \frac{1}{1-z}$$

(see [27]).

Solution.

(i) We obtain the coefficient of z^n in the ordinary generating function using Taylor's theorem

$$\begin{aligned}
G^{(1)}(z) &= (1-z)\Big[1-2\ln\frac{1}{1-z}\Big] & \frac{G^{(1)}(0)}{1!} &= \frac{1}{1!} = \frac{1}{n}\\
G^{(2)}(z) &= 2\ln\frac{1}{(1-z)} - 3 & \frac{G^{(2)}(0)}{2!} &= \frac{-3}{2!} = \frac{-(n+1)}{n(n-1)}\\
G^{(3)}(z) &= \frac{2}{(1-z)} & \frac{G^{(3)}(0)}{3!} &= \frac{2}{3!} = \frac{2}{n(n-1)(n-2)}\\
G^{(4)}(z) &= \frac{2}{(1-z)^2} & \frac{G^{(4)}(0)}{4!} &= \frac{2}{4!} = \frac{2}{n(n-1)(n-2)}\\
&\vdots\\
G^{(n)}(z) &= \frac{2(n-3)!}{(1-z)^{n-2}} & \frac{G^{(n)}(0)}{n!} &= \frac{2(n-3)!}{n!} = \frac{2}{n(n-1)(n-2)}.
\end{aligned}$$

So the coefficients for $n \geq 3$ are

$$[z^n]G(z) = \frac{G^n(0)}{n!} = \frac{2}{n(n-1)(n-2)}.$$

(ii) We expand $(1-z)^2$, multiply by $\ln\frac{1}{1-z}$ and simplify below to obtain the coefficient

$$\begin{aligned}
(1-z)^2\ln\frac{1}{1-z} &= (1-2z+z^2)\ln\frac{1}{1-z}\\
&= \ln\frac{1}{1-z} - 2z\ln\frac{1}{1-z} + z^2\ln\frac{1}{1-z}\\
&= \sum_{n\geq 1}\frac{z^n}{n} - 2z\sum_{n\geq 1}\frac{z^n}{n} + z^2\sum_{n\geq 1}\frac{z^n}{n}\\
&= \sum_{n\geq 1}\frac{z^n}{n} - 2\sum_{n\geq 1}\frac{z^{n+1}}{n} + \sum_{n\geq 1}\frac{z^{n+2}}{n}.
\end{aligned}$$

We change the summation limits to obtain the coefficient below

$$\begin{aligned}[z^n](1-z)^2 \ln \frac{1}{1-z} &= [z^n]\left(\sum_{n\geq 1}\frac{z^n}{n} - 2\sum_{n\geq 2}\frac{z^n}{n-1} + \sum_{n\geq 3}\frac{z^n}{n-2}\right) \\ &= \frac{1}{n} - \frac{2}{n-1} + \frac{1}{n-2} \\ &= \frac{2}{n(n-1)(n-2)}, \quad \text{for } n \geq 3.\end{aligned}$$

We have the coefficient of the generating function at $n = 1, 2$ as

$$[z^n](1-z)^2 \ln \frac{1}{1-z} = \begin{cases} \frac{1}{n}, & n = 1, \\ \frac{-(n+1)}{n(n-1)}, & n = 2 \\ \frac{2}{n(n-1)(n-2)}, & n \geq 3. \end{cases}$$

Sometimes, we employ another approach used in Filion's paper to obtain the asymptotics for the coefficient of a generating function $G(z)$. We assume that $G(z)$ is the rational function $U(z)/V(z)$ with $U(z)$ and $V(z)$ being relatively prime and $V(z) \neq 0$ where $V(z)$ has one solution with minimum modulus z_1 then,

$$[z^k]G(z) = [z^k]\frac{U(z)}{V(z)} \sim -\frac{U(z_1)}{V'(z_1)}\frac{1}{z_1^{k+1}}$$

(see [27], [16]). This approach of extracting coefficients is preferred to Taylor's theorem if we cannot explicitly find the coefficients of the generating function $G(z)$ [27].

Chapter 3

Transfer Graphs and Transfer Matrices

This chapter discuss the tools addressed by Filion needed to obtain the required probabilities later in this dissertation.

3.1 Representing sequences of symbols in a transfer graph

When it comes to aligning sequences, we are looking for similar patterns and whether the sequences match. Generating functions come in handy when searching for similar patterns in sequence alignment. We represent these sequences on a transfer graph which we translate into a transfer matrix to get the weighted generating function [23].

We start with an example to lead up to the transfer graphs and transfer matrices. For example, the animal genomes where DNA methylation occurs on dinucleotide CG alone.

Definition 3.1. *DNA methylation* is a mechanism that controls gene expression.

Definition 3.2. A *walk* is a finite sequence of consecutive edges which joins a sequence of vertices (see [5]).

Remark. Figure 3.1 is a directed graph (edges with arrows) consisting of vertices $\{A, C, G, T\}$, edges {AC, CA, AG, GA, AT, TA, CT, TC, GT, TG, GC but not CG} and loops {AA, CC, GG and TT}. A loop is an edge that connects a vertex to itself. A sequence with no

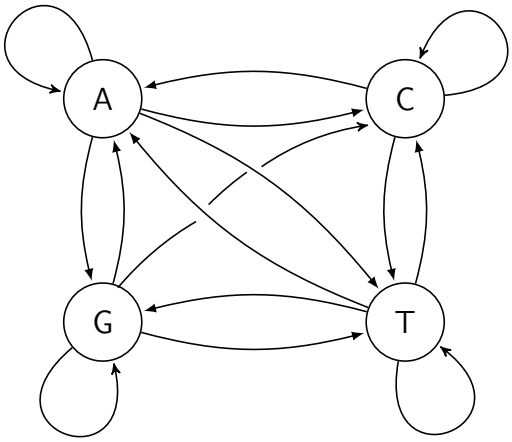


FIGURE 3.1: A walk on the graph above represents a sequence of nucleotides without the edge CG [13, Page 8].

dinucleotide CG is thus represented as a walk (sequence of edges) on the graph in Figure 3.1.

We do not want to confuse the adjacency matrix with the transfer matrix, so we describe the adjacency matrix below.

Definition 3.3. The *adjacency matrix* or *connection matrix* denoted by M is a matrix where the elements dictate if the pairs of vertices are adjacent or not. The elements contained in the adjacency matrix are either 1 or 0 [5].

$$M = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

FIGURE 3.2: *The matrix represents the adjacency matrix of Figure 3.1* [14, Page 8].

In Figure 3.2, the elements 1 and 0 indicate whether one nucleotide (vertex) follows another or not, respectively, thereby implying that the position with element 0 represents that vertex G does not follow vertex C.

Remark. In this dissertation, when we say edge CG does not occur, this does not imply that edge GC does not occur as we can see in Figure 3.1. In Figure 3.2, the element in

position $(2,3)$ is 1, which represents that the dinucleotide GC can occur (vertex C follows vertex G).

Definition 3.4. A graph that has its edges labeled with weighted generating functions is called a *"transfer graph"*. A transfer graph is a graph that contains outgoing edges from the open circle "head vertex" and incoming edges into the closed circle "tail vertex", as well as some edges of the subgraph in between (see [13]).

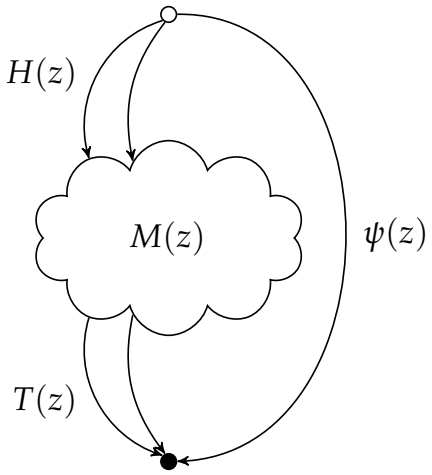


FIGURE 3.3: *A generic transfer graph [13, Page 9].*

The labels of the edges are the weighted generating functions of the classes of combinatorial objects (here: DNA sequences).

3.2 Transfer Matrix

Definition 3.5. A *transfer matrix* is a matrix whose entries are the weighted generating functions of labelled edges of a transfer graph at position (i,j), where i represents the row and j represents the column of the matrix [14]. We denote the transfer matrix as $M_*(z)$.

The subgraph in Figure 3.3 above is represented by $M(z)$ which translates to an $m \times m$ matrix. The edges from the open circle (head vertex) to the subgraph are represented by $H(z)$, a vector of m weighted generating functions. The edges from the subgraph to the closed circle (tail vertex), $T(z)$ is a vector of m weighted generated functions.

We go back to our example on DNA methylation in Section 3.1. In Figure 3.4, we give an example of a transfer graph without the head and tail vertex.

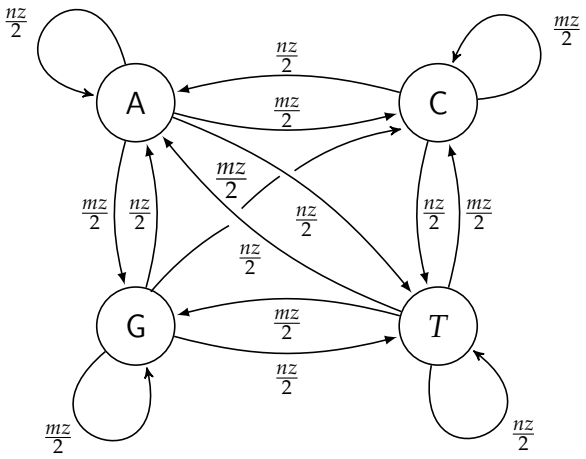


FIGURE 3.4: A walk on the graph above represents a sequence of nucleotides with labelled edges except the edge CG .

We now specify the transfer matrix for the sequences of nucleotides without CG using Figure 3.4. This is not the entire transfer matrix because we do not include the head and tail vectors. If Gs and Cs have a frequency of $m/2$ and As and Ts have a frequency of $n/2$, where $m = 1 - n$, the weighted generating functions of the single nucleotides are $C(z) = G(z) = mz/2$ and $A(z) = T(z) = nz/2$ respectively.

$$M(z) = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} \frac{nz}{2} & \frac{mz}{2} & \frac{mz}{2} & \frac{nz}{2} \\ \frac{nz}{2} & \frac{mz}{2} & 0 & \frac{nz}{2} \\ \frac{nz}{2} & \frac{mz}{2} & \frac{mz}{2} & \frac{nz}{2} \\ \frac{nz}{2} & \frac{mz}{2} & \frac{mz}{2} & \frac{nz}{2} \end{bmatrix} \end{array}$$

In obtaining the weighted generating function of all sequences (single nucleotides) represented in Figure 3.3, we introduce Theorem 3.1.

Theorem 3.1. *Let*

$$M_*(z) = \begin{pmatrix} 0 & H(z) & \psi(z) \\ 0 & M(z) & T(z) \\ 0 & 0 & 0 \end{pmatrix}, \tag{3.1}$$

be the transfer matrix interpreted from the transfer graph in Figure 3.3 where $H(z)$ *is a vector of dimension* $1 \times m$, $T(z)$ *is also a vector of dimension* $m \times 1$, $M(z)$ *is the* $m \times m$ *dimension matrix corresponding to the subgraph, and* $\psi(z)$ *is a 1 by 1 matrix of dimension* 1. *The sequences obtained from the transfer matrix* $M_*(z)$ *have the weighted generating function*

$$\psi(z) + H(z) \cdot Q(z)^{-1}.T(z), \tag{3.2}$$

where $Q(z) = I - M(z)$ *and we assume that* $M(0) = 0$ *(null matrix) and that all* $M(z)$*'s eigenvalues have modulus* ≤ 1 *[14].*

Proof. In order to find the determinant, we first compute $1 - M_*(z)$ as

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & H(z) & \psi(z) \\ 0 & M(z) & T(z) \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -H(z) & -\psi(z) \\ 0 & 1 - M(z) & -T(z) \\ 0 & 0 & 1 \end{pmatrix}.$$

So, the determinant $|I - M_*(z)|$ is given by

$$1(1 - M(z) - 0) + H(z)(0 - 0) - T(z)(0 - 0) = 1 - M(z).$$

Now, the cofactor matrix of $1 - M_*(z)$ is obtained by getting the determinant of every 2×2 minor matrix and writing the solutions obtained in a new matrix form. For the first row elements

$$\begin{vmatrix} 1 - M(z) & -T(z) \\ 0 & 1 \end{vmatrix} = 1 - M(z),$$

$$- \begin{vmatrix} 0 & -T(z) \\ 0 & 1 \end{vmatrix} = 0,$$

$$\begin{vmatrix} 0 & 1 - M(z) \\ 0 & 0 \end{vmatrix} = 0.$$

For the second row elements

$$-\begin{vmatrix} -H(z)0 & -\psi(z) \\ 0 & 1 \end{vmatrix} = H(z),$$

$$\begin{vmatrix} 1 & -\psi(z) \\ 0 & 1 \end{vmatrix} = 1,$$

$$-\begin{vmatrix} 1 & -H(z) \\ 0 & 0 \end{vmatrix} = 0.$$

For the third row elements

$$\begin{vmatrix} -H(z) & -\psi(z) \\ 1-M(z) & -T(z) \end{vmatrix} = H(z)T(z) + \psi(z)(1-M(z)),$$

$$-\begin{vmatrix} 1 & \psi(z) \\ 0 & -T(z) \end{vmatrix} = T(z),$$

$$\begin{vmatrix} 1 & -H(z) \\ 0 & 1-M(z) \end{vmatrix} = 1-M(z).$$

Now the cofactor matrix is represented as

$$\begin{pmatrix} 1-M(z) & 0 & 0 \\ H(z) & 1 & 0 \\ H(z)T(z)+\psi(z)(1-M(z)) & T(z) & 1-M(z) \end{pmatrix}.$$

The transpose of the cofactor matrix called the adjoint of the matrix denoted as Adj() is

$$\begin{pmatrix} 1-M(z) & H(z) & H(z)T(z)+\psi(z)(1-M(z)) \\ 0 & 1 & T(z) \\ 0 & 0 & 1-M(z) \end{pmatrix}.$$

So,

$$(I - M_*(z))^{-1} = \frac{1}{\mathrm{Det}(I - M_*(z))}\mathrm{Adj}((I - M_*(z))$$
$$= \frac{1}{1 - M(z)} \begin{pmatrix} 1 - M(z) & H(z) & H(z)T(z) + \psi(z)(1 - M(z)) \\ 0 & 1 & T(z) \\ 0 & 0 & 1 - M(z) \end{pmatrix}.$$

The weighted generating function of all possible sequences starting at the head vertex and ending at the tail vertex represented in the transfer matrix (3.1) is given by

$$\frac{H(z)T(z) + \psi(z)(1 - M(z))}{1 - M(z)} = H(z)T(z)(1 - M(z))^{-1} + \psi(z),$$

which is the top right entry of the inverse of $I - M(z)$. □

Remark. The primary purpose of the transfer graphs and the transfer matrices is to calculate the weighted generating function of the sequences represented by the paths in a transfer graph. Theorem 3.1 above gives an insight into how the weighted generating function of sequences is calculated given a transfer graph and the transfer matrix.

Let $G(z) = \sum_{k=1}^{\infty} g_k z^k$ for some neighborhood of $z = 0$, we proceed to obtain $[z^k]G(z)$ which can be defined as g_k (see Chapter 2).

Remark. The use of the analytical combinatorics approach allows one to approximate the coefficients of the weighted generating function efficiently.

3.3 Obtaining the coefficients of weighted generating functions

Lemma 3.1. For $n \geq 0$ and every complex number $g \neq 0$ [14],

$$\frac{1}{(1 - z/g)^{n+1}} = \sum_{k=0}^{\infty} \binom{k+n}{n} \frac{z^k}{g^k}.$$

Proof. Let $G(z) = z/g$ in Proposition 2.2. The weighted generating function of the sequences of the elements of $\mathcal{G}$ denoted by $\mathcal{G}^*$ is

$$\frac{1}{1-z/g} = \sum_{k=0}^{\infty} \frac{z^k}{g^k}. \tag{3.3}$$

Now differentiating both sides of Equation (3.3) n times. For the left-hand side, we equate $\frac{1}{1-z/g}$ to y and differentiate as follows.

$$\begin{aligned} y = (1-z/g)^{-1}, \quad y' &= 1/g(1-z/g)^{-2} \\ y'' &= 2/g^2(1-z/g)^{-3} \\ &\vdots \\ y^n &= n!/g^n \frac{1}{(1-z/g)^{n+1}}. \end{aligned}$$

The derivatives for the right hand side are:

$$\begin{aligned} y &= \sum_{k=0}^{\infty} \frac{z^k}{g^k}, \\ y' &= \sum_{k=0}^{\infty} k\frac{z^{k-1}}{g^k} \\ y'' &= \sum_{k=0}^{\infty} k(k-1)\frac{z^{k-2}}{g^k} \\ &\vdots \\ y^n &= \sum_{k=0}^{\infty} k(k-1)...(k-n+1)\frac{z^{k-n}}{g^k}. \end{aligned}$$

Now equating the nth derivative of the left and right hand sides obtained above and shifting the sum index to obtain the required expression,

$$\frac{n!}{g^n}\frac{1}{(1-z/g)^{n+1}} = \sum_{k=n}^{\infty} k(k-1)...(k-n+1)\frac{z^{k-n}}{g^k}.$$

Then

$$\begin{aligned}\frac{1}{(1-z/g)^{n+1}} &= \frac{g^n}{n!}\sum_{k=n}^{\infty}\frac{k!}{(k-n)!}\frac{z^{k-n}}{g^k}\\ &= \sum_{k=n}^{\infty}\frac{k!}{n!(k-n)!}\frac{z^{k-n}}{g^{-n+k}}\\ &= \sum_{k+n=n}^{\infty}\frac{(k+n)!}{n!k!}\frac{z^k}{g^k} \quad \text{replace } k \text{ by } k+n\\ &= \sum_{k=0}^{\infty}\binom{k+n}{n}\frac{z^k}{g^k}.\end{aligned}$$

□

Theorem 3.2. *Given that $U(z)$ and $V(z)$ are polynomials, let $G(z) = U(z)/V(z)$ where the greatest common divisor of U and V is 1 and $V(0)$ is not equal to zero. If $V(z)$ has exactly one solution, with minimum modulus z_1, then*

$$[z^k]G(z) \sim -\frac{U(z_1)}{V'(z_1)}\frac{1}{z_1^{k+1}}.$$

Proof. We can factor the polynomial V as $(z-z_1)(z-z_2)^{v_2},...,(z-z_n)^{v_n}$, where $|z_1| < |z_2| \leq,...,\leq |z_n|$. There exist the complex numbers $\beta_1, \beta_{2,1},..., \beta_{2,v_2},..., \beta_{n,1},..., \beta_{n,v_n}$, such that when $\frac{U(z)}{V(z)}$ is expanded using partial fractions we have

$$\frac{U(z)}{V(z)} = \frac{\beta_1}{z_1-z} + \frac{\beta_{2,1}}{z_2-z} + \frac{\beta_{2,2}}{(z_2-z)^2} + ... + \frac{\beta_{2,v_2}}{(z_2-z)^{v_2}} + ... + \frac{\beta_{n,1}}{z_n-z} + ... + \frac{\beta_{n,v_n}}{(z_n-z)^{v_n}}. \tag{3.4}$$

Now applying Lemma 3.1 by expanding the terms of the sum in Equation (3.4) as

$$\frac{\beta_{j,m}}{(z_j-z)^m} = \frac{\beta_{j,m}}{(z_j(1-z/z_j))^m} = \frac{\beta_{j,m}}{{z_j}^m(1-z/z_j)^m}.$$

Recall that

$$\frac{1}{(1-z/z_j)^{n+1}} = \sum_{k=0}^{\infty}\binom{k+n}{n}\frac{z^k}{{z_j}^k},$$

then

$$\frac{1}{(1-z/z_j)^m} = \sum_{k=0}^{\infty} \binom{k+m-1}{m-1} \frac{z^k}{{z_j}^k},$$

where the knowledge of generating functions in [27] was applied .

We expand each of the n terms of Equation (3.4) as

$$\begin{aligned}\frac{\beta_{j,m}}{{z_j}^m(1-z/z_j)^m} &= \frac{\beta_{j,m}}{{z_j}^{m-1}.z_j(1-z/z_j)^m} \\ &= \frac{\beta_{j,m}}{{z_j}^{m-1}} \sum_{k=0}^{\infty} \binom{k+m-1}{m-1} \frac{z^k}{{z_j}^{k+1}}.\end{aligned}$$

Now by substituting the above equation into Equation (3.4), to obtain the coefficient of z^k in $G(z)$ as

$$[z^k]G(z) = \frac{\beta_1}{{z_1}^{k+1}} + \frac{\alpha_{2,k}}{{z_2}^{k+1}} + ... + \frac{\alpha_{n,k}}{{z_n}^{k+1}}, \tag{3.5}$$

where

$$\alpha_{j,k} = \sum_{m=1}^{v_j} \binom{k+m-1}{m-1} \frac{\beta_{j,m}}{{z_j}^{m-1}} = O(k^{v_j-1}).$$

Since z_1 is the dominant singularity, the expression in Equation (3.5) is reduced to the first term as k increases, so

$$[z^k]G(z) \sim \frac{\beta_1}{{z_1}^{k+1}} \tag{3.6}$$

(see [27], [16]). To get β_1, the first term in Equation (3.4) was retained while the other terms were replaced by the ratio of the polynomials U_1 and V_1 below

$$\frac{U(z)}{V(z)} = \underbrace{\frac{U(z)}{(z_1-z)V_1(z)}}_{(a)} = \frac{\beta_1}{z_1-z} + \frac{U_1(z)}{V_1(z)}.$$

Since $z_1 - z$ does not divide V_1, we then multiply (a) through by $z_1 - z$ and set $z = z_1$ to obtain

$$\frac{U(z)}{V_1(z)} = \beta_1 + \frac{U_1(z)}{V_1(z)}(z_1 - z),$$

when $z_1 = z$,

$$\frac{U(z_1)}{V_1(z_1)} = \beta_1.$$

Now by differentiating $V(z) = (z_1 - z)V_1(z)$ using the product rule, the solution obtained is

$$V'(z) = (z_1 - z)V'(z) - V_1(z)$$

and let $z_1 = z$ to get

$$V'(z_1) = -V_1(z_1).$$

Then,

$$\beta_1 = -\frac{U(z_1)}{V'(z_1)}. \tag{3.7}$$

□

Remark. Equation (3.7) is referred to as an asymptotic approximation. It is more accurate when k gets large. The zeros of $V(z)$ are referred to as "singularities" of the generating function $G(z)$. Theorem 3.2 shows how the coefficients of a generating function are extracted and it also implies that the growth of the coefficients (a constant factor of a term as distinguished from a variable) of the generating function, $G(z)$, asymptotically is determined by the dominant singularity. In proving Theorem 3.2, the expression for a precise estimate of the coefficients of $G(z)$ was obtained.

Theorem 3.3. *If* $G(z) = U(z)/V(z)$ *where,* $G(z) = \sum_{k\geq 0} g_k z^k$ *with the degree of*

polynomial U less than the degree of polynomial of V, then the sequence $(g_0, g_1, g_2, g_3, ...)$, corresponds to a linear recurrence relation whose coefficients are constant (see [14]).

Proof. Let

$$U(z) = u_0 + u_1 z + ... + u_m z^m$$

and

$$V(z) = v_0 + v_1 z + ... + v_n z^n$$

where $m < n$. Then we have

$$U(z) = V(z) \sum_{k \geq 0} g_k z^k,$$

that is

$$u_0 + u_1 z + u_2 z^2 ... + u_m z^m = (v_0 + v_1 z + v_2 z^2 ... + v_n z^n)(g_0 + g_1 z + g_2 z^2 + g_3 z^3 + ...).$$

We compare coefficients below for $k < n$

$$\begin{aligned}
[z^0] &: u_0 = v_0 g_0 & g_0 &= u_0 / v_0 \\
[z^1] &: u_1 = v_1 g_0 + v_0 g_1 & g_1 &= (u_1 - v_1 g_0)/v_0 \\
[z^2] &: u_2 = v_2 g_0 + v_1 g_1 + v_0 g_2 & g_2 &= (u_2 - v_2 g_0 - v_1 g_1)/v_0 \\
&\vdots & &\vdots \\
[z^k] &: u_k = \sum_{n=0}^{k-n} v_{k-n} g_n & g_k &= \frac{u_k - \sum_{n=0}^{k-1} v_{k-n} g_n}{v_0}
\end{aligned}$$

where $u_k := 0$ if $m < k < n$.

For $k - n \geq 0$, we have

$$0 = v_0 g_k + v_1 g_{k-1} + v_2 g_{k-2} + ... + v_n g_{k-n},$$

so,

$$g_k = \frac{-(v_1 g_{k-1} + v_2 g_{k-2} + ... + v_n g_{k-n})}{v_0}.$$

□

Chapter 4

Analytic Combinatorial Approach

4.1 Sequencing

The sequencing process entails breaking down DNA sequences into shorter fragments or pieces. A read is a sequence of base pairs that matches all or part of a single DNA fragment obtained during the laboratory sequencing process. This sequencing procedure is used to determine the order of the nucleotides in DNA.

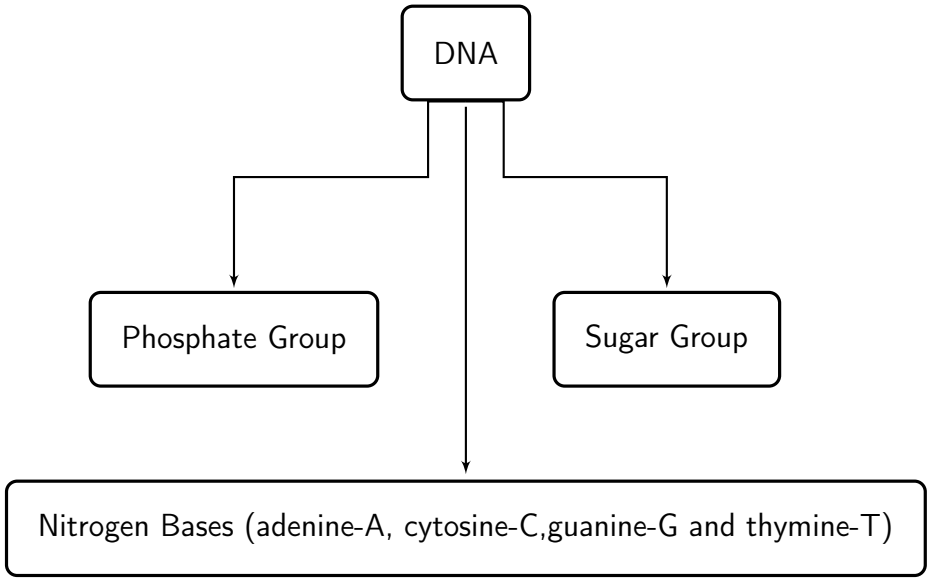


FIGURE 4.1: *The diagram describes what DNA is made up of.*

There are four different types of DNA nucleotides, namely: adenine (A), thymine (T), guanine (G), cytosine (C), where each of the nucleotides contains three molecules: sugar (deoxyribose), a phosphate group, and a nitrogen base.

Definition 4.1. A *DNA molecule* is a collection of nucleotides linked together in a long chain, and arranged one after the other. For example, a portion of DNA molecule could be written as ACTGTCGACCGTG.

Definition 4.2. A *read* is a sequence of nucleotides that corresponds to a part or the entirety of a DNA fragment.

For the purposes of this dissertation, we assume that the read and the reference genome come from the same organism and so there is no genuine genetic difference between their DNA and all differences are the result of sequencing errors.

During the sequencing process, some errors occur that have an impact on the process's outcome [14]. Substitutions, deletions, and insertions are examples of the errors that occur during the sequencing process.

1. An error which occurs when a nucleotide in the reference genome and the read differ is called *substitution*,

2. An error which occurs when a nucleotide can be found in the reference genome but not in the read is called *deletion*, and

3. An error which occurs when a nucleotide can not be found in the reference genome but is found in the read is called *insertion* [13].

When comparing the reference genome to the read, we aim to see if the read contains errors or if all the nucleotides are correct. The structure of a read represented in Figure 4.2 has four symbols, three of which represent a type of error (substitution, deletion, and insertion) and one representing a correct nucleotide. During the process of comparing the read to the reference genome referred to as the *alignment process*, the differences between the sequences are discovered, and the errors in the read are detected. However, if many errors are detected in the read, it becomes difficult to align the read correctly.

Remark. Correct nucleotides can be grouped in an error-free interval if they occur consecutively in a read.

A read can also be divided into 'intervals,' which are referred to as the maximum sequences of matching symbols [14].

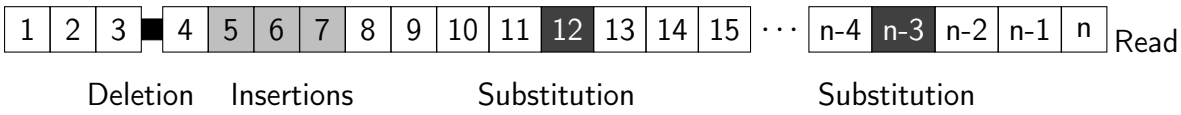


FIGURE 4.2: *This is a sequence of symbols (read) where the white boxes, thick horizontal bar, light grey boxes, and dark grey boxes represent the correct nucleotides, deletions, insertions, and substitutions respectively [14, Page 8]*

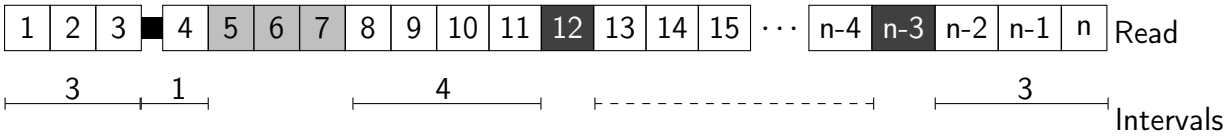


FIGURE 4.3: *This represents a sequence of error-free intervals together with their sizes [14, Page 8].*

Definition 4.3. A sequence of correct nucleotides is referred to as an *error-free interval*. A maximal error-free interval is a sequence of correct nucleotides that can neither be increased on the right side nor on the left side [13].

Remark. After discussing the concepts and theorems required to compute the $P(S)$ in Chapter 2 and Chapter 3, and also representing the sequences of symbols as error-free intervals, we proceed to calculate the $P(S)$ in a read mapping problem.

4.2 Read mapping problem

Read mapping is the process of mapping or aligning the reads obtained from sequencing to the reference genome. Mapping can be done by an individual or by using computer software depending on the number of data involved. By aligning reads to the reference genome, we hope to align a huge number of reads back to the respective regions where they were originally formed The problem of obtaining the best alignment or finding the best match is referred to as *read mapping problem*.

The length of the read and the rate at which errors occur during mapping have a huge impact on correctly mapping the read to the reference genome [15]. A genome sequence contains a large percentage of repeated patterns, which influences the mapping result. Due to these repeated patterns, there are possibilities of having more than one match between

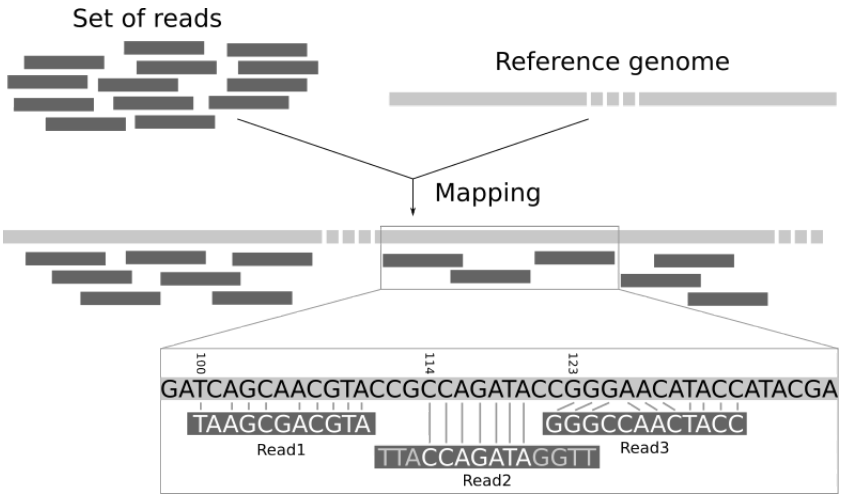


FIGURE 4.4: *The figure shows how reads are being mapped to a reference genome [32]*

the read and the reference genome, which makes locating the region from which the read was obtained difficult [15].

From previous research, BLAST (Basic Local Alignment Search Tool) and FASTA (Fast-all) are two of the algorithms that have been used in solving the read mapping problem, both of which provide a correct solution but are slow when dealing with large amounts of data. BLAST is used to find locally optimal sequence alignments, while FASTA is used to discover the relationship between the sequences [1]. Because of the large amount of data involved, some scientists chose a different approach, heuristic methods, which is much faster but does not guarantee a correct solution. The *Seeding method* is a popular heuristic method for mapping DNA sequences which we are studying in this research.

Definition 4.4. A *seed* is an exact local alignment of the read with the genome. For the purpose of this research, we denote the minimum size of the seed as γ.

The seed helps to streamline the search and allows us to select candidate sequences. When computing $P(S)$, we define a *seed* as a perfect match between the read and reference genome, and represent it as an error-free interval having size (length) $s \geq \gamma$ [14]. Hereafter, when we refer to an error-free interval, we assume it is maximal.

Based on the presence of various sequence errors, Filion (see [14]) estimated $P(S)$. Note that S represents the statement that a read has no seed of length γ. The weighted generating functions of reads with no seed of length γ are calculated by representing them

as sequences of error-free intervals which have size $s < \gamma$, interspersed by error symbols. The formula for computing $P(S)$ is

$$\frac{[z^k]S_\gamma(z)}{[z^k]R(z)} \tag{4.1}$$

where $R(z)$ is the weighted generating function of all the reads represented on a transfer graph and $S_\gamma(z)$ is the weighted generating function of the reads with no seed of length at least γ. The weights of reads of size k with no seed sum up to $[z_k]S_\gamma(z)$ and $[z_k]R(z)$ is the sum of the weight of all reads of size k.

In the sections that follow, I present a more detailed analytical combinatoric approach for obtaining probability that a read has no seed under three different error models computed by Filion and extend his logic to calculate the probability of a seed occuring in a read.

4.3 Substitution Error Model

This error model is the simplest we will be studying because it has a simple transfer graph with the edges easily represented and understood. We ignore the deletions and insertions and model the substitution error alone. We assume that the probability at which a substitution error occurs is p [14]. This error model is represented on a transfer matrix where the subgraph consist of only two vertices: the error-free interval and single substitutions.

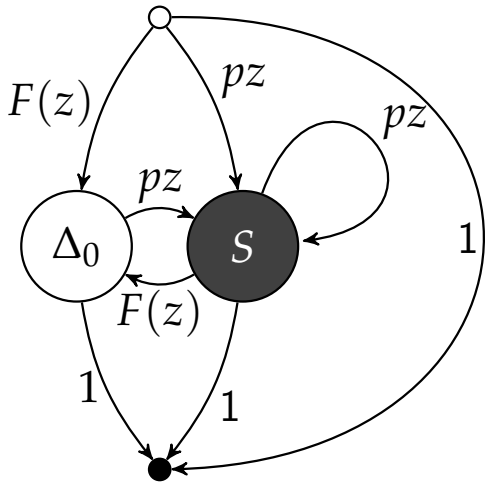


FIGURE 4.5: *The transfer graph representing reads with substitutions (see [14, Page 9]).*

The transfer graph in Figure 4.5 depicts walks that represent sequences of single substitutions and error-free intervals. The symbols Δ_0 and S denote the error-free intervals and the single substitution respectively, and $F(z)$ and pz represent their respective weighted generating functions. Using Definition 4.3, the correct nucleotides were merged together in a single error-free interval,

$$F(z) = qz + (qz)^2 + (qz)^3 + ... = \frac{qz}{1-qz}. \tag{4.2}$$

The transfer matrix connected with the transfer graph in Figure 4.5 is represented as

$$M_*(z) = \begin{array}{c} \\ \circ \\ \Delta_0 \\ S \\ \bullet \end{array} \begin{array}{c} \begin{array}{cccc} \circ & \Delta_0 & S & \bullet \end{array} \\ \begin{bmatrix} 0 & F(z) & pz & 1 \\ 0 & 0 & pz & 1 \\ 0 & F(z) & pz & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

where each entry in the matrix represents an edge in the transfer graph, Figure 4.5. For example, $F(z)$ on the first row second column of the matrix, $M_*(z)$, represents the edge from the head vertex $\circ$ to the error-free interval Δ_0.

Then, we obtain the head vector with the outgoing edges as $(F(z), pz) = H(z)$, the tail vector with the incoming edges as $T(z) = (1,1)^T$ and $\psi(z) = 1$ which is the edge from the head vertex to the tail vertex (see Theorem 3.1).

$$M(z) = \begin{array}{c} \\ \Delta_0 \\ S \end{array} \begin{array}{c} \begin{array}{cc} \Delta_0 & S \end{array} \\ \begin{pmatrix} 0 & pz \\ F(z) & pz \end{pmatrix} \end{array}.$$

The subgraph of $M_*(z)$ is denoted as $M(z)$ above. It is a 2×2 matrix representation of the edges of the vertices of the subgraph in Figure 4.9.

Now, by applying Theorem 3.1, the weighted generating function of error-free intervals is given by

$$\psi(z) + H(z) \cdot (I - M(z))^{-1} \cdot T(z) = R(z). \tag{4.3}$$

We obtained the solution of $(I - M(z))^{-1}$ using Mathematica, first by calculating $|I - M(z)|$ which is $1 - pz(F(z) + 1)$, then,

$$(I - M(z))^{-1} = \frac{1}{1 - pz(F(z) + 1)} \begin{bmatrix} 1 - pz & pz \\ F(z) & 1 \end{bmatrix}.$$

Now substituting $(I - M(z))^{-1}$ into Equation (4.3),

$$\begin{aligned}
R(z) &= 1 + (F(z), pz) \cdot \frac{1}{1 - pz(F(z) + 1)} \begin{bmatrix} 1 - pz & pz \\ F(z) & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= 1 + (F(z), pz) \begin{pmatrix} \frac{1-pz}{1-pz(F(z)+1)} & \frac{pz}{1-pz(F(z)+1)} \\ \frac{F(z)}{1-pz(F(z)+1)} & \frac{1}{1-pz(F(z)+1)} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= 1 + (F(z), pz) \begin{pmatrix} \frac{1}{1-pz(F(z)+1)} \\ \frac{1+F(z)}{1-pz(F(z)+1)} \end{pmatrix} \\
&= 1 + \frac{F(z)}{1 - pz(F(z) + 1)} + \frac{p(z)(1 + F(z))}{1 - pz(F(z) + 1)} \\
&= \frac{1 + F(z)}{1 - pz(F(z) + 1)},
\end{aligned}$$

and by substituting Equation (4.2) into the $R(z)$ above, we have

$$R(z) = \frac{1 + \frac{qz}{1-qz}}{1 - pz\frac{qz}{1-qz} - pz} = \frac{1}{1 - (p + q)z} = \frac{1}{1 - z}. \tag{4.4}$$

Note: $p + q = 1$. So, $[z^k]R(z) = 1$ for $k \geq 0$.

Remark. The Equation (4.4) $R(z)$ is a check and the result is not used here but $F(z)$ in $R(z)$ is replaced by $F_\gamma(z)$ to obtain $S_\gamma(z)$.

Let

$$F_\gamma(z) = \frac{qz(1-(qz)^{\gamma-1})}{1-qz}, \tag{4.5}$$

where $\gamma - 1$ is the maximum size of error-free interval and this is done in order to obtain $S_\gamma(z)$. We substitute $F_\gamma(z)$ in $R(z)$ in place of $F(z)$ to obtain

$$S_\gamma(z) = \frac{1+F_\gamma(z)}{1-pz(1+F_\gamma(z))} = \frac{1+qz+...+(qz)^{\gamma-1}}{1-pz(1+qz+...+(qz)^{\gamma-1})}. \tag{4.6}$$

We then apply Theorem 3.2 to the $S_\gamma(z)$, to get the coefficient of z^k in $S(z)$.

Proposition 4.1. Under substitution model, $P(S)$ is asymptotic to

$$\frac{\beta}{z_1^{k+1}},$$

where

$$\beta = \frac{(1-qz_1)^2}{p^2 z_1(1-\gamma+1-\gamma q z_1)(qz_1)^\gamma},$$

and z_1 is the dominant singularity of $1-pz(1+F_\gamma(z))=0$ [14].

Remark.

$$P(S) = \frac{[z^k]S_\gamma(z)}{[z^k]R(z)} = \frac{[z^k]S_\gamma(z)}{1} \sim \frac{\beta}{z_1^{k+1}}.$$

Proof. We obtain β using the Theorem 3.2 by equating Equation (4.6) to

$$\frac{P(z)}{Q(z)} = \frac{1+F_\gamma(z)}{1-pz(1+F_\gamma(z))} = \frac{1+q(z)+(qz)^2+...+(qz)^{\gamma-1}}{1-pz(1+qz+(qz)^2+...+(qz)^{\gamma-1}},$$

and then equating $Q(z)$ to 0, so we have

$$\begin{aligned} 1-pz(1+F_\gamma(z)) &= 0 \\ 1+F_\gamma(z) &= \frac{1}{pz} \\ F_\gamma(z) &= \frac{1}{pz} - 1. \end{aligned} \tag{4.7}$$

Now we substitute Equation (4.7) into $P(z)$ and set $z = z_1$ to obtain $P(z_1)$

$$P(z_1) = 1 + \frac{1}{pz_1} - 1 = \frac{1}{pz_1}. \tag{4.8}$$

By differentiating $Q(z)$ and setting $z = z_1$ we have

$$Q'(z_1) = -p - p(F_\gamma(z_1)) - pz(F'_\gamma(z_1)). \tag{4.9}$$

We substitute $F_\gamma(z) = \frac{qz(1-(qz)^{\gamma-1})}{1-qz}$ and $F'_\gamma(z) = \frac{qz_1(qz_1)^\gamma - qz_1 + \gamma(qz_1)^\gamma - q\gamma z_1(qz_1)^\gamma}{z_1(1-qz_1)^2}$ into Equation (4.9) to obtain

$$Q'(z_1) = -\frac{p(1-(qz_1)^\gamma(\gamma+1-\gamma qz_1))}{1-(qz_1)^2}.$$

So we have the constant term β in Equation (3.6) in Chapter 3 to be,

$$\begin{aligned}
\beta = -\frac{P(z_1)}{Q'(z_1)} &= -P(z_1) \cdot -\frac{(1-qz_1)^2}{p(1-(qz_1)^\gamma(\gamma+1-\gamma qz))} \\
&= -\frac{1}{pz_1} \cdot -\frac{(1-qz_1)^2}{p(1-(qz_1)^\gamma(\gamma+1-\gamma qz)))} \\
&= \frac{(1-qz_1)^2}{p^2z_1(1-(\gamma+1-\gamma qz_1)(qz)^\gamma)}.
\end{aligned} \tag{4.10}$$

Therefore the expression

$$\frac{(1-qz_1)^2}{p^2z_1(1-(\gamma+1-\gamma qz_1)(qz)^\gamma)} \cdot \frac{1}{z_1^{k+1}} = \frac{\beta}{z_1^{k+1}} \tag{4.11}$$

is asymptotic to the coefficient of z^k in $S_\gamma(z)$, since $[z^k]R(z) = 1$, for any $k \geq 0$, see Equation (4.1).

□

In [9], it is possible to have (25, 50 and 100) base pairs single-end reads which could be used for analysis instead of longer reads. For the purpose of this research, the examples that will be considered for obtaining the required probability under the different error models will include reads of sizes (25, 50, 100 and 200). These examples will be solved using analytic

combinatorial approach here in Chapter 4 and the recurrence approach in Chapter 5. The choice of values for k (read size) and p (probability at which a substitution error occur) in the examples used in the dissertation are to demonstrate how the model works. Below is an example demonstrating how $P(S)$ is obtained.

Example 4.1. Find the probability that a read of size $k = 25, 50, 100, 200$ contains no seed given that $\gamma = 17$ and the probability that a single substitution occurs is $p = 0.12$.

Solution. We obtain the dominant singularity of $S_{17}(z)$ by equating $Q(z)$ in Equation (4.6) to 0 and rewrite the expression $1 + qz + (qz)^2 + ... + q(z)^{\gamma-1}$ as $\frac{1-(qz)^{\gamma}}{1-qz}$. Then we substitute the values $p, q = 1 - p = 0.88$ and γ into $Q(z)$. We solve

$$1 - 0.12z \times \frac{(1 - (0.88z))^{17}}{1 - 0.88z} = 0$$

by using the inbuilt bisection function in Python library to obtain the dominant singularity $z_1 \approx 1.019251$. (see Appendix A.1.1). Then we find that $\beta \approx 1.276608$ using Equation (4.10) and now we obtain $P(S)$ when: $k = 25$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.276608}{(1.019251)^{25+1}} \\ &\sim 0.777585;\end{aligned}$$

$k = 50$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.276608}{(1.019251)^{50+1}} \\ &\sim 0.482746;\end{aligned}$$

$k = 100$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.276608}{(1.019251)^{100+1}} \\ &\sim 0.186063;\end{aligned}$$

and $k = 200$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.276608}{(1.019251)^{200+1}} \\ &\sim 0.027640.\end{aligned}$$

Remark. We calculated the $P(S)$ at $k = 25, 50, 100, 200$, given that $\gamma = 17$ and $p = 0.12$ and based on these probabilities, we deduce that there is an high probability that the read of size $(k = 100, 200)$ has a seed $1 - P(S)$ when only substitution errors occur in the read. For small read sizes $(k = 25, 50)$, the $(1 - P(S)$ is very small. When $1 - P(s)$ is high, the regions of high similarity are found faster during sequence alignment.

4.4 Substitution and deletion

We consider an error model where substitutions and deletions occur during alignments. We assume that some of the nucleotides from the genome sequence (stored in a long molecule of DNA) are deleted with a probability δ (see Figure 4.2). Depending on the amount of nucleotides deleted in Figure 4.3, all deletions are regarded equal [14]. Consecutive sequences of deleted nucleotides are treated as a single deletion which occurs with probability δ.

There are two possibilities of deletions occurring during alignments;

1. the deletion takes place between two correct nucleotides, and

2. a deletion occurring before or after a substitution (see Figure 4.2).

Remark. When a deletion occurs before or after substitution, the error-free interval is not interrupted, and thus $1 - P(S)$ is unaffected. We do not consider situations where deletions occur before or after substitutions because they have no direct effect on the $P(S)$.

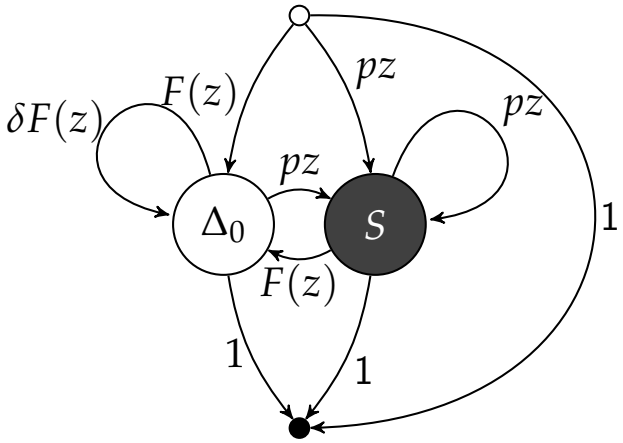


FIGURE 4.6: *The transfer graph representing the reads with substitutions and deletions [14, Page 12].*

We represent the read as a sequence of edges and vertices in Figure 4.6. In Figure 4.6, an edge is labelled $\delta F(z)$, and this edge is a result of having a deletion between two error-free intervals. The weighted generating function of substitution is pz. The weighted generating functions of a deletion and an error-free interval of size 1 are denoted by δ and qz respectively. The weighted generating function of an error-free interval of size $k > 1$ is $(1-\delta)^{k-1}(qz)^k$ since there are $k-1$ "gaps" between the nucleotides. We note that k is at least one, so the $k < 1$ is not relevant here. The weighted generating function of all error-free intervals denoted as $F(z)$ is obtained by summing up all the weighted generating functions of all sizes $(k = 1, k > 1)$ below

$$F(z) = qz + (1-\delta)(qz)^2 + (1-\delta)^2(qz)^3 + (1-\delta)^3(qz)^4 + ... = \frac{qz}{1-(1-\delta)qz}. \tag{4.12}$$

We represent the edges on the transfer graph on the transfer matrix below

$$M_*(z) = \begin{array}{c} \\ \circ \\ \Delta_0 \\ S \\ \bullet \end{array} \begin{array}{c} \begin{array}{cccc} \circ & \Delta_0 & S & \bullet \end{array} \\ \begin{bmatrix} 0 & F(z) & pz & 1 \\ 0 & \delta F(z) & pz & 1 \\ 0 & F(z) & pz & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} .$$

Then $H(z) = (F(z), pz)$ is the head vector; $T(z) = (1,1)^T$ the tail vector; and $\psi(z) = 1$ represents the empty sequence ϵ with size 0. The edges on the subgraph (all other vertices except the head and tail vertices and the directed edges between them) in Figure 4.6 are denoted as

$$M(z) = \begin{array}{c} \\ \Delta_0 \\ S \end{array} \begin{array}{c} \begin{array}{cc} \Delta_0 & S \end{array} \\ \begin{pmatrix} \delta F(z) & pz \\ F(z) & pz \end{pmatrix} \end{array} .$$

Now, we apply Theorem 3.1. The weighted generating function that corresponds to the reads represented in Figure 4.6 is

$$H(z) \cdot (I - M(z))^{-1} \cdot T(z) + \psi(z) = R(z) \ .$$

We solve for $(1 - M(z))^{-1}$ by following the steps below;

$$I - M(z) = \begin{bmatrix} 1 - \delta F(z) & -pz \\ -F(z) & 1 - pz \end{bmatrix},$$

and

$$|I - M(z)| = 1 - pz - (pz(1-\delta) + \delta)F(z) \ .$$

Then,

$$(I - M(z))^{-1} = \frac{1}{|I - M(z)|} \cdot \begin{bmatrix} 1 - pz & pz \\ F(z) & 1 - \delta F(z) \end{bmatrix}.$$

Now we substitute $(I - M(z))^{-1}$ into Equation (3.2),

$$\begin{aligned} R(z) &= 1 + (F(z), pz) \cdot \frac{1}{1 - pz - \delta F(z) - F(z)pz + \delta F(z)pz} \begin{bmatrix} 1 - pz & pz \\ F(z) & 1 - \delta F(z) \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= 1 + (F(z), pz) \begin{pmatrix} \frac{1-pz}{1-pz-\delta F(z)-F(z)pz+\delta F(z)pz} & \frac{pz}{1-pz-\delta F(z)-F(z)pz+\delta F(z)pz} \\ \frac{F(z)}{1-pz-\delta F(z)-F(z)pz+\delta F(z)pz} & \frac{1-\delta F(z)}{1-pz-\delta F(z)-F(z)pz+\delta F(z)pz} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= 1 + (F(z), pz) \begin{pmatrix} \frac{1}{1-pz-\delta F(z)-F(z)pz+\delta F(z)pz} \\ \frac{1+F(z)-\delta F(z)}{1-pz-\delta F(z)-F(z)pz+\delta F(z)pz} \end{pmatrix} \\ &= 1 + \frac{F(z)}{1 - pz - \delta F(z) - F(z)pz + \delta F(z)pz} + \frac{p(z)(1 + F(z) - \delta F(z))}{1 - pz - \delta F(z) - F(z)pz + \delta F(z)pz} \\ &= \frac{1 + F(z)(1 - \delta)}{1 - pz - \delta F(z) - F(z)pz + \delta F(z)pz}. \end{aligned} \tag{4.13}$$

Remark. By substituting Equation (4.12) into Equation (4.13), we do a check to obtain the result $\frac{1}{1-z}$, which implies that the $[z^k]R(z)$ is 1 for any $k \geq 0$.

$$\begin{aligned} R(z) &= \frac{1 + \frac{qz}{1-(1-\delta)qz}(1 - \delta)}{1 - pz\frac{qz}{(1-(1-\delta)qz} - pz + \delta\frac{qz}{(1-(1-\delta)qz}pz - \delta\frac{qz}{(1-(1-\delta)qz}} \\ &= \frac{1}{1 - (1 - \delta)qz - pqz^2 - pz(1 - (1 - \delta)qz) + \delta pqz^2 - \delta qz} \\ &= \frac{1}{1 - z}. \end{aligned} \tag{4.14}$$

We replace $F(z)$ in Figure 4.12 by

$$F_\gamma(z) = \frac{qz - (1 - \delta)^{\gamma-1}(qz)^\gamma}{(1 - (1 - \delta)qz)}, \tag{4.15}$$

where $\gamma - 1$ is the maximum size of error-free interval and this is done in order to obtain $S_\gamma(z)$. So,

$$S_\gamma(z) = \frac{1 + F_\gamma(z)(1-\delta)}{1 - pz - (pz(1-\delta) + \delta)F_\gamma(z)pz}. \tag{4.16}$$

We then apply Theorem 3.2 to the $S_\gamma(z)$, to get the coefficient of z^k in $S_\gamma(z)$.

Proposition 4.2. Under the substitution and deletion error model, $P(S)$ is asymptotic to

$$\frac{\beta}{z_1^{k+1}}, \tag{4.17}$$

where

$$\beta = \frac{z_1(1-(1-\delta)qz_1)^2}{(\delta + (1-\delta)pz_1)((p+q\delta)z_1)\lambda((1-\delta)^{\gamma-1}(qz_1)^\gamma))},$$

and z_1 is the dominant singularity of $1 - pz - (pz(1-\delta) + \delta)F_\gamma(z) = 0$, where $F_\gamma(z)$ is given in Equation (4.15) [14].

Proof. □

We equate $S_\gamma(z)$ to

$$\frac{P(z)}{Q(z)} := \frac{1 + F_\gamma(z)(1-\delta)}{1 - pz - (pz(1-\delta) + \delta)F_\gamma(z)},$$

where $P(z)$ and $Q(z)$ are polynomials and we equate $Q(z)$ to 0 to obtain

$$F_\gamma(z) = \frac{1 - pz}{\delta + pz - \delta pz}, \tag{4.18}$$

which is the point where we have our singularity. Now we obtain $Q'(z_1)$ by differentiating $Q(z)$ and setting $z = z_1$. We differentiate

$$Q(z) = 1 - pz - (\delta + pz - \delta pz)F_\gamma(z),$$

to obtain

$$Q'(z_1) = -p - (p(1-\delta))F_\gamma(z_1) - (pz_1(1-\delta)+\delta)F'_\gamma(z_1).$$

We substitute the truncated $F(z)$ from Equation (4.15) and its differentiation

$$F'_\gamma(z_1) = \frac{q + (qz_1(\gamma-1)(1-\delta)-\gamma)((1-\delta)qz_1)^{\gamma-1}}{(1-(1-\delta)qz_1)^2},$$

into $Q'(z_1)$ to get

$$Q'(z_1) = \frac{(p+q\delta)z_1 - \lambda(1-\delta)^{\gamma-1}(qz_1)^\gamma}{z_1(1-(1-\delta)qz_1)^2},$$

where

$$\lambda = \gamma\delta - (1-\delta)(-p-\delta q - p\gamma + \delta q\gamma)z_1 - \gamma(1-\delta)^2 pqz_1^2. \tag{4.19}$$

Now we substitute Equation (4.18) into $P(z)$ and let $z = z_1$ to obtain $P(z_1)$

$$\begin{aligned} P(z_1) &= 1 + (1-\delta)\left[\frac{1-pz_1}{\delta + pz_1 - \delta pz_1}\right] \\ &= \frac{\delta + pz_1 - \delta pz_1 + (1-\delta)(1-pz_1)}{\delta + pz_1 - \delta pz_1} \\ &= \frac{1}{\delta + pz_1(1-\delta)}. \end{aligned}$$

So we have the constant term β as

$$\begin{aligned} \beta = \frac{-P(z_1)}{Q'(z_1)} &= -\frac{1}{\delta + pz_1(1-\delta)} \cdot \frac{1}{Q'(z_1)} \\ &= \frac{z_1(1-(1-\delta)qz_1)^2}{(\delta+(1-\delta)pz_1)((p+q\delta)z_1)\lambda((1-\delta)^{\gamma-1}(qz_1)^\gamma))}, \end{aligned} \tag{4.20}$$

where λ is given in Equation (4.19). So we have

$$P(S) = \frac{[z^k]S_\gamma(z)}{[z^k]R(z)} \sim \frac{z_1(1-(1-\delta)qz_1)^2}{(\delta+(1-\delta)pz_1)((p+q\delta)z_1)\lambda((1-\delta)^{\gamma-1}(qz_1)^\gamma))} \cdot \frac{1}{z_1^{k+1}}.$$

Recall that $[z^k]R(z) = 1$ (see Equation 4.14).

Example 4.2. Find the probability that a read of size $k = 25, 50, 100, 200$ contains no seed given that $\gamma = 17,\ p = 0.05,\ \delta = 0.16$.

Solution. Given that $p = 0.05$, then $q = (1-p) = 0.95$. We obtain the dominant singularity of $S_{17}(z)$ by substituting the values given into $Q(z)$.

$$0 = 1 - 0.05z - (0.05z - (0.16)(0.05z) + 0.16)(0.95z + 0.84(0.95z)^2 + ... + 0.84^{15}(0.95)^{16}). \tag{4.21}$$

We rewrite $0.95z + 0.84(0.95z)^2 + ... + 0.84^{15}(0.95)^{16}$ in Equation (4.21) as

$$\left[\frac{0.95z - 0.84^{16}(0.95)^{17}}{1 - ((0.84)(0.95)z)}\right],$$

(see Equation (4.15)) to obtain

$$0 = 1 - 0.05z - (0.042z + 0.16)\left[\frac{0.95z - 0.84^{16}(0.95)^{17}}{1 - (0.798z)}\right].$$

We use the inbuilt numerical bisection function in Python to obtain $z_1 \approx 1.005725$ (see Appendix A.1.2) and by substituting the value of z_1 in Equation (4.20) we obtain the constant $\beta \approx 1.082240$ which we use in calculating $P(S)$. When $k = 25$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.082240}{(1.005725)^{26}} \\ &\sim 0.932960;\end{aligned}$$

$k = 50$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.082240}{(1.005725)^{51}} \\ &\sim 0.808876;\end{aligned}$$

When $k = 100$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.082240}{(1.005725)^{101}} \\ &\sim 0.608023;\end{aligned}$$

when $k = 200$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.082240}{(1.005725)^{201}} \\ &\sim 0.343554.\end{aligned}$$

Remark. We obtained the probabilities that reads of sizes 25, 50, 100 and 200 have no seed $(P(S))$ in the example above. From the values obtained, we were able to calculate the probability that the read has a seed $(1 - P(S))$. For a read of size $k = 50$ and $k = 200$, $1 - P(S)$ equals 0.191124 and 0.656446 respectively. The higher the number of reads, the higher the probability that there is a seed. The error model also influences the probabilities obtained (comparing the probabilities obtained when substitution and deletion occur to the probabilities obtained when only substitution occurs).

4.5 Substitutions, Deletions and Insertions

We have discussed scenarios when substitution errors occur and when substitution and deletion errors occur, and now we will go over the most difficult scenario: when substitution, deletion, and insertion errors occur in a read. When three different types of errors occur during sequencing, the transfer graph for alignment process is more complex.

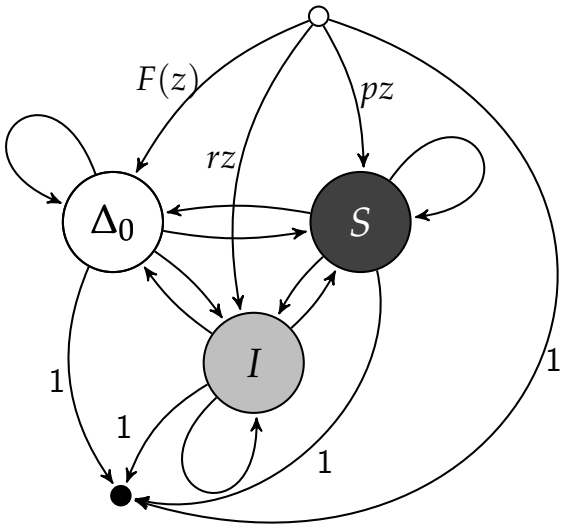


FIGURE 4.7: *The graph reflects the transfer graph of the substitutions, deletions, and insertions error model.*

Remark. Some of the edges in the graph above were not labeled to make it easier to understand and to keep the graph from becoming too complicated.

The "full error model" refers to the occurrence of all the errors (substitutions, deletions and insertions) [14]. We note the following before proceeding.

1. We cannot determine the difference between substitution and insertion when deletion occurs immediately before or after insertion. So, in a situation where deletion occurs immediately before or after an insertion, the deletion is regarded as a substitution.

2. When insertions occur multiple times consecutively, it is called a *burst*, and we indicate r as probability of having the first insertion and $\tilde{r} > r$ as the probability of having other insertions occurring.

Remark. All deletions that do not occur before or after an insertion can be ignored since they have no impact on the size of the error-free interval (see Figure 4.3). Note that deletions have size $s = 0$ [14].

The probability that a substitution error occurs is denoted by p, the probability that the correct nucleotide occurs is q and the probability that an insertion error occurs is r. These probabilities sum to 1 $(p + q + r = 1)$. We make an assumption that the probability at which an insertion burst stops is $1 - \tilde{r}$ at every point the burst occurs except the first one.

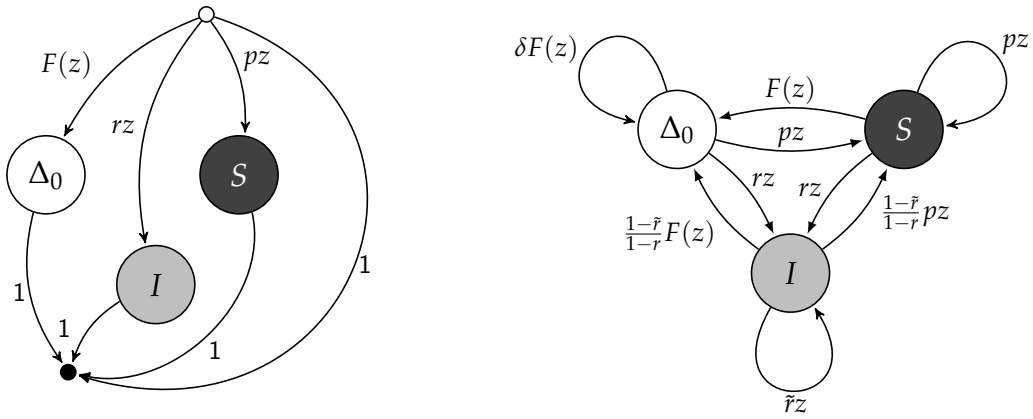


FIGURE 4.8: *The graphs depict the substitutions, deletions, and insertions transfer graph, with the subgraph (B) detached from (A) [14, Page 14].*

The vertices Δ_0, S and I are the error-free interval, single substitution, and single insertions respectively. The edges labeled in (B), Figure 4.8 are explained below.

1. The edges labeled by $F(z)$ and pz are the weighted generating functions of the error-free interval and single substitution respectively. We have edge $F(z)$ when the arrow points to the error-free interval either from the head vertex $(\circ)$ or the single substitution and pz when the arrow points to substitution either from the error-free interval or from the head vertex $(\circ)$ or when there is an edge that maps vertex S to itself (loop).

2. The edge labeled $\delta F(z)$ is called a loop because it maps the vertex Δ_0 to itself and it is the weighted generating function of when a deletion occurs between two error-free intervals (recall that deletion does not have an impact on error-free intervals).

3. The probability that an insertion burst terminates is $1-\tilde{r}$, and once it does, an error-free interval or substitution occurs after it. Since the total weight of an error-free interval or substitution occurring after an insertion burst ends is $p+q<1$, the weighted generating function must be reduced by a factor of $p+q=1-r$.

   When an error-free interval occurs immediately after an insertion, the weighted generating function is $\frac{1-\tilde{r}}{1-r}F(z)$, and when a substitution error occurs immediately after an insertion, the weighted generating function is $\frac{1-\tilde{r}}{1-r}pz$.

4. The first insertion nucleotide and each subsequent nucleotide of an insertion burst have their weighted generating functions as rz and $\tilde{r}z$ respectively. The $\tilde{r}z$ labels the loop with the arrow mapping I to itself.

5. The edges which point to the tail vertex ($\bullet$) from all other vertices Δ, S, I and $\circ$ are denoted by 1 because their sizes do not have an effect on the weighted generating function.

Using the combinatorial techniques outlined in Chapter 3, we can now calculate $P(S)$. Recall that $F(z)$ is the error-free interval's weighted generating function and can be expressed as

$$F(z)=qz+(1-\delta)(qz)^2+(1-\delta)^2(qz)^3+(1-\delta)^3(qz)^4+\ldots=\frac{qz}{1-(1-\delta)qz}. \tag{4.22}$$

We represent the labels of the edges of the transfer graph on the transfer matrix below

$$M_*(z)=\begin{array}{c} \\ \circ \\ \Delta_0 \\ S \\ I \\ \bullet \end{array}\begin{array}{c} \begin{array}{ccccc} \circ & \Delta_0 & S & I & \bullet \end{array} \\ \begin{bmatrix} 0 & F(z) & pz & rz & 1 \\ 0 & \delta F(z) & pz & rz & 1 \\ 0 & F(z) & pz & rz & 1 \\ 0 & \frac{1-\tilde{r}}{1-r}F(z) & \frac{1-\tilde{r}}{1-r}pz & \tilde{r}z & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

Then, the head vector is $H(z) = (F(z), pz, rz)$; the tail vector is $T(z) = (1,1,1)^T$ and $\psi(z) = 1$ represents the empty sequence ϵ with size 0.

The edges on the subgraph (B) (except the edges on the head and tail vertices) represented in Figure 4.8 are denoted as

$$M(z) = \begin{array}{c} \\ \Delta_0 \\ S \\ I \end{array} \begin{array}{c} \begin{array}{ccc} \Delta_0 & S & I \end{array} \\ \begin{pmatrix} \delta F(z) & pz & rz \\ F(z) & pz & rz \\ \frac{1-\tilde{r}}{1-r}F(z) & \frac{1-\tilde{r}}{1-r}pz & \tilde{r}z \end{pmatrix} \end{array}.$$

For example, the entries in the first column of the matrix, $(\delta F(z), pz$ and $rz)$ represent the edges from vertex Δ_0 to Δ_0 and Δ_0 to S and Δ_0 to I respectively. Now, we apply Theorem 3.1. The weighted generating function that corresponds to the reads represented in Figure 4.8 is

$$\psi(z) + H(z) \cdot (I - M(z))^{-1} \cdot T(z) = R(z).$$

We solve for $(I - M(z))^{-1}$ by doing some matrix multiplications using Mathematica and substitute it into the R(z) above to obtain

$$R(z) = \frac{(1-r)(1-\tilde{r}z+rz)(1+(1-\delta)F(z))}{1-\mu(z)-\nu(z)F(z)}, \tag{4.23}$$

where $\mu(z)$ and $\nu(z)$ are polynomials of degree two written below as

$$\begin{aligned} \mu(z) &= r + (1-r)(p+\tilde{r})z - p(\tilde{r}-r)z^2, \text{ and} \\ \nu(z) &= \delta(1-r) + ((1-r)(p-\delta(p+\tilde{r})) + (1-\tilde{r})r)z - p(1-\delta)(\tilde{r}-r)z^2. \end{aligned} \tag{4.24}$$

Remark. Then we substitute $\mu(z)$, $\nu(z)$ and $F(z)$ into Equation (4.23) and simplify to get,

$$R(z) = \frac{1}{1-z},$$

which serves as a check to confirm that $[z^k]R(z)$ is 1 for $k \geq 0$.

A read which has size $s = k$ with no seed has a probability of $[z^k]S_\gamma(z)$. We replace $F(z)$ in Equation (4.22) with

$$F_\gamma(z) = \frac{qz - (1-\delta)^{\gamma-1}(qz)^\gamma}{(1-(1-\delta)qz)}, \tag{4.25}$$

where $\gamma - 1$ is the maximum size of error-free interval. The use of $F_\gamma(z)$ in place of $F(z)$ is to obtain

$$S_\gamma(z) = \frac{(1-r)(1-\tilde{r}z+rz)(1+(1-\delta)F_\gamma(z))}{1-\mu(z)-\nu(z)F_\gamma(z)}, \tag{4.26}$$

where $\mu(z)$ and $\nu(z)$ are the same as in Equation (4.24).

Remark. 1. Equation (4.26) will be

$$S_\gamma(z) = \frac{1+F_\gamma(z)(1-\delta)}{1-pz-(pz(1-\delta)+\delta)F_\gamma(z)pz},$$

when $\tilde{r} = r = 0$, then $\mu(z) = pz$ and $\nu(z) = pz(1-\delta)+\delta$, which is the $S_\gamma(z)$ when only substitution and deletion errors occur (see Equation (4.16)).

2. In addition, $\delta = 0$ in Equation (4.26), it simplifies to

$$S_\gamma(z) = \frac{1+F_\gamma(z)}{1-pz(1+F_\gamma(z))},$$

which is the $S_\gamma(z)$ when only substitution errors occur (see Equation (4.6)).

We then apply Theorem 3.2 to the $S_\gamma(z)$ in Equation (4.26) to get the coefficient of z^k in $S_\gamma(z)$.

Proposition 4.3. The $P(S)$ under the substitution, insertion and deletion error model is asymptotic to

$$\frac{\beta}{z_1^{k+1}}, \tag{4.27}$$

where

$$\beta = \frac{(1-r)(1-(\tilde{r}-r)z_1)(1+(1-\delta)F_\gamma(z_1))}{\mu'(z_1)+\nu'(z_1)F_\gamma(z_1)+\nu(z_1)F'_\gamma(z_1)},$$

and z_1 is the dominant singularity of the denominator in Equation (4.26). In addition, $1-\mu(z)-\nu(z)F_\gamma(z)=0$ given that $\mu(z)$ and $\nu(z)$ is the same as in Equation (4.24).

Proof. We equate $S_\gamma(z)$ to

$$\frac{P(z)}{Q(z)} := \frac{(1-r)(1-\tilde{r}z+rz)(1+(1-\delta)F_\gamma(z))}{1-\mu(z)-\nu(z)F_\gamma(z)}.$$

Now we obtain $Q'(z_1)$ by differentiating $Q(z)$ and replacing z by z_1. Again we make use of the truncated $F(z)$, which we call $F_\gamma(z)$ from Equation (4.25) to get

$$Q(z) = 1-\mu(z)-\nu(z)F_\gamma(z),$$

and hence

$$Q'(z_1) = -\mu'(z_1)-\nu'(z)F_\gamma(z_1)-\nu(z)F'_\gamma(z_1).$$

We set $z=z_1$ to obtain

$$P(z_1) = (1-r)(1-(\tilde{r}-r)z_1)(1+(1-\delta)F_\gamma(z_1)) \ .$$

Hence we have the constant term β as

$$\beta = \frac{-P(z_1)}{Q'(z_1)} = -\frac{(1-r)(1-(\tilde{r}-r)z_1)(1+(1-\delta)F_\gamma(z_1))}{Q'(z_1)},$$
$$\beta = \frac{(1-r)(1-\tilde{r}z_1+rz_1)(1+(1-\delta)F_\gamma(z_1))}{\mu'(z_1)+\nu'(z_1)F_\gamma(z_1)+\nu(z_1)F'_\gamma(z_1)}, \tag{4.28}$$

where

$$F_\gamma(z_1) = \frac{qz_1-(1-\delta)^{\gamma-1}(qz_1)^\gamma}{1-(1-\delta)qz_1},$$

and

$$F'_\gamma(z_1) = \frac{q+((1-d)(qz_1))^{\gamma-1)}((1-d)(\gamma-1)qz_1-\gamma)}{(1-(1-\delta)qz_1)^2}.$$

Then we have

$$P(S) = \frac{[z^k]S_\gamma(z)}{[z^k]R(z)} \sim \frac{(1-r)(1-\tilde{r}z_1+rz_1)(1+(1-\delta)F_\gamma(z_1))}{\mu'(z_1)+\nu'(z_1)F_\gamma(z_1)+\nu(z_1)F'_\gamma(z_1)} \cdot \frac{1}{z_1^{k+1}}.$$

□

Example 4.3. Find the probability that a read of size $k = 25, 50, 100, 200$ contains no seed given that $\gamma = 17$, $\delta = 0.15$, $p = 0.05$, $r = 0.06$ and $\tilde{r} = 0.45$.

Solution. By substituting the values given into Equation (4.24), we have

$$\mu(z) = 0.06 + 0.47z - 0.0195z^2, \quad \text{and}$$
$$\nu(z) = 0.141 + 0.0095z - 0.016575z^2.$$

We obtain the dominant singularity of $S_{17}(z)$ by substituting the $\mu(z)$, $\nu(z)$ and q into $Q(z)$ and equating it to 0. Note $q = 1-(p+r)$, then we have

$$\begin{aligned} 0 = \quad & 1 - (0.06 + 0.47z - 0.0195z^2) - (0.141 + 0.0095z - 0.016575z^2)\cdot \\ & (0.89z + 0.85(0.89z)^2 + ... + 0.85^{15}(0.89)^{16}). \end{aligned}$$

We rewrite $0.89z + 0.85(0.89z)^2 + ... + 0.85^{15}(0.89)^{16}$ as

$$\frac{0.89z - 0.85^{16}(0.89z)^{17}}{1-((0.85)(0.89)z)}$$

and use the inbuilt numerical bisection function in Python library to obtain the dominant singularity $z_1 \approx 1.002498$ (see Appendix A.1.3). By substituting the value of z_1 into Equation (4.20), we obtain the constant $\beta \approx 1.03601$ which we use in calculating $P(S)$ when $k = 25$,

$$\begin{aligned} [z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.03601}{(1.002498)^{26}} \\ &\sim 0.970940; \end{aligned}$$

$k = 50$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.03601}{(1.002498)^{51}} \\ &\sim 0.912230;\end{aligned}$$

When $k = 100$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.03601}{(1.002498)^{101}} \\ &\sim 0.805247;\end{aligned}$$

When $k = 200$,

$$\begin{aligned}[z^k]S_{17}(z) &\sim \frac{\beta}{z_1^{k+1}} \\ &\sim \frac{1.03601}{(1.002498)^{201}} \\ &\sim 0.627448.\end{aligned}$$

Remark. We obtained the $P(S)$ for reads of sizes 25, 50, 100 and 200 in the example above. From the values obtained, as the size of read increase $(1 - P(S))$ increases (see Appendix A.2.3). One thing to also note here is that when the three errors (substitution, deletion and insertion) occur together, the probability that the read sizes have a seed of length γ is smaller compared to the when only two (substitution and deletion) or one (substitution) error occur.

4.5.1 Approximation

With the calculations done in the examples given under each of the error models (substitution only; substitution and deletion; and substitution, deletion and insertion), we were able to tell that the approximations obtained using analytic combinatorics are accurate with long reads. The error models involved also influences the probability obtained.

Theorem 3.2 aimed at helping to obtain estimates faster. Approximated values can be used in place of the true value if a scenario is created to check the output in the worst conditions. In situations where there is a longer read (larger size), approximations are more accurate and there are fewer errors. In the alignment, the accuracy of approximations cannot be improved due to the fact that the size of the read is determined by the technology used in sequencing the reads. When the dominant singularity (singularity with the smallest modulus) is used in computing the estimates, the estimates converge faster compared to when other singularities are used. Approximated values obtained using the analytic combinatoric approach are less accurate in scenarios when

1. when reads are small in size,

2. when the singularities have very close moduli.

In the situation where the read is of size 50 or less, coefficients are obtained by recurrence to obtain better accuracy and to reduce approximation error. When computing the s_k value using the recurrence method, we obtain the exact $P(S)$ for the three error model considered in this research.

Chapter 5

Recurrence Approach

In this chapter, we used a different approach than the one used in Filion's paper, [14], to calculate the probability that a read of size k has a seed. We obtained the probability using the recurrence approach and compared the solution with the probability obtained using the analytic combinatorics approach.

5.1 Recurrence approach (substitution error model)

As mentioned in Chapter 3, the probability that a read of size k has a seed can be calculated by a recurrence approach using Theorem 3.3. We rewrite Equation (4.6) as

$$\begin{aligned} S_\gamma(z) = \frac{P(z)}{Q(z)} &:= \frac{1+qz+\ldots+(qz)^{\gamma-1}}{1-pz(1+qz+\ldots+(qz)^{\gamma-1})} \\ &= \frac{\frac{1-(qz)^\gamma}{1-qz}}{1-pz\left(\frac{1-(qz)^\gamma}{1-qz}\right)} \\ &= \frac{1-(qz)^\gamma}{1-qz-pz(1-(qz)^\gamma)} \quad \text{where,} \quad p+q=1 \\ &= \frac{1-(qz)^\gamma}{1-z+pq^\gamma z^{\gamma+1}}. \end{aligned} \tag{5.1}$$

By applying Theorem 3.3, we have

$$P(z) = Q(z)S_\gamma(z), \tag{5.2}$$

where $P(z)$ and $Q(z)$ are the same as in Equation (5.1) and

$$S_\gamma(z) = s_0 + s_1 z + s_2 z^2 + s_3 z^3 + ... + s_{\gamma-1} z^{\gamma-1} + s_\gamma z^\gamma + s_{\gamma+1} z^{\gamma+1} + ...$$

Now we substitute the expressions for $P(z), Q(z)$ and $S_\gamma(z)$ into Equation (5.2) to get

$$1 - q^\gamma z^\gamma = [1 - z + pq^\gamma z^{\gamma+1}][s_0 + s_1 z + s_2 z^2 + s_3 z^3 + ... + s_\gamma z^\gamma + s_{\gamma+1} z^{\gamma+1} + ...]. \tag{5.3}$$

We expand and compare the coefficients of z^k on the left and right side of Equation (5.3) and obtain solutions s_k for when k is $\{0, 1, 2, ..., \gamma, > \gamma\}$. We obtain $s_k = 1$ for the values of $k = 0, 1, ..., \gamma - 1$ as follows

$$\begin{aligned}
[z^0] &: 1 = 1 \cdot s_0 \implies s_0 = 1 \\
[z^1] &: 0 = 1 \cdot s_1 - 1 \cdot s_0 \implies s_1 = s_0 = 1 \\
[z^2] &: 0 = 1 \cdot s_2 - 1 \cdot s_1 \implies s_2 = s_1 = 1 \\
&\vdots \\
[z^{\gamma-1}] &: 0 = 1 \cdot s_{\gamma-1} - 1 \cdot s_{\gamma-2} \implies s_{\gamma-1} = 1 \\
[z^\gamma] &: -q^\gamma = 1 \cdot s_\gamma - 1 \cdot s_{\gamma-1}
\end{aligned}$$

where $s_{\gamma-1} = 1$, then

$$s_\gamma = 1 - q^\gamma.$$

Now for $k = \gamma + 1$,

$$[z^{\gamma+1}] : 0 = s_0 pq^\gamma - s_\gamma + s_{\gamma+1}$$

and hence

$$s_{\gamma+1} = s_\gamma - s_0 pq^\gamma. \tag{5.4}$$

So for any $k > \gamma$,

$$s_k = s_{k-1} + pq^\gamma \cdot s_{k-(\gamma+1)},$$

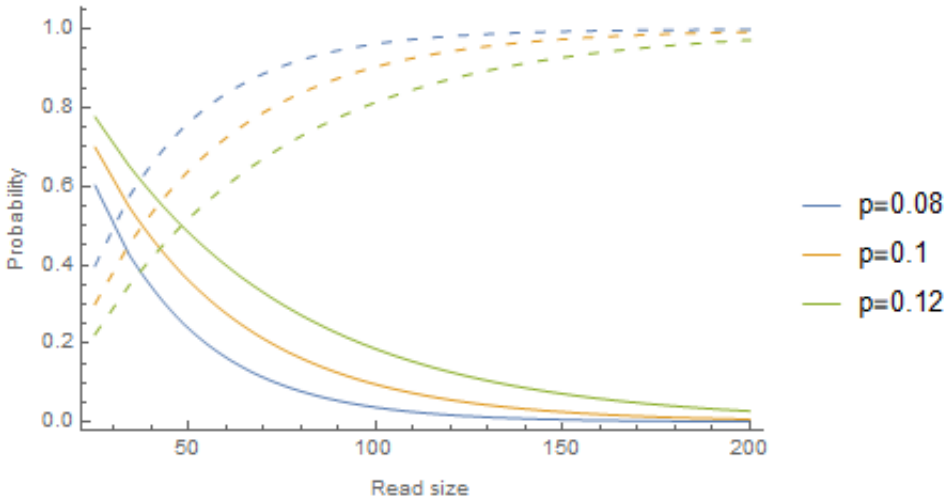


FIGURE 5.1: *The solid lines represent $P(S)$ while the dashed lines represent $1-P(S)$. The y-axis represents the probability plotted against the read size on the x-axis at $p = 0.08,\ 0.1$ and 0.12.*

where $k \geq \gamma + 1$ in Equation (5.4). Therefore, in summary we have

$$s_k = \begin{cases} 1, & 0 \leq k < \gamma, \\ 1 - q^{\gamma}, & k = \gamma, \\ s_{k-1} - pq^{\gamma} \cdot s_{k-\gamma-1} & k > \gamma. \end{cases} \tag{5.5}$$

Figure 5.1 was plotted by substituting the values from Example 4.1 into the recurrence relation, Equation (5.5), using one p value at a time.

Remark. Using Mathematica (see Appendix A.2.1), we inserted the values given in Example 4.1 in the recurrence and plotted the graph in Figure 5.1.

5.2 Recurrence approach (substitution and deletion error model)

Now we apply Theorem 3.3 to obtain $P(S)$. We substitute Equation (4.15) into Equation (4.16) below

$$\begin{aligned} S_\gamma(z) = \frac{P(z)}{Q(z)} &:= \frac{1+\frac{qz-(1-\delta)^{\gamma-1}(qz)^\gamma}{(1-(1-\delta)qz)}(1-\delta)}{1-pz-(\delta-pz+\delta pz)\frac{qz-(1-\delta)^{\gamma-1}(qz)^\gamma}{(1-(1-\delta)qz)}} \\ &= \frac{1-(1-\delta)^\gamma(qz)^\gamma}{((1-(p+q)z+(\delta+pz-\delta pz)((1-\delta)^{\gamma-1}(qz)^\gamma))} \\ &= \frac{1-(1-\delta)^\gamma(qz)^\gamma}{((1-z+(\delta+pz-\delta pz)((1-\delta)^{\gamma-1}(qz)^\gamma))}. \end{aligned}$$

We denote $[z^k]S_\gamma(z)$ as s_k and obtain the s_k for all k by applying Theorem 3.3,

$$P(z) = Q(z)S_\gamma(z). \tag{5.6}$$

We write Equation (5.6) as follows;

$$1-(1-\delta)^\gamma(qz)^\gamma = (1-z+(\delta+pz-\delta pz)(1-\delta)^{\gamma-1}(qz)^\gamma)(S_\gamma(z)) \ ,$$

where $S_\gamma(z) = s_0 + s_1z + s_2z^2 + s_3z^3 + ... + s_{\gamma-1}z^{\gamma-1} + s_\gamma z^\gamma + s_{\gamma+1}z^{\gamma+1} + ...$.
So,

$$\begin{aligned} 1-(1-\delta)^\gamma(qz)^\gamma = (1-z+\delta(1-\delta)^{\gamma-1}(qz)^\gamma + pq^\gamma(1-\delta)^\gamma z^{\gamma+1})\cdot \\ (s_0 + s_1z + s_2z^2 + s_3z^3 + ... + s_{\gamma-1}z^{\gamma-1} + s_\gamma z^\gamma + s_{\gamma+1}z^{\gamma+1} + ...). \end{aligned} \tag{5.7}$$

We expand and compare the coefficients of z^k on the left and right side of Equation (5.7) and obtain solutions s_k for when k is $\{0, 1, 2, ..., \gamma - 1, \gamma, > \gamma\}$ as follows

$$\begin{aligned}
[z^0] &: 1 = 1 \cdot s_0 \implies s_0 = 1 \\
[z^1] &: 0 = 1 \cdot s_1 - 1 \cdot s_0 \implies s_1 = s_0 = 1 \\
[z^2] &: 0 = 1 \cdot s_2 - 1 \cdot s_1 \implies s_2 = s_1 = 1 \\
&\vdots \\
[z^{\gamma-1}] &: 0 = 1 \cdot s_{\gamma-1} - 1 \cdot s_{\gamma-2} \implies s_{\gamma-1} = 1 \\
[z^{\gamma}] &: -(1-\delta)^{\gamma} q^{\gamma} = 1 \cdot s_{\gamma} - 1 \cdot s_{\gamma-1} + s_0 \delta (1-\delta)^{\gamma-1} q^{\gamma}
\end{aligned}$$

where $s_{\gamma-1} = 1$ and $s_0 = 1$, so

$$\begin{aligned}
s_{\gamma} &= 1 - \delta(1-\delta)^{\gamma-1} q^{\gamma} - (1-\delta)^{\gamma} q^{\gamma} \\
&= 1 - (1-\delta)^{\gamma-1} q^{\gamma} (\delta + (1-\delta)) \\
&= 1 - (1-\delta)^{\gamma-1} q^{\gamma}.
\end{aligned}$$

Now the coefficient of z^{γ} in Equation (5.7) is

$$0 = -s_{\gamma} + s_{\gamma+1} + s_1 \delta (1-\delta)^{\gamma-1} q^{\gamma} + s_0 p q^{\gamma} (1-\delta)^{\gamma},$$

and

$$s_{\gamma+1} = s_{\gamma} - s_1 \delta (1-\delta)^{\gamma-1} q^{\gamma} - s_0 p q^{\gamma} (1-\delta)^{\gamma}.$$

For any $k > \gamma$,

$$s_k = s_{k-1} - s_{k-\gamma} \delta (1-\delta)^{\gamma-1} q^{\gamma} - p q^{\gamma} (1-\delta)^{\gamma} s_{k-(\gamma+1)}.$$

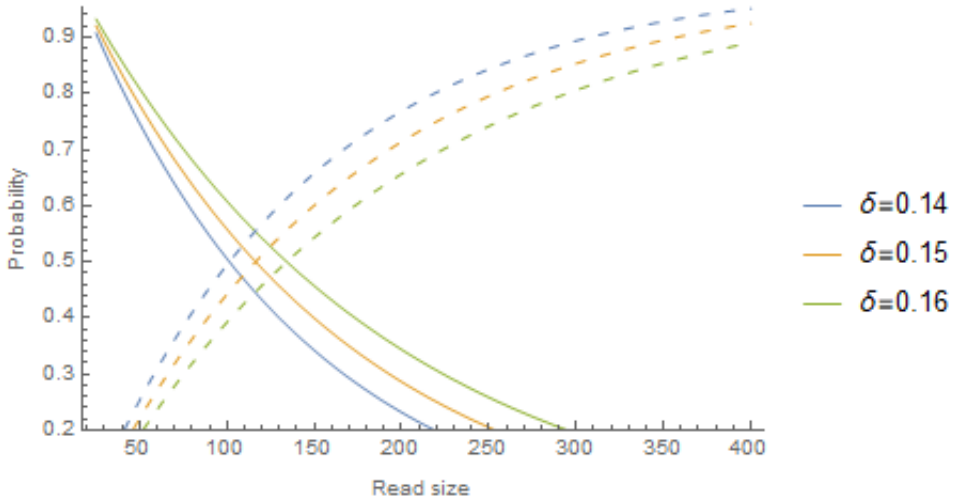


FIGURE 5.2: *The solid lines represent $P(S)$ while the dashed lines represent $1-P(S)$. The y-axis represents the probability plotted against the read size on the x-axis at $\delta = 0.14,\ 0.15$ and 0.16.*

In summary, we have

$$s_k = \begin{cases} 1, & 0 \leq k < \gamma, \\ 1-(1-\delta)^{\gamma-1}q^{\gamma}, & k = \gamma, \\ s_{k-1} - \delta(1-\delta)^{\gamma-1}q^{\gamma} \cdot s_{k-\gamma} - pq^{\gamma}(1-\delta)^{\gamma} \cdot s_{k-\gamma-1} & k > \gamma. \end{cases} \tag{5.8}$$

Figure 5.2 was plotted by substituting the values from Example 4.2 into the recurrence relation, Equation (5.8), using one δ value at a time.

Remark. Using Mathematica (see Appendix A.2.2), we inserted the values given in Example 4.2 and two other δ's ($\delta = 0.14,\ 0.15$) in the recurrence and plotted the graphs in Figure 5.2. In plotting the graph, the values of p and q remain the same for each of the δ given.

5.3 Recurrence approach (substitution, deletion and insertion error model)

Now we obtain $P(S)$ by using a linear recurrence and applying Theorem 3.3. We substitute Equation (4.25) into Equation (4.26) below;

$$S_\gamma(z) = \frac{P(z)}{Q(z)} := \frac{(1-r)(1-(\tilde{r}-r)z))(1+(1-\delta)\frac{qz-(1-\delta)^{\gamma-1}(qz)^\gamma}{(1-(1-\delta)qz)})}{1-\mu(z)-\nu(z)\frac{qz-(1-\delta)^{\gamma-1}(qz)^\gamma)}{(1-(1-\delta)qz)}}.$$

We denote $[z^k]S_\gamma(z)$ as s_k to obtain every $k > 1$ and by applying Theorem 3.3, we have

$$P(z) = Q(z)S_\gamma(z), \tag{5.9}$$

where $S_\gamma(z) = \sum_{n\geq 0} s_n z^n$. Then

$$\begin{aligned} P(z) &= \frac{(1-r)(1-(\tilde{r}-r)z)(1-(1-\delta)qz+(1-\delta)qz-(1-\delta)^\gamma(qz)^\gamma}{1-(1-\delta)qz} \\ &= \frac{(1-r)(1-(\tilde{r}-r)z)(1-((1-\delta)qz)^\gamma)}{1-(1-\delta)qz} \\ &= \frac{(1-r)-(1-r)(\tilde{r}-r)z-(1-r)((1-\delta)qz)^\gamma+(1-r)(\tilde{r}-r)((1-\delta)q)^\gamma z^{\gamma+1}}{1-(1-\delta)qz}. \end{aligned}$$

We rewrite $P(z)$ in terms of p_i

$$P(z) = p_0 + p_1 z + p_\gamma z^\gamma + p_{\gamma+1} z^{\gamma+1}.$$

We express $Q(z)$ as

$$\begin{aligned} Q(z) &= 1-\mu(z)-\nu(z)\frac{qz(1-((1-\delta)qz)^{\gamma-1}}{1-(1-\delta)qz} \\ &= \frac{1-(1-\delta)qz-(\mu(z))(1-(1-\delta)qz)-(\nu(z))(qz(1-((1-\delta)qz)^{\gamma-1})}{1-(1-\delta)qz}, \end{aligned}$$

where we denote $\mu(z) = \mu_0 + \mu_1 z + \mu_2 z^2$ and $\nu(z) = \nu_0 + \nu_1 z + \nu_2 z^2$. We expand and arrange the numerator of $Q(z)$ as

$$(1 - \mu_0 + (-(1-\delta)q - \mu_1 + \mu_0(1-\delta)q - \nu_0 q)z + (-\mu_2 + \mu_1(1-\delta)q - \nu_1 q)z^2 + (\mu_2(1-\delta)q - \nu_2 q)z^3 + \nu_0(1-\delta)^{\gamma-1}q^\gamma z^\gamma + \nu_1(1-\delta)^{\gamma-1}q^\gamma z^{\gamma+1} + \nu_2(1-\delta)^{\gamma-1}q^\gamma z^{\gamma+2}).$$

Then we write $Q(z)$ in terms of q_i

$$Q(z) = q_0 + q_1 z_1 + q_2 z^2 + q_3 z^3 + q_\gamma z^\gamma + q_{\gamma+1} z^{\gamma+1} + q_{\gamma+2} z^{\gamma+2}.$$

We have Equation (5.9) to be

$$p_0 + p_1 z + p_\gamma z^\gamma + p_{\gamma+1} z^{\gamma+1} = (q_0 + q_1 z_1 + q_2 z^2 + q_3 z^3 + q_\gamma z^\gamma + q_{\gamma+1} z^{\gamma+1} + q_{\gamma+2} z^{\gamma+2}) \cdot (s_0 + s_1 z + s_2 z^2 + s_3 z^3 + ... + s_{\gamma-1} z^{\gamma-1} + s_\gamma z^\gamma + s_{\gamma+1} z^{\gamma+1} + ...). \tag{5.10}$$

By comparing coefficients of z^k in Equation (5.10), we obtain solutions for when $0 \leq k < \gamma$.
For $k = 0$, we compute s_0 by considering Equation (5.10), we have

$$p_0 = s_0 q_0$$

and substituting p_0 and q_0 we have

$$(1 - r) = s_0(1 - r)$$
$$s_0 = 1.$$

For the case $k = 1$, we compare the coefficients of z^1 in Equation (5.10), thus we obtain

$$p_1 = s_0 q_1 + s_1 q_0.$$

Substituting,

$$\begin{aligned}-(1-r)(\tilde{r}-r) &= s_0(-(1-\delta)q-\mu_1+\mu_0(1-\delta)q-\nu_0 q)+s_1(1-\mu_0)\\ &= (-(1-\delta)q-(1-r)(p+\tilde{r})+r(1-\delta)q-\delta(1-r)q+s_1(1-r).\end{aligned}$$

Dividing by $1-r$ on both sides, yields

$$-(\tilde{r}-r) = (-p-q-\tilde{r})+s_1,$$

since $p+q+r=1$. Consequently,

$$-\tilde{r}+r+1-r+\tilde{r} = s_1,$$

and hence, $s_1 = 1$.

For $k=2$, the coefficients of z^2 in Equation (5.10) are

$$p_2 = s_0 q_2 + s_1 q_1 + s_2 q_0.$$

Now we substitute the terms, expand and simplify as follows;

$$\begin{aligned}0 &= s_0(-\mu_2+\mu_1(1-\delta)q-\nu_1 q)+s_1(-(1-\delta)q-\mu_1+\mu_0(1-\delta)q-\nu_0 q)+s_2(1-\mu_0)\\ &= p(\tilde{r}-r)+(1-r)(p+\tilde{r})(1-\delta)q-((1-r)(p-\delta(p+\tilde{r}))+(1-\tilde{r})r)q+\\ &\quad (-(1-\delta)q-(1-r)(p+\tilde{r})+r(1-\delta)q-\delta(1-r)q)+s_2(1-r)\\ &= p(\tilde{r}-r)+q(\tilde{r}-r)+(1-r)(-q-p-\tilde{r}+s_2).\end{aligned}$$

Since $p+q+r=1$, we have

$$\begin{aligned}0 &= (1-r)(\tilde{r}-r-1+r-\tilde{r}+s_2)\\ &= (1-r)(-1+s_2)\end{aligned}$$

and hence $s_2 = 1$.

For $k = 3$, the coefficients of z^3 in Equation (5.10) are

$$0 = s_0 q_3 + s_1 q_2 + s_2 q_1 + s_3 q_0.$$

We substitute the terms, expand and simplify as follows;

$$\begin{aligned}
0 &= s_0(\mu_2(1-\delta)q - \nu_2 q) + s_1(-\mu_2 + \mu_1(1-\delta)q - \nu_1 q) + s_2(-(1-\delta)q - \mu_1 + \\
&\quad \mu_0(1-\delta)q - \nu_0 q) + s_3(1-\mu_0) \\
&= (-(p(\tilde{r}-r)(1-\delta)q + (p(1-\delta)(\tilde{r}-r)q) + (p(\tilde{r}-r) + (1-r)(p+\tilde{r})(1-\delta)q \\
&\quad - ((1-r)(p-\delta(p+\tilde{r})) + (1-\tilde{r})r)q) + (-(1-\delta)q - (1-r)(p+\tilde{r}) + r(1-\delta)q - \\
&\quad \delta(1-r)q) + s_3(1-r).
\end{aligned}$$

Hence, $s_3 = 1$.

The computations for $\{s_4, \ldots, s_{\gamma-3}, s_{\gamma-2}, s_{\gamma-1}\}$ are similar to the earlier computations of s_k where $\{k = 1,2,3\}$, so we omit them and proceed to obtain s_γ. Then for z^γ, we have

$$p_\gamma = s_0 q_\gamma + s_\gamma q_0 + s_{\gamma-1} q_1 + s_{\gamma-2} q_2 + s_{\gamma-3} q_3,$$

where $s_{\gamma-1}$, $s_{\gamma-2}$, $s_{\gamma-3}$ equal 1. We now substitute the terms needed and simplify to get

$$\begin{aligned}
-(1-r)((1-\delta)q)^\gamma &= \nu_0(1-\delta)^{\gamma-1}q^\gamma + s_\gamma(1-\mu_0) + s_{\gamma-1}(1-\delta)q - \mu_1 + \mu_0(1-\delta)q - \\
&\quad \nu_0 q) + s_{\gamma-2}(-\mu_2 + \mu_1(1-\delta)q - \nu_1 q) + s_{\gamma-3}(\mu_2(1-\delta)q - \nu_2 q) \\
&= \delta(1-r)((1-\delta)^{\gamma-1}q^\gamma) + s_\gamma(1-r) + (-(1-\delta)q - (1-r)(p+\tilde{r}) + \\
&\quad r(1-\delta)q - \delta(1-r)q) + p(\tilde{r}-r) + (1-r)(p+\tilde{r})(1-\delta)q - \\
&\quad ((1-r)(p-\delta(p+\tilde{r})) + (1-\tilde{r})r)q \\
&= (1-r)[\delta((1-\delta)^{\gamma-1}q^\gamma) + s_\gamma + -q - p - \tilde{r} + \tilde{r} - r],
\end{aligned}$$

then we divide both sides by $(1-r)$ to obtain

$$-(1-\delta)^{\gamma}q^{\gamma} = \delta((1-\delta^{\gamma-1}q^{\gamma}) + s_{\gamma} - q - p - \tilde{r} + \tilde{r} - r,$$

and where $p+q+r=1$,

$$\begin{aligned} s_{\gamma} &= -(1-\delta)^{\gamma}q^{\gamma} - \delta(1-\delta^{\gamma-1}q^{\gamma}) + 1 \\ s_{\gamma} &= 1 - (1-\delta)^{\gamma-1}q^{\gamma}. \end{aligned}$$

For $k = \gamma + 1$, we compare the coefficients in Equation (5.10) and obtain

$$p_{y+1} = s_0 q_{y+1} + s_1 q_y + s_y q_1 + s_{y-1} q_2 + s_{y-2} q_3 + s_{\gamma+1} q_0.$$

We substitute in the terms required and simplify

$$\begin{aligned} (1-r)(\tilde{r}-r)((1-\delta)q)^{\gamma} &= \nu_1((1-\delta)^{\gamma-1}q^{\gamma}) + \nu_0(1-\delta)^{\gamma-1}q^{\gamma} + s_{\gamma-2}(\mu_2(1-\delta)q - \nu_2 q) + \\ &\quad s_{\gamma}(-(1-\delta)q - \mu_1 + \mu_0(1-\delta)q - \nu_0 q) + s_{\gamma-1}(-\mu_2 + \mu_1(1-\delta)q - \\ &\quad \nu_1 q) + s_{\gamma+1}(1-r) \\ &= ((1-r)(p-\delta(p+\tilde{r})) + (1-\tilde{r})r)(1-\delta)^{\gamma-1}q^{\gamma}) + \\ &\quad \delta(1-r)(1-\delta)^{\gamma-1}q^{\gamma}) + s_{\gamma}(-(1-\delta)q - (1-r)(p+\tilde{r}) + r(1-\delta)q - \\ &\quad \delta(1-r)q) + (p(\tilde{r}-r) + (1-r)(p+\tilde{r})(1-\delta)q - \\ &\quad ((1-r)(p-\delta(p+\tilde{r})) + (1-\tilde{r})r)q + s_{\gamma+1}(1-r), \end{aligned}$$

we divide both sides by $1-r$ and have

$$\begin{aligned} (\tilde{r}-r)((1-\delta)q)^{\gamma} &= (p - \delta(p+\tilde{r}) + \frac{(1-\tilde{r})r}{1-r}((1-\delta)^{\gamma-1}q^{\gamma}) + \delta((1-\delta)^{\gamma-1}q^{\gamma}) + \\ &\quad (1-(1-\delta)^{\gamma-1}q^{\gamma})(-q-p-\tilde{r}) + (\tilde{r}-r) + s_{\gamma+1}. \end{aligned}$$

Now make $s_{\gamma+1}$ the subject of the formula to obtain

$$s_{\gamma+1} = (\tilde{r}-r)((1-\delta)q)^{\gamma} - (p-\delta(p+\tilde{r}) + \frac{(1-\tilde{r})r}{1-r}((1-\delta)^{\gamma-1}q^{\gamma}) - \delta((1-\delta)^{\gamma-1}q^{\gamma}) - (1-(1-\delta)^{\gamma-1}q^{\gamma})(-q-p-\tilde{r}) - (\tilde{r}-r).$$

Factorise and apply the expression $p+q+r=1$ to simplify

$$\begin{aligned} s_{\gamma+1} &= -((1-\delta)^{\gamma-1}q^{\gamma})\Big[-(\tilde{r}-r)(1-\delta) + \Big(p-\delta(p+\tilde{r}) + \frac{(1-\tilde{r})r}{1-r}\Big) + \delta + 1 - r + \tilde{r}\Big] + 1 \\ &= 1 - ((1-\delta)^{\gamma-1}q^{\gamma})\Big[p+\delta q + \frac{(1-\tilde{r})r}{1-r} + 1\Big] \\ &= 1 - ((1-\delta)^{\gamma-1}q^{\gamma})\Big[p+\delta q + \frac{1-\tilde{r}r}{1-r}\Big]. \end{aligned}$$

When $k=\gamma+2$, the coefficients of $z^{\gamma+2}$ in Equation (5.10) are

$$0 = s_0q_{y+2} + s_1q_{y+1} + s_2q_{\gamma} + s_{\gamma}q_2 + s_{y-1}q_3 + s_{y+1}q_1 + s_{y+2}q_0.$$

We substitute the known terms and simplify as follows;

$$\begin{aligned} 0 &= s_0\nu_2((1-\delta)^{\gamma-1}q^{\gamma} + s_1\nu_1((1-\delta)^{\gamma-1}q^{\gamma}) + s_2\nu_0(1-\delta)^{\gamma-1}q^{\gamma} + s_{\gamma+2}(1-\mu_0) + \\ &\quad s_{\gamma}(-\mu_2 + \mu_1(1-\delta)q - \nu_1q) + s_{\gamma+1}(-(1-\delta)q - \mu_1 + \mu_0(1-\delta)q - \nu_0q) \\ &= s_0(-p(1-\delta)(\tilde{r}-r))((1-\delta)^{\gamma-1}q^{\gamma}) + s_1(1-r)(p-\delta(p+\tilde{r}) + \frac{(1-\tilde{r})r}{1-r})((1-\delta)^{\gamma-1}q^{\gamma}) \\ &\quad + s_2(1-r)\delta((1-\delta)^{\gamma-1}q^{\gamma}) + s_{\gamma}(1-r)(\tilde{r}-r) + s_{\gamma+1}(1-r)(-q-p-\tilde{r}) + s_{\gamma+2}(1-r). \end{aligned}$$

We divide both sides by $1-r$ and simplify

$$\begin{aligned} 0 &= s_0(-p(1-\delta)^{\gamma}q^{\gamma})\frac{\tilde{r}-r}{1-r} + s_1\Big(p-\delta(p+\tilde{r}) + \frac{(1-\tilde{r})r}{1-r}\Big)((1-\delta)^{\gamma-1}q^{\gamma}) \\ &\quad + s_2\delta((1-\delta)^{\gamma-1}q^{\gamma}) + s_{\gamma}(\tilde{r}-r) + s_{\gamma+1}(-q-p-\tilde{r}) + s_{\gamma+2}. \end{aligned}$$

Note that $p+q+r=1$, so

$$s_{\gamma+2} = s_0 p(1-\delta)^\gamma q^\gamma\Big(\frac{\tilde{r}-r}{1-r}\Big) - s_1\Big(p-\delta(p+\tilde{r})+\frac{(1-\tilde{r})r}{1-r}\Big)((1-\delta)^{\gamma-1}q^\gamma)$$
$$- s_2\delta((1-\delta)^{\gamma-1}q^\gamma) - s_\gamma(\tilde{r}-r) - s_{\gamma+1}(-1+r-\tilde{r}).$$

In general, when $k > \gamma+1$, we have

$$s_k = p(1-\delta)^\gamma q^\gamma\Big(\frac{\tilde{r}-r}{1-r}\Big)\cdot s_{k-\gamma-2} - \Big(p-\delta(p+\tilde{r})+\frac{(1-\tilde{r})r}{1-r}\Big)((1-\delta)^{\gamma-1}q^\gamma)\cdot s_{k-\gamma-1}$$
$$- \delta((1-\delta)^{\gamma-1}q^\gamma)\cdot s_{k-\gamma} - (\tilde{r}-r)s_{k-2} + s_{k-1}(1+\tilde{r}-r).$$

In summary, we can express s_k as follows:

$$s_k = \begin{cases} 1, & 0 \le k < \gamma, \\ 1-(1-\delta)^{\gamma-1}q^\gamma, & k=\gamma, \\ 1-((1-\delta)^{\gamma-1}q^\gamma)\Big(p+\delta q+\frac{1-\tilde{r}r}{1-r}\Big), & k=\gamma+1, \\ (1+\tilde{r}-r)\cdot s_{k-1} - (\tilde{r}-r)\cdot s_{k-2} & \\ -\delta(1-\delta)^{\gamma-1}q^\gamma\cdot s_{k-\gamma} & \\ +\Big(-p+\delta(p+\tilde{r})-\frac{(1-\tilde{r})r}{1-r}\Big)(1-\delta)^{\gamma-1}q^\gamma\cdot s_{k-\gamma-1} & \\ +pq^\gamma(1-\delta)^\gamma\frac{\tilde{r}-r}{1-r}\cdot s_{k-\gamma-2}, & k>\gamma+1. \end{cases} \tag{5.11}$$

Note that Equation (5.11) is slightly different from the original text ([14]) because a typo was spotted and corrected here.

Remark. Using Mathematica (see Appendix A.2.3), we inserted the values given in Example 4.3 and two other r's ($r = 0.04,\ 0.05$) in the recurrence and plotted the graphs in Figure 5.3. In plotting the graph, the values of p remain the same for each r given, but for each of the given r, there is a corresponding q obtained from the formula $p+q+r=1$.

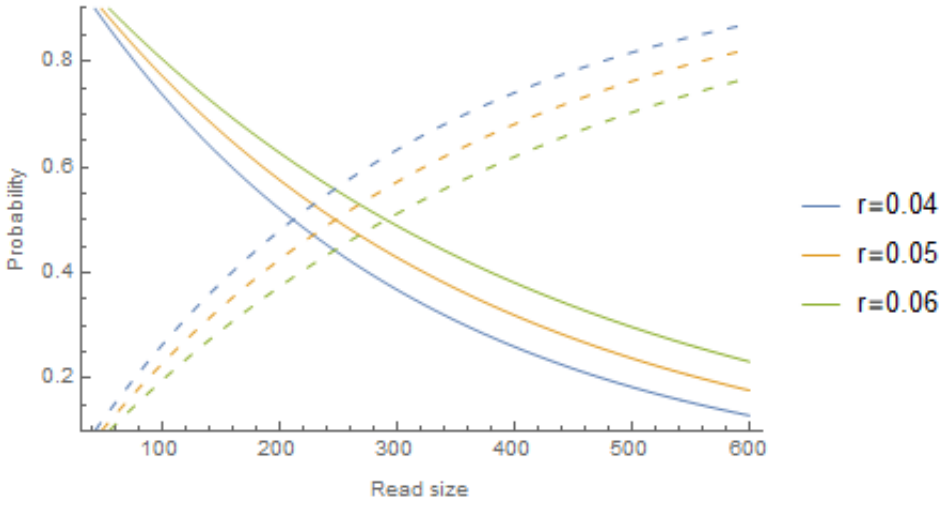


FIGURE 5.3: *The solid lines represent* $P(S)$ *while the dashed lines represent* $1 - P(S)$. *The* y*-axis represents the probability plotted against the read size on the* x*-axis at* $r = 0.04,\ 0.05$ *and* 0.06.

Chapter 6

Conclusion

In this dissertation, we focused on the alignment of DNA sequences. We studied the paper "Analytic combinatorics for computing seeding probabilities" by Filion [14] to find the similarities between sequences (using exact seeds) in order to align the sequences faster. We employed some mathematical tools (analytic combinatorics) in obtaining the probability that a read of size k has a seed $(1 - P(S))$ under the different error models (substitution only; substitution and deletion; and substitution, deletion and insertion). We described these sequences of symbols in terms of weighted generating functions and represented them on a transfer graph. The weighted generating functions, represented as edges in the transfer graph, are converted into a transfer matrix to get the weights of the sequences of symbols referred to as reads (error-free intervals) using Theorem 3.1. This led us to obtain the weights of reads with no seed $(S_\gamma(z))$. We found the probability that a read of size k has no seed $(P(S))$ under three error models by taking the proportion of weights of reads with no seed over all reads. This involved extracting coefficients of weighted generating functions where we employed the use of singularity analysis in Theorem 3.2.

Filion mentioned in his paper, [14], that the probability computations for short reads are not exact using the analytic combinatorics approach but are better computed using the recurrence approach. This motivated us to build on Filion's work by delving into the recurrence method extensively and address the computations for short reads (read sizes 25 and 50). We elaborated on using the recurrence method (with the help of Mathematica) to obtain the probabilities required for reads of length 25, 50, 100, and 200. We obtained the recurrence relation (s_k) by equating $S_\gamma(z)$ to the ratio of two polynomials and compared

the coefficients on either side of the equation to obtain $P(S)$. We used the same values as the examples considered in the analytic combinatorics approach in the recurrence relation approach.

We compared the probabilities obtained from the two approaches (see Appendix A.2) and plotted the graphs for the probabilities $P(S)$ and $1 - P(S)$ against the read sizes under different error models. We concluded that the recurrence approach and the analytic combinatorial approach gave similar results. We also deduced from the graphs plotted that $1 - P(S)$ increases monotonically with read size. The smaller the read size, the lesser the accuracy of the probability obtained using the analytic combinatorial approach. The result presented in [14] helps to streamline the search for similarities between sequences by locating seeds (exact) in reads thereby speeding up sequence alignment.

In [14], Filion focused on obtaining the probability that a read has no exact seed (the basic kind of seeds) and did not cover how the probability that a read has no seeds for other kind of seeds (transition or score seed, see [31]). This creates an opportunity for further research on obtaining the probability that a read has no specific seed type. For this research we assume that the only differences that occur during alignment of sequences are as a result of sequencing errors. Further studies can be done where genetic differences between sequences are taken into consideration and not sequencing errors only.

Appendix A

Appendix

A.1 Numerical bisection using Python

A.1.1 Substitution only, Example 4.1

```
def f(x):
    answer = 1-0.12*x*((1-(0.88*x)**17 ) / (1-0.88*x))
    return answer

# We must specify limits a and b in the arguments list
# so the method can find the root somewhere in between them.
bisect_root = optimize.bisect(f, a=1, b=2)
# Print out the root. Also print out the value of f at the root,
# which should be zero if the root has been found accurately.
z= print(f"The root of the function f(x) is: {bisect_root:.6f}. At
                                    this point, f(x) = {f(bisect_root)
                                    :.6f}")

The root of the function f(x) is: 1.019251. At this point, f(x) = 0.
                                    000000
```

A.1.2 Substitution and deletion only, Example 4.2

```
def f(x):
    answer = 1-0.05*x-((0.042*x+0.16)*(0.95*x-((0.84**16)*(0.95*x)**17
                                        )) / (1-0.798*x))
```

```
    return answer

# We must specify limits a and b in the arguments list
# so the method can find the root somewhere in between them.
bisect_root = optimize.bisect(f, a=1, b=2)
# Print out the root. Also print out the value of f at the root,
# which should be zero if the root has been found accurately.
z= print(f"The root of the function f(x) is: {bisect_root:.6f}. At
                                    this point, f(x) = {f(bisect_root)
                                    :.6f}")

The root of the function f(x) is: 1.005725. At this point, f(x) = 0.
                                    000000
z=1.005725
```

A.1.3 Substitution, deletion, insertion, Example 4.2

```
def f(x):
    answer = (0.94-0.47*x+0.0195*x**2)-(((0.141+0.0095*x-(0.016575*x**
                                        2))*(0.89*x-((0.85**16)*(0.89*x
                                        )**17 ))) / (1-0.7565*x))
    return answer

# We must specify limits a and b in the arguments list
# so the method can find the root somewhere in between them.
bisect_root = optimize.bisect(f, a=1, b=2)
# Print out the root. Also print out the value of f at the root,
# which should be zero if the root has been found accurately.
z= print(f"The root of the function f(x) is: {bisect_root:.6f}. At
                                    this point, f(x) = {f(bisect_root)
                                    :.6f}")

The root of the function f(x) is: 1.002498. At this point, f(x) = 0.
                                    000000
```

A.2 Recurrence Plots

A.2.1 Recurrence plots for substitution only

In[ ]:=

```
Plot1 = With[{p1 = 0.08, q1 = 0.92, p2 = 0.1, q2 = 0.9, p3 = 0.12, q3 = 0.88}, ListLinePlot[
   Transpose[RecurrenceTable[{G[k] == G[k - 1] - p1 q1^17 G[k - 18], G[0] == 1, G[1] == 1,
      G[2] == 1, G[3] == 1, G[4] == 1, G[5] == 1, G[6] == 1, G[7] == 1, G[8] == 1, G[9] == 1,
      G[10] == 1, G[11] == 1, G[12] == 1, G[13] == 1, G[14] == 1, G[15] == 1, G[16] == 1,
      G[17] == 1 - q1^17, T[k] == T[k - 1] - p2 q2^17 T[k - 18], T[0] == 1, T[1] == 1, T[2] == 1,
      T[3] == 1, T[4] == 1, T[5] == 1, T[6] == 1, T[7] == 1, T[8] == 1, T[9] == 1, T[10] == 1,
      T[11] == 1, T[12] == 1, T[13] == 1, T[14] == 1, T[15] == 1, T[16] == 1, T[17] == 1 - q2^17,
      U[k] == U[k - 1] - p3 q3^17 U[k - 18], U[0] == 1, U[1] == 1, U[2] == 1, U[3] == 1, U[4] == 1,
      U[5] == 1, U[6] == 1, U[7] == 1, U[8] == 1, U[9] == 1, U[10] == 1, U[11] == 1, U[12] == 1,
      U[13] == 1, U[14] == 1, U[15] == 1, U[16] == 1, U[17] == 1 - q3^17}, {G, T, U}, {k, 25, 200}]],
   DataRange → {25, 200}, PlotRange → All, PlotStyle → {Medium, Medium, Medium},
   PlotLegends → {"p=0.08", "p=0.1", "p=0.12"}]]
```

In[ ]:=

```
Plot2 = With[{p1 = 0.08, q1 = 0.92, p2 = 0.1, q2 = 0.9, p3 = 0.12, q3 = 0.88 },
  ListLinePlot[
   1 -
    (Transpose[RecurrenceTable[{G[k] == G[k - 1] - p1 q1^17 G[k - 18], G[0] == 1, G[1] == 1,
       G[2] == 1, G[3] == 1, G[4] == 1, G[5] == 1, G[6] == 1, G[7] == 1, G[8] == 1, G[9] == 1,
       G[10] == 1, G[11] == 1, G[12] == 1, G[13] == 1, G[14] == 1, G[15] == 1, G[16] == 1, G[17] == 1 - q1^17,
       T[k] == T[k - 1] - p2 q2^17 T[k - 18], T[0] == 1, T[1] == 1, T[2] == 1, T[3] == 1, T[4] == 1,
       T[5] == 1, T[6] == 1, T[7] == 1, T[8] == 1, T[9] == 1, T[10] == 1, T[11] == 1, T[12] == 1,
       T[13] == 1, T[14] == 1, T[15] == 1, T[16] == 1, T[17] == 1 - q2^17,
       U[k] == U[k - 1] - p3 q3^17 U[k - 18], U[0] == 1, U[1] == 1, U[2] == 1, U[3] == 1, U[4] == 1,
       U[5] == 1, U[6] == 1, U[7] == 1, U[8] == 1, U[9] == 1, U[10] == 1, U[11] == 1, U[12] == 1,
       U[13] == 1, U[14] == 1, U[15] == 1, U[16] == 1, U[17] == 1 - q3^17}, {G, T, U}, {k, 25, 200}]]),
   DataRange → {25, 200}, PlotRange → All,
   PlotStyle → {Dashing[Medium], Dashing[Medium], Dashing[Medium]}]]
```

The Plot1 and Plot2 represents the graphs in substitution error model. The recurrence solutions of $P(S)$ and $1 - P(S)$ were first obtained before the graph was plotted using the listLinePlot function in Mathematica.

A.2.2 Recurrence plots for substitution and deletion

```
In[ ]:=
Plot3 = With[{p = 0.05, q = 0.95, δ1 = 0.14, δ2 = 0.15, δ3 = 0.16},
  ListLinePlot[
   Transpose[RecurrenceTable[{G[k] == G[k - 1] - δ1 q^17 (1 - δ1)^16 G[k - 17] - p q^17 (1 - δ1)^17 G[k - 18],
      G[0] == 1, G[1] == 1, G[2] == 1, G[3] == 1, G[4] == 1, G[5] == 1, G[6] == 1, G[7] == 1, G[8] == 1,
      G[9] == 1, G[10] == 1, G[11] == 1, G[12] == 1, G[13] == 1, G[14] == 1, G[15] == 1, G[16] == 1,
      G[17] == 1 - (1 - δ1)^16 q^17,
      T[k] == T[k - 1] - δ2 q^17 (1 - δ2)^16 T[k - 17] - p q^17 (1 - δ2)^17 T[k - 18], T[0] == 1, T[1] == 1,
      T[2] == 1, T[3] == 1, T[4] == 1, T[5] == 1, T[6] == 1, T[7] == 1, T[8] == 1, T[9] == 1,
      T[10] == 1, T[11] == 1, T[12] == 1, T[13] == 1, T[14] == 1, T[15] == 1, T[16] == 1,
      T[17] == 1 - (1 - δ2)^16 q^17,
      U[k] == U[k - 1] - δ3 q^17 (1 - δ3)^16 U[k - 17] - p q^17 (1 - δ3)^17 U[k - 18], U[0] == 1, U[1] == 1,
      U[2] == 1, U[3] == 1, U[4] == 1, U[5] == 1, U[6] == 1, U[7] == 1, U[8] == 1, U[9] == 1,
      U[10] == 1, U[11] == 1, U[12] == 1, U[13] == 1, U[14] == 1, U[15] == 1, U[16] == 1,
      U[17] == 1 - (1 - δ3)^16 q^17}, {G, T, U}, {k, 25, 400}]], DataRange → {25, 400},
   PlotRange → {0.2, 1.0}, PlotLegends → {"δ=0.14", "δ=0.15", "δ=0.16"},
   PlotStyle → {Medium, Medium, Medium}]]
In[ ]:=
Plot4 = With[{p = 0.05, q = 0.95, δ1 = 0.14, δ2 = 0.15, δ3 = 0.16},
  ListLinePlot[
   1 -
    (Transpose[RecurrenceTable[{G[k] == G[k - 1] - δ1 q^17 (1 - δ1)^16 G[k - 17] - p q^17 (1 - δ1)^17 G[k - 18],
       G[0] == 1, G[1] == 1, G[2] == 1, G[3] == 1, G[4] == 1, G[5] == 1, G[6] == 1, G[7] == 1,
       G[8] == 1, G[9] == 1, G[10] == 1, G[11] == 1, G[12] == 1, G[13] == 1, G[14] == 1, G[15] == 1,
       G[16] == 1, G[17] == 1 - (1 - δ1)^16 q^17,
       T[k] == T[k - 1] - δ2 q^17 (1 - δ2)^16 T[k - 17] - p q^17 (1 - δ2)^17 T[k - 18], T[0] == 1,
       T[1] == 1, T[2] == 1, T[3] == 1, T[4] == 1, T[5] == 1, T[6] == 1, T[7] == 1, T[8] == 1,
       T[9] == 1, T[10] == 1, T[11] == 1, T[12] == 1, T[13] == 1, T[14] == 1, T[15] == 1, T[16] == 1,
       T[17] == 1 - (1 - δ2)^16 q^17,
       U[k] == U[k - 1] - δ3 q^17 (1 - δ3)^16 U[k - 17] - p q^17 (1 - δ3)^17 U[k - 18], U[0] == 1,
       U[1] == 1, U[2] == 1, U[3] == 1, U[4] == 1, U[5] == 1, U[6] == 1, U[7] == 1, U[8] == 1,
       U[9] == 1, U[10] == 1, U[11] == 1, U[12] == 1, U[13] == 1, U[14] == 1, U[15] == 1, U[16] == 1,
       U[17] == 1 - (1 - δ3)^16 q^17}, {G, T, U}, {k, 25, 400}]]), DataRange → {25, 400},
   PlotRange → {0.2, 1.0}, PlotStyle → {Dashing[Medium], Dashing[Medium], Dashing[Medium]}]]
```

The Plot3 and Plot4 represents the graphs in substitution and deletion error model. The recurrence solutions of $P(S)$ and $1 - P(S)$ were first obtained before the graph was plotted using the listLinePlot function in Mathematica.

A.2.3 Recurrence plots for substitution, deletion and insertion

```
In[ ]:= Plot5 = With[{p = 0.05, q1 = 0.91, r1 = 0.04, q2 = 0.9, r2 = 0.05, q3 = 0.89, r3 = 0.06,
    R = 0.45, δ = 0.15},
   ListLinePlot[
    (Transpose[
      RecurrenceTable[
       {G[k] == (1 + R - r1) G[k - 1] - (R - r1) G[k - 2] + p q1^17 (1 - δ)^17 (R - r1)/(1 - r1) G[k - 19] +
          q1^17 (1 - δ)^16 (-p + δ (p + R) - r1 (1 - R)/(1 - r1)) G[k - 18] - δ q1^17 (1 - δ)^16 G[k - 17],
        G[0] == 1, G[1] == 1, G[2] == 1, G[3] == 1, G[4] == 1, G[5] == 1, G[6] == 1, G[7] == 1,
        G[8] == 1, G[9] == 1, G[10] == 1, G[11] == 1, G[12] == 1, G[13] == 1, G[14] == 1, G[15] == 1,
        G[16] == 1, G[17] == 1 - (1 - δ)^16 q1^17, G[18] == 1 - (1 - δ)^16 q1^17 ((1 - r1 R)/(1 - r1) + p + q1 δ),
        T[k] == (1 + R - r2) T[k - 1] - (R - r2) T[k - 2] + p q2^17 (1 - δ)^17 (R - r2)/(1 - r2) T[k - 19] +
          q2^17 (1 - δ)^16 (-p + δ (p + R) - r2 (1 - R)/(1 - r2)) T[k - 18] - δ q2^17 (1 - δ)^16 T[k - 17],
        T[0] == 1, T[1] == 1, T[2] == 1, T[3] == 1, T[4] == 1, T[5] == 1, T[6] == 1, T[7] == 1,
        T[8] == 1, T[9] == 1, T[10] == 1, T[11] == 1, T[12] == 1, T[13] == 1, T[14] == 1, T[15] == 1,
        T[16] == 1, T[17] == 1 - (1 - δ)^16 q2^17, T[18] == 1 - (1 - δ)^16 q2^17 ((1 - r2 R)/(1 - r2) + p + q2 δ),
        U[k] == (1 + R - r3) U[k - 1] - (R - r3) U[k - 2] + p q3^17 (1 - δ)^17 (R - r3)/(1 - r3) U[k - 19] +
          q3^17 (1 - δ)^16 (-p + δ (p + R) - r3 (1 - R)/(1 - r3)) U[k - 18] - δ q3^17 (1 - δ)^16 U[k - 17],
        U[0] == 1, U[1] == 1, U[2] == 1, U[3] == 1, U[4] == 1, U[5] == 1, U[6] == 1, U[7] == 1,
        U[8] == 1, U[9] == 1, U[10] == 1, U[11] == 1, U[12] == 1, U[13] == 1, U[14] == 1, U[15] == 1,
        U[16] == 1, U[17] == 1 - (1 - δ)^16 q3^17, U[18] == 1 - (1 - δ)^16 q3^17 ((1 - r3 R)/(1 - r3) + p + q3 δ)},
       {G, T, U}, {k, 25, 600}]]), DataRange → {25, 600}, PlotRange → {0.1, 0.9},
    PlotLegends → {"r=0.04", "r=0.05", "r=0.06"}, PlotStyle → {Medium, Medium, Medium}]]
```

The Plot5 and Plot6 represents the graphs in substitution, deletion and insertion error model. The recurrence solutions of $P(S)$ and $1 - P(S)$ were first obtained before the graphs were plotted using the listLinePlot function in Mathematica.

In[ ]:=

```
Plot6 = With[{p = 0.05, q1 = 0.91, r1 = 0.04, q2 = 0.9, r2 = 0.05, q3 = 0.89, r3 = 0.06,
   R = 0.45, δ = 0.15},
  ListLinePlot[
   1 -
    (Transpose[RecurrenceTable[
       {G[k] == (1 + R - r1) G[k - 1] - (R - r1) G[k - 2] + p q1^17 (1 - δ)^17 (R - r1)/(1 - r1) G[k - 19] +
          q1^17 (1 - δ)^16 (-p + δ (p + R) - r1 (1 - R)/(1 - r1)) G[k - 18] - δ q1^17 (1 - δ)^16 G[k - 17],
        G[0] == 1, G[1] == 1, G[2] == 1, G[3] == 1, G[4] == 1, G[5] == 1, G[6] == 1, G[7] == 1,
        G[8] == 1, G[9] == 1, G[10] == 1, G[11] == 1, G[12] == 1, G[13] == 1, G[14] == 1, G[15] == 1,
        G[16] == 1, G[17] == 1 - (1 - δ)^16 q1^17, G[18] == 1 - (1 - δ)^16 q1^17 ((1 - r1 R)/(1 - r1) + p + q1 δ),
        T[k] == (1 + R - r2) T[k - 1] - (R - r2) T[k - 2] + p q2^17 (1 - δ)^17 (R - r2)/(1 - r2) T[k - 19] +
          q2^17 (1 - δ)^16 (-p + δ (p + R) - r2 (1 - R)/(1 - r2)) T[k - 18] - δ q2^17 (1 - δ)^16 T[k - 17],
        T[0] == 1, T[1] == 1, T[2] == 1, T[3] == 1, T[4] == 1, T[5] == 1, T[6] == 1, T[7] == 1,
        T[8] == 1, T[9] == 1, T[10] == 1, T[11] == 1, T[12] == 1, T[13] == 1, T[14] == 1, T[15] == 1,
        T[16] == 1, T[17] == 1 - (1 - δ)^16 q2^17, T[18] == 1 - (1 - δ)^16 q2^17 ((1 - r2 R)/(1 - r2) + p + q2 δ),
        U[k] == (1 + R - r3) U[k - 1] - (R - r3) U[k - 2] + p q3^17 (1 - δ)^17 (R - r3)/(1 - r3) U[k - 19] +
          q3^17 (1 - δ)^16 (-p + δ (p + R) - r3 (1 - R)/(1 - r3)) U[k - 18] - δ q3^17 (1 - δ)^16 U[k - 17],
        U[0] == 1, U[1] == 1, U[2] == 1, U[3] == 1, U[4] == 1, U[5] == 1, U[6] == 1, U[7] == 1,
        U[8] == 1, U[9] == 1, U[10] == 1, U[11] == 1, U[12] == 1, U[13] == 1, U[14] == 1, U[15] == 1,
        U[16] == 1, U[17] == 1 - (1 - δ)^16 q3^17, U[18] == 1 - (1 - δ)^16 q3^17 ((1 - r3 R)/(1 - r3) + p + q3 δ)},
       {G, T, U}, {k, 25, 600}]]), DataRange → {25, 600}, PlotRange → {0.1, 0.9},
   PlotStyle → {Dashing[Medium], Dashing[Medium], Dashing[Medium]}]]
```

Bibliography

[1] S. F. Altschul, W. Gish, W. Miller, *et al.*, "Basic local alignment search tool", *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[2] S. F. Altschul, T. L. Madden, A. A. Schäffer, *et al.*, "Gapped blast and psi-blast: A new generation of protein database search programs", *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[3] S. Batzoglou, L. Pachter, J. P. Mesirov, *et al.*, "Human and mouse gene structure: Comparative analysis and application to exon prediction", *Genome Research*, vol. 10, no. 7, pp. 950–958, 2000.

[4] S. Beretta, "Algorithms for strings and sequences: Pairwise alignment", *Encyclopedia of Bioinformatics and Computational Biology, eds. S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach*, pp. 22–29, 2018.

[5] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[6] M. Brudno, M. Chapman, B. Göttgens, *et al.*, "Fast and sensitive multiple alignment of large genomic sequences", *BMC Bioinformatics*, vol. 4, no. 1, pp. 1–11, 2003.

[7] M. Brudno, C. B. Do, G. M. Cooper, *et al.*, "Lagan and multi-lagan: Efficient tools for large-scale multiple alignment of genomic dna", *Genome Research*, vol. 13, no. 4, pp. 721–731, 2003.

[8] I. Bucak and V. Uslan, "An analysis of sequence alignment: Heuristic algorithms", in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, IEEE, 2010, pp. 1824–1827.

[9] S. Chhangawala, G. Rudy, C. E. Mason, *et al.*, "The impact of read length on quantification of differentially expressed genes and splice junction detection", *Genome Biology*, vol. 16, no. 1, pp. 1–10, 2015.

[10] S. J. Cokus, S. Feng, X. Zhang, *et al.*, "Shotgun bisulphite sequencing of the arabidopsis genome reveals dna methylation patterning", *Nature*, vol. 452, no. 7184, pp. 215–219, 2008.

[11] A. L. Delcher, S. Kasif, R. D. Fleischmann, *et al.*, "Alignment of whole genomes", *Nucleic Acids Research*, vol. 27, no. 11, pp. 2369–2376, 1999.

[12] S. Dreyfus, "Richard bellman on the birth of dynamic programming", *Operations Research*, vol. 50, no. 1, pp. 48–51, 2002.

[13] G. J. Filion, "Analytic combinatorics for bioinformatics i: Seeding methods", *bioRxiv*, p. 205 427, 2017. DOI: https://doi.org/10.1101/205427.

[14] ——, "Analytic combinatorics for computing seeding probabilities", *Algorithms*, vol. 11, no. 1, p. 3, 2018.

[15] G. J. Filion, R. Cortini, and E. Zorita, "Calibrating seed-based alignment heuristics with sesame", *bioRxiv*, 2019. DOI: 10.1101/619155. eprint: https://www.biorxiv.org/content/early/2019/12/12/619155.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2019/12/12/619155.

[16] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*. Addison-Wesley Publishing Company, 1994.

[17] P. H. Guzzi, "Algorithms for strings and sequences: Multiple alignment", *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 30, 2018.

[18] W. Haque, A. Aravind, and B. Reddy, "Pairwise sequence alignment algorithms: A survey", in *Proceedings of the 2009 conference on Information Science, Technology and Applications*, 2009, pp. 96–103.

[19] J. Kim, M. Ji, and G. Yi, "A review on sequence alignment algorithms for short reads based on next-generation sequencing", *IEEE Access*, vol. 8, pp. 189 811–189 822, 2020.

[20] S. Kumar and A. Filipski, "Multiple sequence alignment: In pursuit of homologous dna positions", *Genome Research*, vol. 17, no. 2, pp. 127–135, 2007.

[21] B. Langmead, C. Trapnell, M. Pop, *et al.*, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome", *Genome Biology*, vol. 10, no. 3, pp. 1–10, 2009.

[22] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform", *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[23] D. J. Lipman, S. F. Altschul, and J. D. Kececioglu, "A tool for multiple sequence alignment", *Proceedings of the National Academy of Sciences*, vol. 86, no. 12, pp. 4412–4415, 1989.

[24] M. L. Metzker, "Sequencing technologies—the next generation", *Nature Reviews Genetics*, vol. 11, no. 1, pp. 31–46, 2010.

[25] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[26] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison.", *Proceedings of the National Academy of Sciences*, vol. 85, no. 8, pp. 2444–2448, 1988.

[27] R. Sedgewick and P. Flajolet, *An introduction to the analysis of algorithms*. Addison-Wesley, Longman Publishing Co., 2013.

[28] J. T. Simpson, K. Wong, S. D. Jackman, *et al.*, "Abyss: A parallel assembler for short read sequence data", *Genome Research*, vol. 19, no. 6, pp. 1117–1123, 2009.

[29] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences", *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[30] J. L. Spouge, "Sequence alignment statistics", in *Problem Solving Handbook in Computational Biology and Bioinformatics*, L. S. Heath and N. Ramakrishnan, Eds. Boston, MA: Springer US, 2011, pp. 45–60, ISBN: 978-0-387-09760-2. DOI: 10.

1007/978-0-387-09760-2_3. [Online]. Available: https://doi.org/10.1007/978-0-387-09760-2_3.

[31] Y. Sun and J. Buhler, "Choosing the best heuristic for seeded alignment of dna sequences", *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–12, 2006.

[32] J. Wolff, B. Batut, and H. Rasche, *Mapping (galaxy training materials)*, [Online; accessed Mon Mar 14 2022], Nov. 2021. [Online]. Available: %5Curl%7Bhttps://training.galaxyproject.org/training-material/topics/sequence-analysis/tutorials/mapping/tutorial.html%7D.