Counting Unique Molecular Identifiers Using PCR-branching Models

Master's thesis in Master Programme Data Science and AI

Yizhe Gu
Hongyi Zhan

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022

Counting Unique Molecular Identifiers Using PCR-branching Models

Yizhe Gu
Hongyi Zhan

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2022

Counting Unique Molecular Identifiers Using PCR-branching Models
Yizhe Gu
Hongyi Zhan

Supervisor: Serik Sagitov, Department of Mathematical Sciences, Chalmers
Examiner: Marina Axelson-Fisk, Department of Mathematical Sciences, Chalmers

Master's Thesis 2022
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: DNA double-helix structure

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Counting Unique Molecular Identifiers Using PCR-branching Models
Yizhe Gu
Hongyi Zhan
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

DNA sequencing technology plays an essential role in biomedical research, especially in cancer genetics. However, errors in the sequencing process can confuse the results of detecting the actual variants. In order to distinguish these two mutations, researchers add Unique Molecular Identifiers(UMI), a unique tag sequence to each fragment of the original DNA molecules and do UMI counting on the sequenced molecules. Since the whole sequencing process, including DNA barcoding and amplification, is complex. Many parameters are invisible in actual biological experiments. A DNA barcoding algorithm based on level order traversal and a mathematical model of DNA amplification assuming molecules with memory and based on the growth patterns of population and molecular diversity is introduced, which are the critical steps for simulation. The project aims to obtain simulation results similar to the actual laboratory data by adjusting the model parameters and thereby determining the appropriate parameter values to help biological experiments perform better.

Keywords: DNA sequencing, Polymerase Chain Reaction(PCR), Unique Molecular Identifiers(UMI), Size-dependent branching.

Acknowledgements

We would like to express sincerest gratitude to our supervisor, Serik Sagitov, for his guidance in the past six months, discussing with us every week and giving us a lot of research directions. We also thanks him for helping us the paper writing and presentation.

We are very grateful to our examiner, Marina Axelson-Fisk, for recommending such an interesting topic, giving us the opportunity to participate in this project and following up on our progress during the period of master thesis.

Anders Ståhlberg and Manuel Marceliano Luna Santa-María who are the researchers in Sahlgrenska Center for Cancer Research in Gothenburg answer our doubts on biology and show us around the research center to understand the process of biology experiment more visually. Thanks to their help, the project could be carried out smoothly.

Yizhe Gu, Hongyi Zhan, Gothenburg, May 2022

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

PCR	Polymerase Chain Reaction
UMI	Unique molecular identifiers
NGS	Next-generation Sequencing
CDF	Cumulative Distribution Function

Nomenclature

Below is the nomenclature of parameters that have been used throughout this thesis.

Parameters

r_1, r_2, r_3, r_4	DNA barcoding efficiency rates
r	The average of DNA amplification rate for exponential growth
α	the α in the beta distribution of DNA amplification
d_2	The second dilution rate
K	The environment in DNA amplification
M	The mutation rate in DNA amplification

Contents

List of Figures

List of Tables

1
Introduction

Since researchers began to dissect the double helix structure of DNA, significant efforts have been made to explore the complexity and variability of genomes. DNA sequencing, the most popular technology in molecular biology, is the process of determining the ACGT sequence of four nucleotides in a DNA segment which can help researchers analyze genes better in the field of biotechnology and medical research. For example, DNA sequencing helps determine the type of cancer, which then guides the patient's treatment decisions. In addition, It also helps researchers understand changes in cancer genes and develop new drugs. The primary barriers to deeper genome investigation are sequencing throughput limitations and high costs. High-throughput technologies have initially addressed these issues which makes the expenditure fall rapidly, giving rise to a new term: Next-Generation Sequencing (NGS). This massively parallel sequencing technique can determine the order of nucleotides in the complete or partial genome of DNA or RNA. Applying NGS technology to DNA sequencing allows for massively parallel DNA analysis, increasing the speed and reducing the cost. However, since the whole sequencing process is very complicated and many parameters are not visible in the actual biological experiments, it is necessary to design reasonable mathematical models and find parameter values to guide further on how to conduct the experiments more efficiently and cost-effectively.

Polymerase Chain Reaction(PCR) used in DNA sequencing can amplify specific DNA fragments to make enough analyzable copies. It can be regarded as a particular DNA replication in vitro, which can significantly increase DNA trace amounts. The sample is first heated to denature and separate the DNA into two single strands, and then two new DNA strands are constructed using the original strand as a template with the help of polymerase. Since the advent of PCR, mutation gene detection technology based on PCR has developed rapidly. Not only can it detect the mutated gene in a short time, but even a minimal amount of tissue can be detected by PCR amplification.

However, some errors also occur and are not easily detectable in DNA sequencing. To improve the accuracy and exclude errors caused by DNA polymerases and sequencing processes in the subsequent analysis, the concept of Unique Molecular Identifiers (UMI), also known as molecular barcoding, is introduced. It is a unique tag sequence which is added to each fragment of the original sample to distinguish between thousands of different fragments in the same sample. In turn, PCR amplification bias can be handled in DNA sequencing. For example, the two types of

errors are indistinguishable without the UMIs, as shown on the left of Figure 1.1. However, if UMIs are added, the copies with the same UMI should be derived from the same molecule in the result. Therefore, if it is a low-frequency mutation on the original molecule, all copies of it should carry this mutation. If it is a random error, it is more likely to be present in only one copy or several copies because of an error that happened to occur during a round of PCR amplification or sequencing. Therefore, it is essential to count UMIs in DNA sequencing.

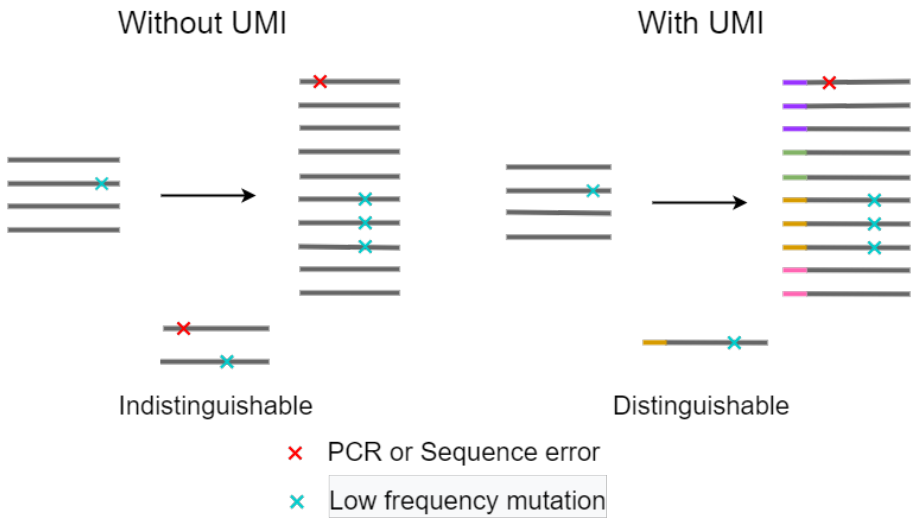

Figure 1.1: How to distinguish the error types by UMIs.

1.1 Objective

This thesis project aims to develop mathematical models and algorithms to reproduce actual data, which is the result of DNA sequencing, thus discovering the hidden parameter values in the experiments. The whole experiment is divided into two major parts: PCR barcoding and PCR amplification. The findings of the project will be helpful in future research by Sahlgrenska Center for Cancer Research in Gothenburg.

1.2 Research questions

The research questions which are chosen as a guideline for the thesis project are

- How do the different parameters during the simulation influence the final results?
- How to select parameters to avoid overfitting.
- How to tune the model within the margin of errors to get valuable parameters on the premise of reproducing the real data?

1.3 Outline

Section 2 and Section 3 provide the background and theory of the chosen models and algorithms based on DNA sequencing. Section 2 describes the primary methods for DNA amplification and sequencing and then summarizes the existing methods through an extensive literature reading in this field. Section 4 introduces the specific process of implementing the models and algorithms, and which results are shown in Section 5 with a brief discussion. The comparison and analysis of the whole project are discussed in detail in Section 6, where the future work is also included. Finally, Section 7 summarizes the entire work of the thesis project.

2

Background

The development of DNA sequencing has dramatically advanced the exploration of biological and medical research. From the dideoxyribonucleotide chain-termination method invented by Sanger in the 1970s, which can only sequence one DNA fragment [1] to next-generation sequencing(NGS), a massively parallel sequencing technology which identifies variants and mutant alleles with higher sensitivity, DNA sequencing has entered an era of high throughput and low cost.

DNA sequencing is separated into four main processes: DNA barcoding, DNA amplification, sequencing, and data analysis which the first three steps are shown in Figure 2.1.

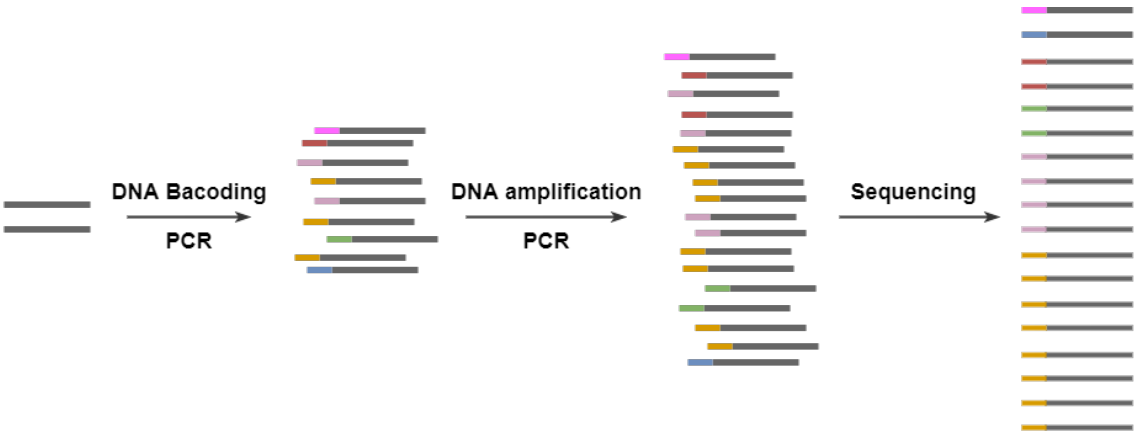

Figure 2.1: An example of DNA sequencing.

However, the experiment is not simply to count UMIs. The sequencing results include the cluster size and the number of cluster sizes. The cluster size is written as X, which means that for one kind of UMI, it appears X times. The number of cluster sizes is written as Y, which means that there are Y kinds of UMIs appearing X times. According to this counting method, the sequencing results in Figure 2.1 are shown in Table 2.1.

Table 2.1: The result of DNA sequencing in Figure 2.1.

Cluster size (X)	1	2	3	4	5	6	7	8
The number of cluster size (Y)	2	2	0	1	0	0	0	1

Since individual DNA molecules cannot be replicated quantitatively for detection, counting the number of individual DNA molecules becomes very difficult [2]. Besides, DNA damage and polymerase inactivation can also cause errors in sequencing results [3]. Therefore, to overcome these difficulties, Unique Molecular Identifiers(UMIs) are introduced into the sequencing process, which are short sequences composed of 8-12 nucleotides and are attached explicitly to each target DNA molecule by a limited number of PCR cycles during the DNA barcoding process of library preparation. After sequencing, it can be traced back to the original DNA as long as the number of the same UMI is counted. Meanwhile, performing the standard multiplex pre-amplification approach can avoid non-specific PCR products due to the generation of randomized UMI sequences [4]. If the DNA sample is too small, minute but essential information will be challenged to retain. Therefore, all molecules with UMIs are subjected to PCR amplification to obtain more copies of DNA after barcoding, and then the reads of UMIs can be counted after sequencing.

Researchers have shown that the branching process can be used as a stochastic model for the problem of counting molecules after multiple cycles of PCR amplification [5] [6] which can also be applied to the PCR amplification process with UMIs [7] [8]. The multi-type Galton-Watson process with immigration proposed by Serik Sagitov and Anders Ståhlberg [8] is a particular example of the multi-type Galton-Watson process with neutral mutations studied in [9]. In the first part of this project, we focus on using the branching process formulation in [8] as a theoretical basis to design a typical algorithm of level order traversal of a Binary tree to simulate the barcoding process and verify the theoretical results in the paper.

For amplification before sequencing, we mainly concentrate on the amplification rate. Florian et al. proposed that PCR amplification still follows a Galton-Watson branching process [10] and UMI reads are determined by a Poissonian sampling model for NGS [11]. However, since the polymerase works feverishly at the beginning, gradually inactively and finally does not work with increasing PCR cycles, the actual PCR amplification rate is not a simple constant probability. Jagers et al. introduce a formula with quite large Michaelis-Menten constant K and the number of molecules to describe the probability of successful amplification [12]. Lalam et al. [13] introduce a saturation parameter on top of [12] to precisely calculate amplification rate. In this project, we build a model of amplification rate based on [12] [13] where molecules have memory and each amplification cycle obeys a new beta distribution.

Since lower amplification rates result in fewer UMI reads, leading to complete unsequencing and variation that can occur during sequencing, the results of UMI reads are still inaccurate. For sequencing errors, Smith et al. proposed an algorithm to merge the original UMI with error versions that are highly similar to it [14]. Pflug et al. introduce an error correction threshold to remove UMI reads below this threshold [7]. In this project, edit distance which indicates the number of nucleotide differences between two UMIs is used for correction.

3

Theory

The theory of the whole process shown in Figure 3.1 is described in Section 3. First, the target molecules are amplified in a limited number of cycles with four different barcoding efficiency rates. UMI is added to the molecules that meet the conditions in this barcoding process during 3 or 5 PCR cycles. Next, the molecules after the first stage dilution are amplified in 26 or 28 PCR cycles based on the amplification rate corresponding to the growth pattern of population size and the diversity of molecules with memory. Then the molecules are sequenced after the second dilution, and the UMI reads are the final results.

Section 3.1 introduces the perfect and imperfect duplication in the PCR barcoding process. The detailed DNA amplification and sequencing are shown in Section 3.2 and Section 3.3, respectively.

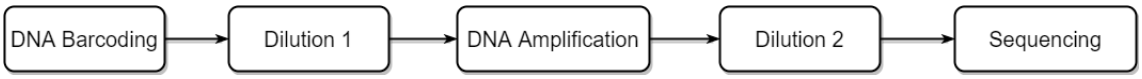

Figure 3.1: The whole process of the project.

3.1 DNA barcoding

The accuracy in the PCR process can be improved with the help of adding UMI, which is beneficial to error correction. In this PCR barcoding process, we mainly focus on the results of UMI counting starting from a double-stranded molecule. According to biological theory, UMIs are only counted on the complete DNA strand, that is, a DNA strand with a head and tail containing both forward primer and reverse primer.

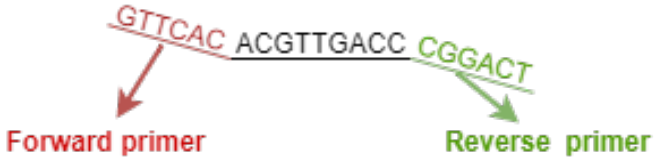

Figure 3.2: A complete DNA strand

We will take the PCR barcoding process at cycle $t = 3$ as an example to elaborate on the perfect and imperfect PCR barcoding processes, respectively.

3.1.1 Perfect PCR barcoding process

The DNA is assumed to be fully amplified during the barcoding PCR process. That is, the PCR barcoding efficiency rate is 100%. The outcomes of three cycles of the perfect PCR barcoding process are shown in Figure 3.3. At the top of the figure, the double strands at $t = 0$ represent the initial DNA molecule which are labelled 0 and 1. The new strand obtained by taking the strand of the initial molecule as a template must be incomplete. The difference is that there is no UMI in the primer of the new strand, which is copied according to the sense strand, while the UMI exists in the primer of the new strand obtained by using the nonsense strand as the template. Therefore, in the following cycles, numbers are still used to represent newly generated incomplete DNA strands without UMIs, and those with UMIs are represented by lowercase letters a, c, g. It can be considered that there are two kinds of primers, and when a DNA strand containing a primer is used as a template for amplification, it must generate a new DNA strand with both primers, that is, a full strand named with capital letters A, B, C, D, E, F. Furthermore, we can conclude that when the complete strand containing the two primers is used as the template strand, the new strand obtained by amplification must be a complete strand. The difference is that the nonsense strand obtained by the complete sense strand is the same type as the template strand, while the sense strand obtained by the complete nonsense strand is a new complete strand containing the new UMI.

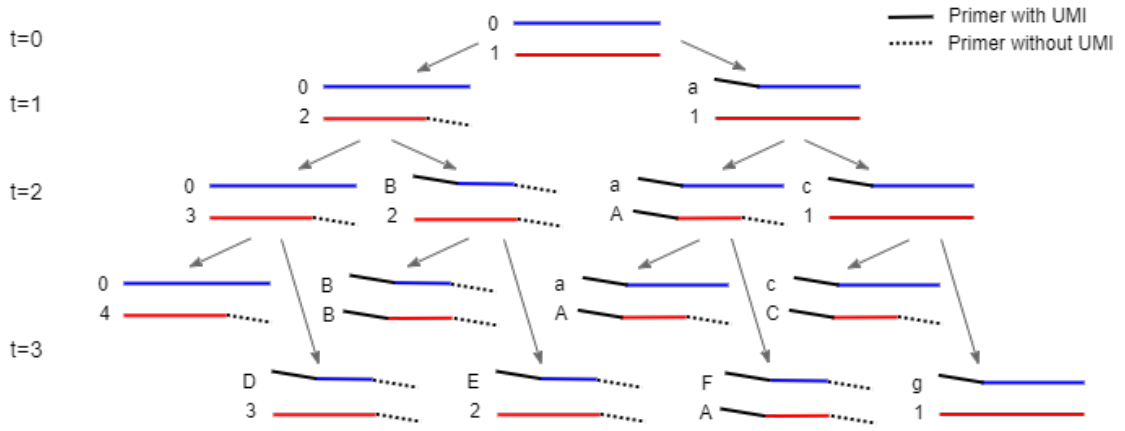

Figure 3.3: The perfect amplification in $t = 3$

According to the above rules, we can get six types of DNA molecules

- S_1, the original sense strand in blue labeled by 0
- S_2, the original nonsense strand in red labeled by 1
- S_3, the incomplete nonsense strands without UMI in red labeled by $2, 3, 4, \ldots$
- S_4, the incomplete sense strands with UMI in blue labeled by $a, b, c, \ldots$

- S_5, complete sense strands in blue labeled by B, D, E, F, ...
- S_6, complete nonsense strands in red labeled by A, C, ...

The generation of new strands can be written as

$$\begin{aligned} S_1 \longrightarrow S_3, \quad S_2 \longrightarrow S_4, \quad S_3 \longrightarrow S_5, \\ S_4 \longrightarrow S_6, \quad S_5 \longrightarrow S_6, \quad S_6 \longrightarrow S_5 \end{aligned} \tag{3.1}$$

When $t = 3$, there are eight complete single-stranded molecules and six unique UMIs, A, B, C, D, E, F, which have the following size

Table 3.1: The cluster and its size of perfect PCR barcoding process at t=3.

Cluster	A	B	C	D	E	F
Size	2	2	1	1	1	1

It can be generalized to the situation from $t = 2$ to $t = 6$. Their cluster size and numbers are shown in Table 3.2 according to the counting method in Section 2. That is, when t=2, two kinds of UMIs appear once. When t=3, four kinds of UMIs appear once, two kinds of UMIs appear twice and so on.

Table 3.2: The cluster sizes and their numbers of perfect PCR amplification at t cycles.

Cluster size	1	2	3	4	5	*Total*
$t = 2$	2	0	0	0	0	2
$t = 3$	4	2	0	0	0	6
$t = 4$	8	4	2	0	0	14
$t = 5$	16	8	4	2	0	30
$t = 6$	32	16	8	4	2	62

A cluster is a collection of complete molecules with the same UMI. Based on an early version of Serik paper [8], let $Z_t(m)$ be the number of cluster size m at cycle t. It is obvious that

$$Z_t(m) = 0, \quad m \geq t \tag{3.2}$$

let Z_t be the total number of cluster size at cycle t.

$$Z_t = Z_t(1) + ... + Z_t(t-1) \tag{3.3}$$

According to Table 3.2 and Figure 3.3, we can deduce

$$Z_1 = 0, \quad Z_{t+1} = 2Z_t + 2, \quad t \geq 1 \tag{3.4}$$

Therefore,

$$Z_t = 2^t - 2, \quad t \geq 1 \tag{3.5}$$

Furthermore, we can get

$$Z_{t+1}(1) = Z_t + 2, \quad t \geq 1 \tag{3.6}$$

$$Z_{t+1}(m+1) = Z_t(m), \quad m \geq 2, t \geq 1 \tag{3.7}$$

inferring that increasing the cluster size by one reduces the number of clusters by half

$$Z_t(m) = 2^{t-m}, \quad m \geq 1, t \geq m+1 \tag{3.8}$$

Therefore, the proportion of the number of cluster sizes to the total number for large t is

$$Z_t(m)/Z_t \to 2^{-m}, \quad m \geq 1, t \to \infty \tag{3.9}$$

3.1.2 Imperfect PCR barcoding process

In practical situations, we cannot guarantee that each DNA strand can be amplified entirely affected by the experimental environment. Therefore, four different probabilities r_1, r_2, r_3 and r_4 are used in this PCR barcoding process which is related to the generation of new strands. According to information from Sahlgrenska Center for Cancer Research, their relationship satisfies

$$0 < r_1 < r_2, r_3 < r_4 < 1 \tag{3.10}$$

r_1 is the PCR barcoding efficiency rate of $S_1 \longrightarrow S_3$ and $S_2 \longrightarrow S_4$ in equation 3.1. r_2 is the PCR barcoding efficiency rate of $S_3 \longrightarrow S_5$, which may be similar to r_3, the PCR barcoding efficiency rate of $S_4 \longrightarrow S_6$. r_4 is associated with the generation from complete strands $S_5 \longrightarrow S_6$ and $S_6 \longrightarrow S_5$.

It is straightforward that the cluster size of the imperfect PCR barcoding process is fewer than that of the perfect PCR barcoding process due to the barcoding efficiency rates.

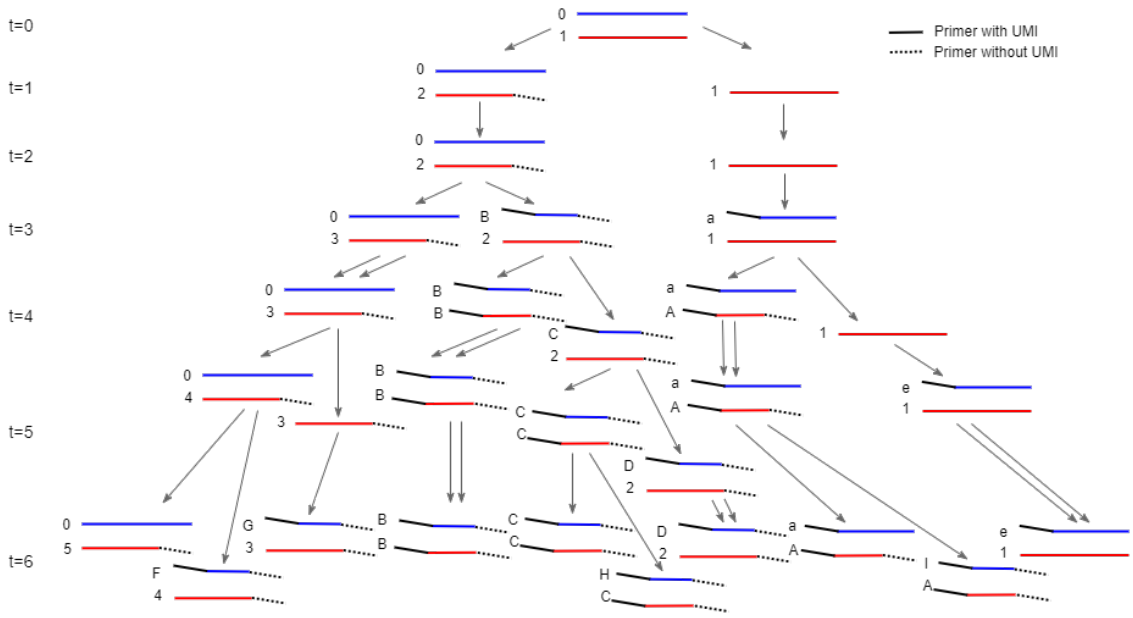

Figure 3.4: An example of the imperfect PCR barcoding process

The numbers of each cluster size in Figure 3.4 are

Table 3.3: The cluster sizes and their numbers of imperfect PCR barcoding process at t cycles.

Cluster size	1	2	3	4	5	*Total*
$t = 2$	0	0	0	0	0	0
$t = 3$	1	0	0	0	0	1
$t = 4$	2	1	0	0	0	3
$t = 5$	2	2	0	0	0	4
$t = 6$	3	2	1	0	0	6

Based on an early version of Serik paper [8], for one particular molecule which is duplicated by efficiency rate r, the process can be regarded as a Binomial distribution $Bin(1, r)$. Since all molecules are independent in the process, with the number of type S_i molecules at cycle t being Z_t^i, we get the branching process.

$$\begin{aligned} Z_t^3 &= Bin(1, r1) + Z_{t-1}^3 \\ Z_t^4 &= Bin(1, r1) + Z_{t-1}^4 \\ Z_t^5 &= Bin(Z_{t-1}^3, r2) + Bin(Z_{t-1}^6, r4) + Z_{t-1}^5 \\ Z_t^6 &= Bin(Z_{t-1}^4, r3) + Bin(Z_{t-1}^5, r4) + Z_{t-1}^6 \end{aligned} \tag{3.11}$$

The expectation $M_t^i = E(Z_t^i)$

$$M_t^3 = M_t^2 = r_1 + M_{t-1}^2 = r_1 t$$

$$M_t^5 = r_1 r_2 (t-1) + M_{t-1}^5 + r_4 M_{t-1}^6 \tag{3.12}$$

$$M_t^6 = r_1 r_3 (t-1) + r_4 M_{t-1}^5 + M_{t-1}^6$$

To obtain the number of cluster sizes, two parts are divided to compute. Let $X_t(m)$ be the number of S_4 clusters of size m at cycle t. Then for $1 \leq m \leq t-1$,

$$\begin{aligned} X_t(m) &= Bin(X_{t-1}(m-1), r_3) + Bin(X_{t-1}(m), 1-r_3) \\ X_t(0) &= Bin(1, r_1) + Bin(X_{t-1}(0), 1-r_3) \end{aligned} \tag{3.13}$$

And

$$Z_t^4 = X_t(0) + ... + X_t(t-1) \tag{3.14}$$

Expected values $K_t(m) = E(X_t(m))$, for $0 \leq m \leq t-1$,

$$K_t(m) = K_{t-1}(m-1)r_3 + K_{t-1}(m)(1-r3), \quad K_t(-1) = \frac{r_1}{r_3} \tag{3.15}$$

The second part is S_5 cluster which uses $Y_t(m)$ to represent the number of cluster size m at cycle t. Then for $2 \leq m \leq t-1$,

$$\begin{aligned} Y_t(1) &= Bin(Z_{t-1}^3, r_2) + Bin(Z_{t-1}^6, r_4) + Bin(Y_{t-1}(1), 1-r_4) \\ Y_t(m) &= Bin(Y_{t-1}(m-1), r_4) + Bin(Y_{t-1}(m), 1-r_4) \end{aligned} \tag{3.16}$$

And

$$Z_t^5 = Y_t(1) + ... + Y_t(t-1) \tag{3.17}$$

Expected values $L_t(m) = E(Y_t(m))$, for $0 \leq m \leq t-1$,

$$\begin{aligned} L_t(1) &= r_1 r_2 (t-1) + r_4 M_{t-1}^6 + L_{t-1}(1)(1-r_4) \\ L_t(m) &= L_{t-1}(m-1)r_4 + L_{t-1}(m)(1-r4) \end{aligned} \tag{3.18}$$

Furthermore, the number and the total number of clusters and their expectation for $1 \leq m \leq t-1$ are

$$\begin{aligned} Z_t &= Z_t^4 + Z_t^5 - X_t(0) \\ Z_t(m) &= X_t(m) + Y_t(m) \end{aligned} \tag{3.19}$$

$$M_t = E(Z_t) = M_t^4 + M_t^5 - K_t(m)$$
$$M_t(m) = E(Z_t(m)) = K_t(m) + L_t(m) \tag{3.20}$$

3.1.3 Breadth-First Search

The traversal algorithm Breadth-first search(BFS) is used for connected graphs, which is a blind search. The goal is to do all nodes searching in the graph and examine the results systematically which means it does not consider the possible location of the result but scours the entire graph until it finds a result. Generally, the queue, a data structure, is used to assist the implementation of the BFS algorithm. The steps of this algorithm are

- Add the root node into the queue.
- Take the first node from the queue and check whether it is the wanted node or not. If it is found, end the search and return the result. Otherwise add all unexamined nodes to the queue.
- When the queue is empty which means all nodes has been went through, end the search and return the results.
- Repeat step 2.

For example, the order of nodes in Figure 3.5 is $A \to B \to C \to D \to E \to F \to G$.

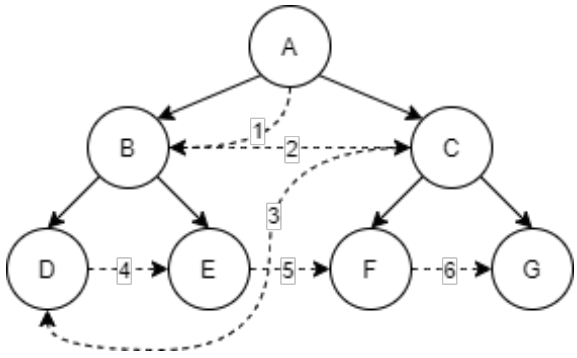

Figure 3.5: The order of nodes in Breadth-first search

The PCR barcoding process of DNA in Figure 3.5 can be regarded as a connected graph where the new molecules are generated one by one according to the barcoding rules, and the UMIs are counted at the same time. The only difference is that if the target molecule with UMI is found, the program does not stop but continues going through all the molecules until it finishes all cycles.

3.2 Dilution

The reagent containing DNA molecules is diluted during the dilution, and then a portion of it is extracted for use in the next stage. The parameter in this process is called the dilution rate. Dilution rate is the main difference between these two dilution stages, while other issues are so minor that they make little difference. For dilution 1, its dilution rate is 3, while for dilution 2, this is a hyperparameter symbolized as $d2$. Whether it can participate in the next stage is independent of each molecule. Taking the rate in the first dilution as an example, it equals three means that for each molecule, it has a one-third probability of advancing to the next stage and a probability of two-thirds of being discarded. This is a Bernoulli experiment.

$$cluster\ size\ after\ dilution \sim Binomial(cluster\ size\ before\ dilution, \frac{1}{dilution\ rate}) \tag{3.21}$$

This equation shows the situation in the following. Let us consider a cluster after the dilution. Its cluster size will follow a binomial distribution, with the parameters being the cluster size before dilution and the inverse dilution rate.

3.3 DNA amplification

DNA amplification can be viewed as DNA replication. The polymerase is an enzyme involved in DNA replication. It catalyzes the polymerization of deoxyribonucleotides, mainly in the form of a template.

Similar to DNA barcoding, amplification is not always achieved successfully, so the amplification efficiency rate is used to measure the probability that a molecule will amplify successfully or not under the influence of the environment and the overall population size in the environment. Also, amplification does not always produce a perfect complementary DNA molecule. Sometimes mutation occurs, also known as the phantom, which produces a molecule that is not perfectly complementary to the original molecule. The mutation rate is to measure the probability of this phenomenon.

3.3.1 Amplification efficiency rate

Traditional measure [12] for amplification efficiency rate has some limitations under this topic. It considers only one kind of molecule, which means every molecule is treated equally in the simulation. Furthermore, it does not reflect the differences among molecules and their own property in the environment. Instead, the PCR amplification steps that must be performed before sequencing often amplify different molecules with different efficiencies, thus biasing the sequencing results and the measured abundance [15]. As a result, variation based on beta distribution is introduced to show the diversity of the amplification efficiency rate of different molecules.

At the same time, molecules are assumed to have some inherent properties, also known as memory. If a cluster of molecules exhibits a greater efficiency rate than others at the start, it can be expected to maintain in the future amplification.

The most essential things for distribution are the mean and variance. These will be elaborated on below.

3.3.1.1 Amplification efficiency rate from overall perspective

[12] reveals the relationship between molecule population size, environment and amplification efficiency rate. Despite the variability between different molecules, this theory can be used as the overall expression of this process. It is used as the mean of the beta distribution in the whole simulation. This shows that if all molecules are treated as a whole, how do they amplify based on the environment and population size.

Suppose the number of molecules at round $\#n$ in the environment is z_n, the probability of successful amplification r_n is given by

$$r_n = \frac{K}{K + z_n} \tag{3.22}$$

, where K is the Michaelis-Menten constant. K represents the support of the environment for amplification and will not change across the process. However, K does not mean the limitation of the environment.

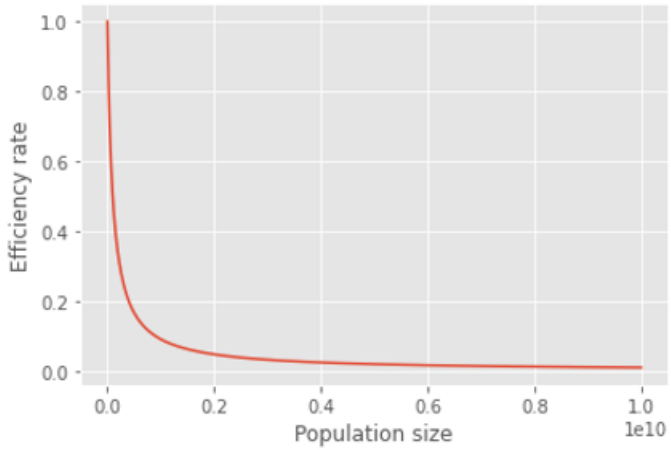

Figure 3.6: The relationship between efficiency rate and population size when $K = 10^8$

Figure 3.6 illustrates the equation 3.22.The situation is that when z_n is much smaller than K, r_n is close to 1. Furthermore, with the increase of z_n, r_n gradually converges to 0 but the whole amplification will not stop. As a result, the growth of the population size will be exponential in the beginning. When z_n is much bigger than K, the growth of the number of the molecules in the environment will be linear

because:

$$\begin{aligned} r_n \cdot z_n &= \frac{K z_n}{K + z_n} \\ &= \frac{K}{\frac{K}{z_n} + 1} \end{aligned}$$

so that

$$\lim_{z_n \to \infty} r_n \cdot z_n = K \tag{3.23}$$

When z_n is big enough, the environment will increase K more molecules in every amplification round.

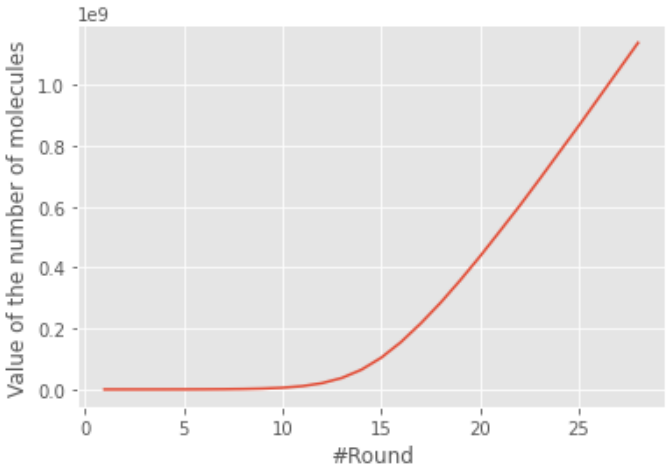

Figure 3.7: Total number of molecules during amplification with $K = 10^8$

Figure 3.7 illustrates how the change of two phases. When $z_n \leq 10^8$, about round 15 in the figure, z_n grows exponentially. After round 15, amplification quickly turns into the linear growth phase. This phenomenon is called saturation[23]. Some models will introduce a new parameter to describe the point that divides the two phases, which is also called the saturation point. However, this point will be very close to K[23]. Since we do not want too many parameters, which may result in overfitting, we assume that K is the watershed into the saturation phase.

3.3.1.2 Linear and exponential amplification

Figure 3.7 shows how the amplification goes from linear growth to exponential and into saturation. However, the situation in the actual experiment is that the efficiency rate in the exponential phase is not very close to 1, but only a very large number[13]. Therefore we add a hyperparameter r to fix the distribution of the efficiency rate before the population size reaches K. Figure 3.8 illustrates the change of efficiency rate of one cluster of molecules in the simulation.

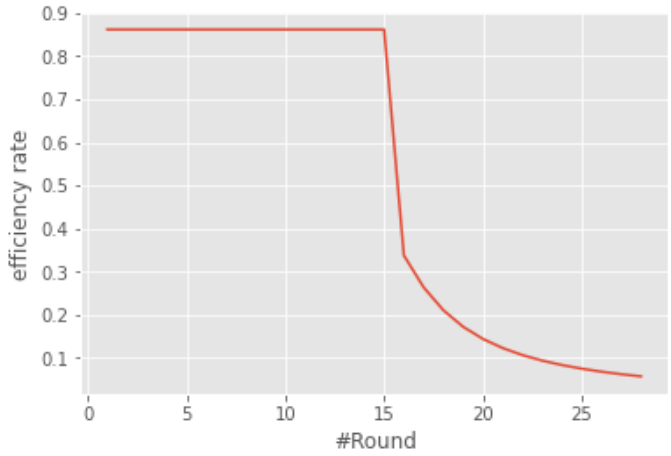

Figure 3.8: The change of the efficiency rate of one cluster of molecules at different round n

3.3.1.3 Variation of efficiency rate

The simulation model uses the beta distribution to model the variation of efficiency rates r_n of different clusters at different round n to show the diversity among molecules. Here focus on the difference in the efficiency rates of different clusters at the same round.

$$r_n \sim Beta(\alpha_n, \beta_n) \quad (r \in [0, 1)) \tag{3.24}$$

The mean of Beta distribution is

$$\mathbb{E}(Beta(\alpha_n, \beta_n)) = \frac{\alpha_n}{\alpha_n + \beta_n} \tag{3.25}$$

To make the mean of the Beta distribution equals to **??**,

$$\frac{K}{K + z_n} = \frac{\alpha_n}{\alpha_n + \frac{z_n \cdot \alpha_n}{K}} = \frac{\alpha_n}{\alpha_n + \beta_n} \tag{3.26}$$

so that

$$\beta_n = \frac{z_n \cdot \alpha_n}{K} \tag{3.27}$$

For beta distribution, the parameters α_n and β_n have 7 different combinations with different shapes as shown in Figure 3.9

For efficiency rate, a reasonable assumption is that it obeys a skew-normal distribution, instead of a monotonically increasing or decreasing one. As a result, α_n should be greater than one across the process.

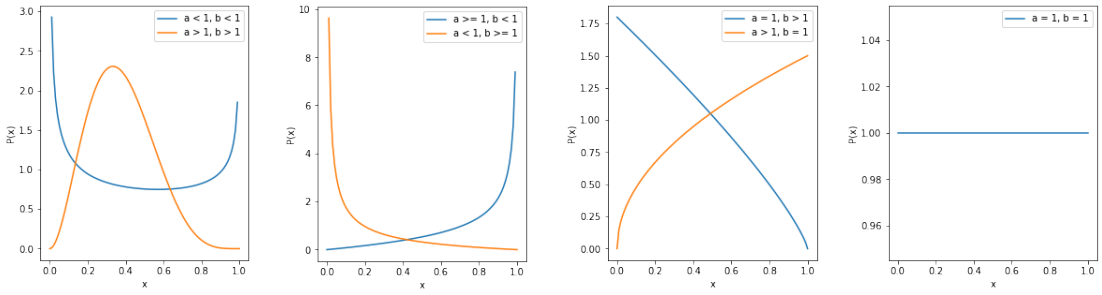

Figure 3.9: The shape of 7 different combinations of parameters α_n and β_n for distribution $Beta(\alpha_n, \beta_n)$

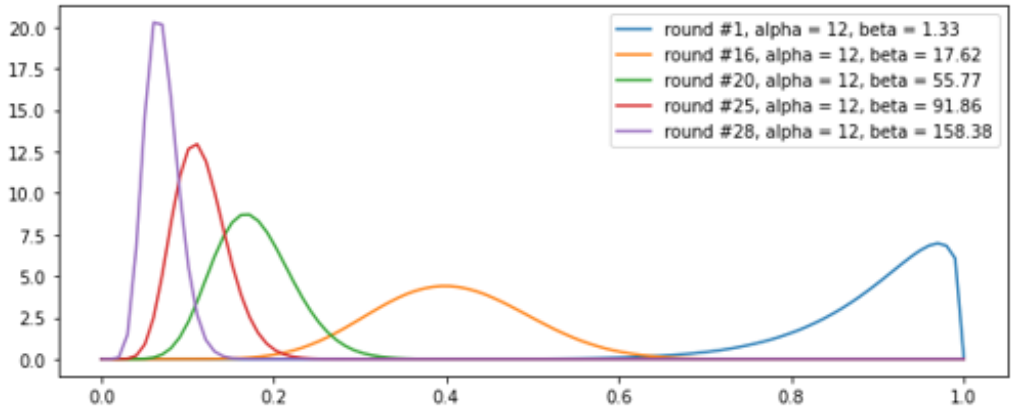

Figure 3.10: Beta distribution in simulation at different rounds: blue #round 1, orange #round 16, green #round 20, red #round 25, purple #round 28 with $\alpha = 5$ and $K = 10^8$

Suppose α_n is fixed across the simulation, which means $\alpha_n = \alpha$, where α is a hyperparameter in the simulation. Figure 3.10 exemplifies how the change of beta distribution will be during amplification under this assumption. As for amplification proceeds, the center of the distribution moves left. Furthermore, the distribution changes dramatically when the population size approaches K, which is about the 15th round.

At the same time, we focus on the change of the variance during amplification. The variance of Beta distribution is

$$Var(Beta(\alpha, \beta_n)) = \frac{\alpha\beta_n}{(\alpha + \beta_n)^2(\alpha + \beta_n + 1)} \tag{3.28}$$

Use $\mathbb{E}_n$ as abbreviation of $\mathbb{E}(Beta(\alpha, \beta_n)$. Substituting $\beta_n = \frac{\alpha}{\mathbb{E}_n} - \alpha$ gives

$$Var(Beta(\alpha, \beta_n)) = \frac{(1 - \mathbb{E}_n)\mathbb{E}_n^2}{\alpha + \mathbb{E}_n}, \tag{3.29}$$

which can be viewed as a convex quadratic function of $\mathbb{E}_n$. From equation 3.25, when α is fixed, $\mathbb{E}_n$ is a monotonically decreasing function of β_n while equation 3.27

points out β_n is monotonically increasing with z_n. As a result, as z_n becomes bigger, $Var(Beta(\alpha, \beta_n))$ will increase and then decrease. Figure 3.11 illustrates the change of variance. This change is in line with the general perception. When amplification begins, most of the molecules amplify at a very high-efficiency rate at first. As for amplification proceeds, a portion of the molecules has entered a linear growth phase, while another portion is still growing exponentially, creating a high point of variance. After amplification enters the second phase, the vast majority of molecules grow linearly in number, so that the variance decreases.

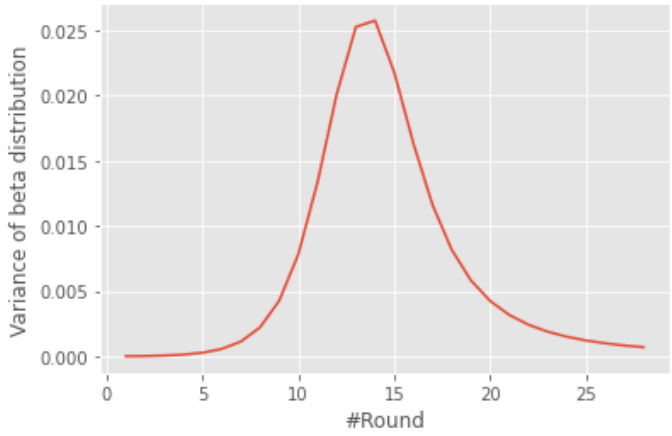

Figure 3.11: Variance of beta distributions at different #round in amplification with $\alpha = 5$ and $K = 10^8$

Some idea may think the variance among molecules is always exist. As a result, another assumption is that the variance of distribution may be fixed. Under this idea, suppose variance is fixed to σ,

$$\frac{\frac{z \cdot a^2}{K}}{(a + \frac{z \cdot a}{K})^2(a + \frac{z \cdot a}{K} + 1)} = \sigma$$
$$\sigma[(K + z)^3 a^3 + K(K + z)^2 a^2] = K^2 z a^2$$

Since $a \neq 0$, so that

$$a = \frac{K(Kz - \sigma(K + z)^2)}{\sigma(K + z)^3} \tag{3.30}$$
$$b = \frac{z(Kz - \sigma(K + z)^2)}{\sigma(K + z)^3} \tag{3.31}$$

$$\frac{K(Kz - \sigma(K + z)^2)}{\sigma(K + z)^3} > 1$$
$$\sigma < \frac{K^2 z}{(2K + z)(K + z)^2} \tag{3.32}$$

Since α has been supposed to be bigger than 1. Moreover, denominator is of higher order than the numerator. The variance will be very small to meet all the requirements. As a result, we still use *alpha* as the hyperparameter instead of variance.

3.3.1.4 The property of molecules

The property of a molecule lies in the fact that different molecules will act differently in the same environment. The very natural idea is to assume that there is a sufficient number of clusters and order them by r decently. Then another assumption can be made that the ranking of the molecules does not change in each round of amplification. This can be called the memory of molecules.

A cumulative Distribution Function (CDF) is introduced to represent this feature. For point $(x, y)(y \in [0, 1]$ in CDF, it represents that for a distribution, it is in the y quantile when the value x is taken. In other words, if we take N samples from the distribution (N is large enough), then x is larger than $N \cdot y$ of the samples.

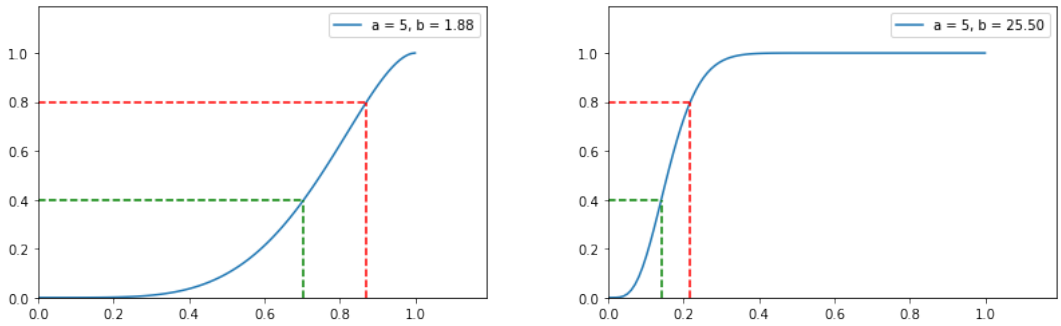

Figure 3.12: CDF of beta distributions in different rounds and their 0.4 quantile and 0.8 quantile. Left panel: #round 13. Right panel: # round 20.

Suppose there are two different molecules, one in the quantile 0.4 of the distribution and the other in the quantile 0.8. Figure 3.12 shows how to determine their efficiency rates at different rounds. In each round, the values of x corresponding to $y = 0.4$ and $y = 0.8$ in the CDF of beta distribution are the efficiency rates of the two molecules. At #round 13, efficiency rates are 0.70 and 0.87 respectively. Moreover, at #round 20, they are 0.14 and 0.22.

Figure 3.13 shows the comparison between without and with memory. Suppose the efficiency rate of molecules is independent. Alternatively, we can say they do not have memory. At every round of amplification, the sample from beta distribution decides their efficiency rate. The left figure shows how it will be under this scenario. The efficiency rate goes up and down rapidly, which is not reasonable. The different property of different molecules results in diversity.

Furthermore, this significant fluctuation does not reflect the idea properly. The correct figure shows the efficiency rate after introducing the idea of memory. It becomes relatively smooth. It keeps the same in the exponential growth phase and drops after entering linear growth.

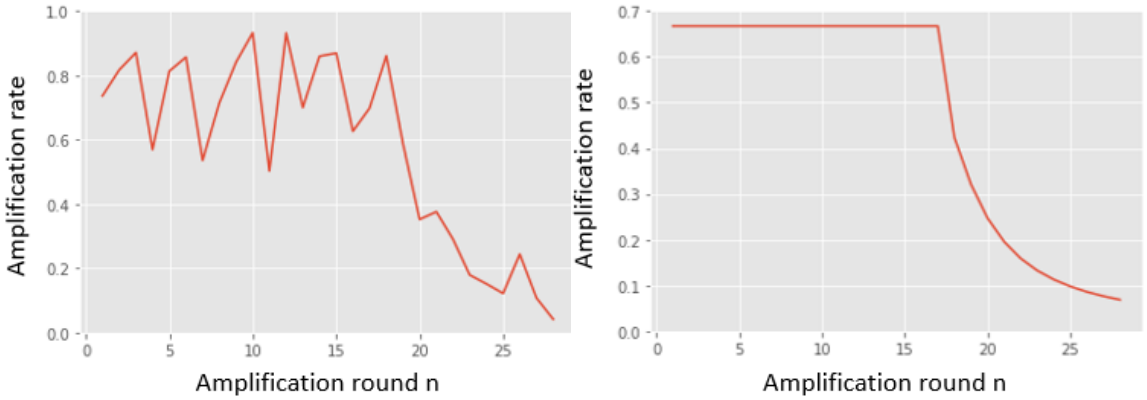

Figure 3.13: The comparison between the change of r_n of a particular cluster of molecules during amplification when without and with memory.

3.3.2 Amplification mutation

Mutations arise as mistakes in DNA replication and when DNA polymerases copy damaged templates[20]. At each replication, mutation may occur, creating a molecule that does not belong to the original cluster. This molecule creates a new cluster and is amplified with other molecules in subsequent experiments. For instance, if a mutation occurs in one amplification, the number of erroneous molecules increases even if it never occurs again. The ratio to the total number is not constant. Therefore, estimating the mutation rate is an arduous task. The simulation and method of estimation will be described later.

3.4 DNA sequencing

Modern sequencing technology decomposes the input genomic DNA (or, in some cases, reverse transcription RNA) into millions of nucleotide sequences, which is also called reading. The errors caused by experiments are so stubborn that despite the continuous improvement of sequencing technology nowadays, the data produced by these technologies are still biased by introducing random and systematic errors[21].The error rate in sequencing is typically 0.1% - 1% for each base pair sequenced[22].

3.4.1 Error correction

In bioinformatics, edit distance is used to eliminate the error caused by sequencing error[14]. The edit distance quantifies the different degrees of the two strings. It calculates the minimum number of operands required to convert one string to another. The Levenshtein distance is one kind of edit distance used in our simulation. It allows deletion, insertion and substitution. Suppose we have two molecules a and b. Their length is $|a|$ and $|b|$ respectively. The Levenshtein distance between a and

b is noted as $lev(a, b)$.

$$lev(a,b) = \begin{cases} |a|, & |b| = 0, \\ |b|, & |a| = 0, \\ lev(tail(a), tail(b), & a[0] = b[0] \\ 1 + min \begin{cases} lev(tail(a), b) \\ lev(a, tail(b)) \\ lev(tail(a), tail(b) \end{cases}, & otherwise \end{cases} \tag{3.33}$$

where the $tail()$ of some string x is a string of all but the first character of x, and $x[n]$ is the nth character of the string x, counting from 0. Figure 3.14 illustrates

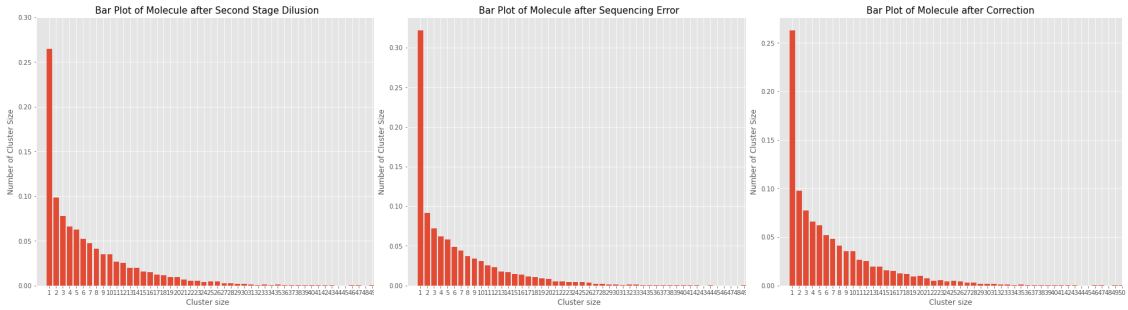

Figure 3.14: Left panel is the frequency diagram without introducing sequencing error. The middle panel is the on with error and the right panel is after correction with edit distance equal to 1.

the result of one of our simulations. When we introduce sequencing error, the percentage of singleton increases dramatically. However, the difference compared to the first graph's data is tiny after correction. The bioinformatics way of handling the sequencing error has an excellent effect. As a result, we will not pay attention to sequencing errors in the following discussion.

3.5 Error in experiment

Errors in NGS are caused by various factors, including DNA damage, errors caused by polymerase during library construction, and sequencer read errors[16, 17, 18]. The error occurs when, during amplification or sequencing, it changes a base pair in the UMI so that the molecule no longer belongs to the original cluster. The different polymerases in the experiment showed different errors in the results[19]. The polymerase used in this experiment is Platinum SF, and for each base pair, the probability of error is 0.178%. The length of UMI is 12 so that the total error of the whole experiment is

$$TotalError = 1 - (1 - 0.178\%)^{12} = 2.12\% \tag{3.34}$$

After discussion with researchers and the fact we introduced before, the sequencing error is about 0.1%. So the percentage of molecules with error,notated as M_e before

sequencing, is

$$M_e + (1 - M_e) * [1 - (1 - 0.1\%)^{12}] = TotalError$$

so that

$$M_e = 0.91\% \tag{3.35}$$

In the simulation, the percentage of molecules with error will be compared with 0.91%.

4

Methods

This section explains the methods applied in this project. Section 4.1 describe the characteristics of the lab data. The application of the algorithm in Section 3.1 and the implementation of amplification rules in the PCR barcoding and amplification processes are described in Section 4.2 and Section 4.3. The simulation of DNA sequencing is shown in Section 4.4.

4.1 Data description

The data is from Sahlgrenska Center for Cancer Research in Gothenburg. For each set of data, it is obtained from 7750 double-stranded DNA molecules after the process of barcoding, amplification and sequencing. Each set of data contains molecule cluster size and the number of these cluster sizes. There are five kinds of molecules and two experimental setups. The same experiment is repeated three times for each kind of molecule. The only difference is the experimental setups. One is PCR barcoding cycle $t = 3$, PCR amplification cycle $x = 28$ and the other is PCR barcoding cycle $t = 5$, PCR amplification cycle $x = 26$.

First, the data distribution is observed by drawing histograms for a preliminary analysis. The horizontal coordinate is cluster size, and the vertical coordinate is the number of cluster sizes. The bar charts in the first line is at $t = 3, x = 28$ and in the second line is at $t = 5, x = 26$. For each column, it represents a molecule. From left to right, they are TP53_374_358, TP53_202_B, TP53_153_168, TP53_098_109 and TP1.

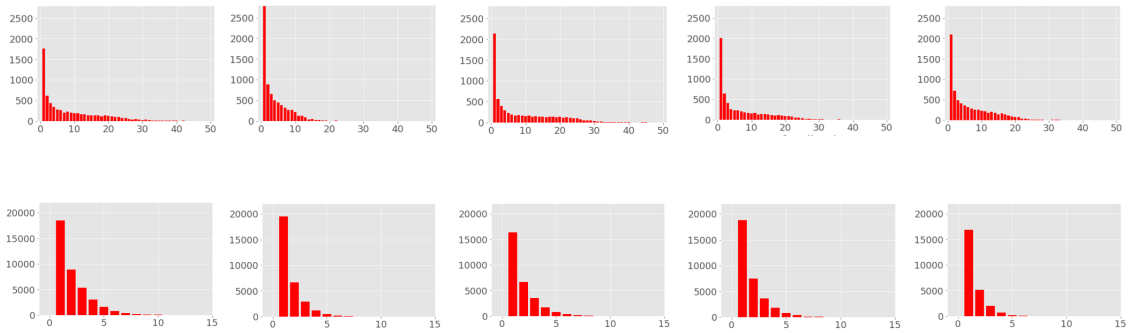

Figure 4.1: The distribution of lab data.

From these bar charts, It can be found that the number of molecules with cluster size 1 far exceeds the other cluster sizes, and the overall trend is decreasing, which is that the greater the cluster size is, the less the number of molecules with that cluster size is. Besides, compared the first line with the second line, the number of clusters at $t = 3, x = 28$ is significantly greater than that at $t = 5, x = 26$, which has a longer tail in the distribution. Here, since cluster size after 50 is small and almost invisible in the bar chart, we only show the data of the first 50, the actual tail of the first situation may reach about 70, while the other is around 10. We are mainly focused on the singletons and the tails.

For each kind of molecule, its total number is obtained by Equation 3.3. The total number at $t = 3, x = 28$ is smaller than the total number at $t = 5, x = 26$ for the first four molecules. However, the last one is exactly the opposite of them. Besides, the first four molecules all belong to the family of TP53. Therefore, we do not consider the last TP1 molecule and remove it.

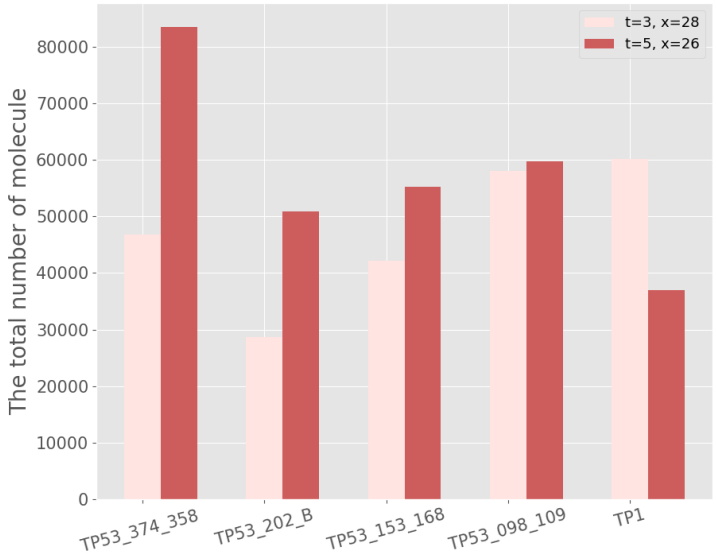

Figure 4.2: The total number of cluster size at $t = 3, x = 28$ and $t = 5, x = 26$.

In order to deeply explore and analyze why such data appears and which parameters are they caused by, we perform the simulation based on Section 3 according to the flow of biological experiments, DNA barcoding, Dilution 1, DNA amplification, Dilution 2 and DNA sequencing, add distributions for the parameters in the model and find the reasonable parameter values by adjustment to get similar results to the experimental data.

4.2 DNA barcoding simulation

4.2.1 Initialization

According to the description of Section 3.1, six kinds of molecules can be obtained in the process of barcoding, which needs to be named and classified in the program to facilitate the subsequent UMI counting. Since UMIs are only added to the different complete strands and influenced by the position of the complete strands, the shapes of the complete strands first need to be distinguished, representing the complete sense strands and the complete nonsense strands respectively. Moreover, in order to precisely determine which molecule each molecule will be amplified into in the PCR barcoding process, the relationship of strand generation is also necessary to be defined, which is shown in Table 4.1 with the types of template strands on the left and the types of new strands which are amplified by the corresponding template strands on the right.

Table 4.1: The representation of the relationship of strands generation in code.

Template chain	*New chain*
000u	001
000d	100
100	201
102	201
001	102
201	102

In Table 4.1, "000u" and "000d" represents the original sense strands and the original nonsense strands, respectively, which are in S_1 and S_2 target. "102" and "201" represents the complete sense strands and complete nonsense strands, respectively, which are in the S_5 and S_6 target. "001" and "100" represents the strands in S_3 and S_4 target respectively.

4.2.2 Level Order Traversal

Before defining the rules of DNA barcoding in detail, the main framework of the program, Level Order Traversal, which is the application of Breadth-First search(BFS), is introduced first. The DNA amplification process for barcoding can be regarded as a binary tree which consists of a root node and many child nodes. In this DNA binary tree, the root node represents the initial DNA molecules, and its child nodes correspond to the new DNA molecules produced during the PCR process. Level Order Traversal traverses the nodes layer by layer according to the tree diagram so that the UMI can be counted for subsequent DNA amplification and sequencing.

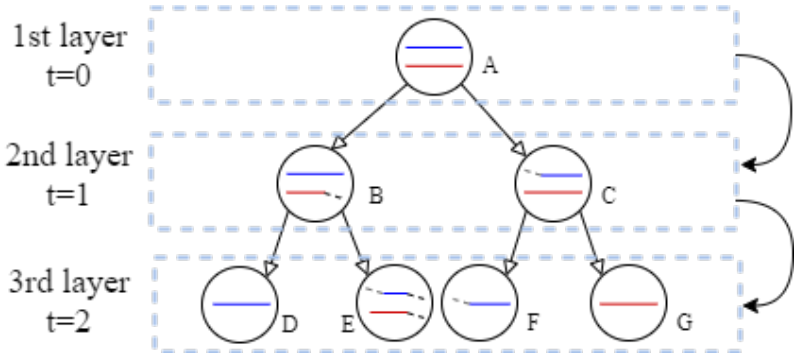

Figure 4.3: An example of Level Order Traversal for DNA barcoding

It looks the same as the Breadth-First search. However, the format of result required by the Level Order Traversal needs to distinguish each level to know which cycle each molecule is in, that is, to return a two-dimensional array $[[A][BC][DEFG]]$ which is different from that of BFS. The traversal result of BFS is a one-dimensional array $[ABCDEFG]$, and each layer cannot be distinguished.

The specific process is to put the root node, that is, the initial DNA template, into the double-ended queue firstly, where the elements can be popped from both ends and insertion and deletion operations can be performed on both sides. For each cycle, DNA molecules in a layer are stored in the queue, and one DNA molecule is removed at a time for PCR barcoding according to the DNA barcoding function. The new DNA molecules are put into the left and right nodes if the amplification is successful. If either strand is not successfully amplified, the strand is placed directly into the node, and the other node holds the next generation of DNA amplified from the other strand, which is the situation of node D and node E in Figure 4.3. If both strands are not successfully replicated, then the two single strands are placed in the left and right nodes, respectively, which is the situation of node F and node G in Figure 4.3. For the node with only one strand, one of its child nodes will be empty. Therefore, the nodes that are not empty are placed in the queue, and the operation is repeated for the next cycle. The cycle t is used to limit the number of iterations so as to traverse the nodes layer by layer. In other words, t is the layer.

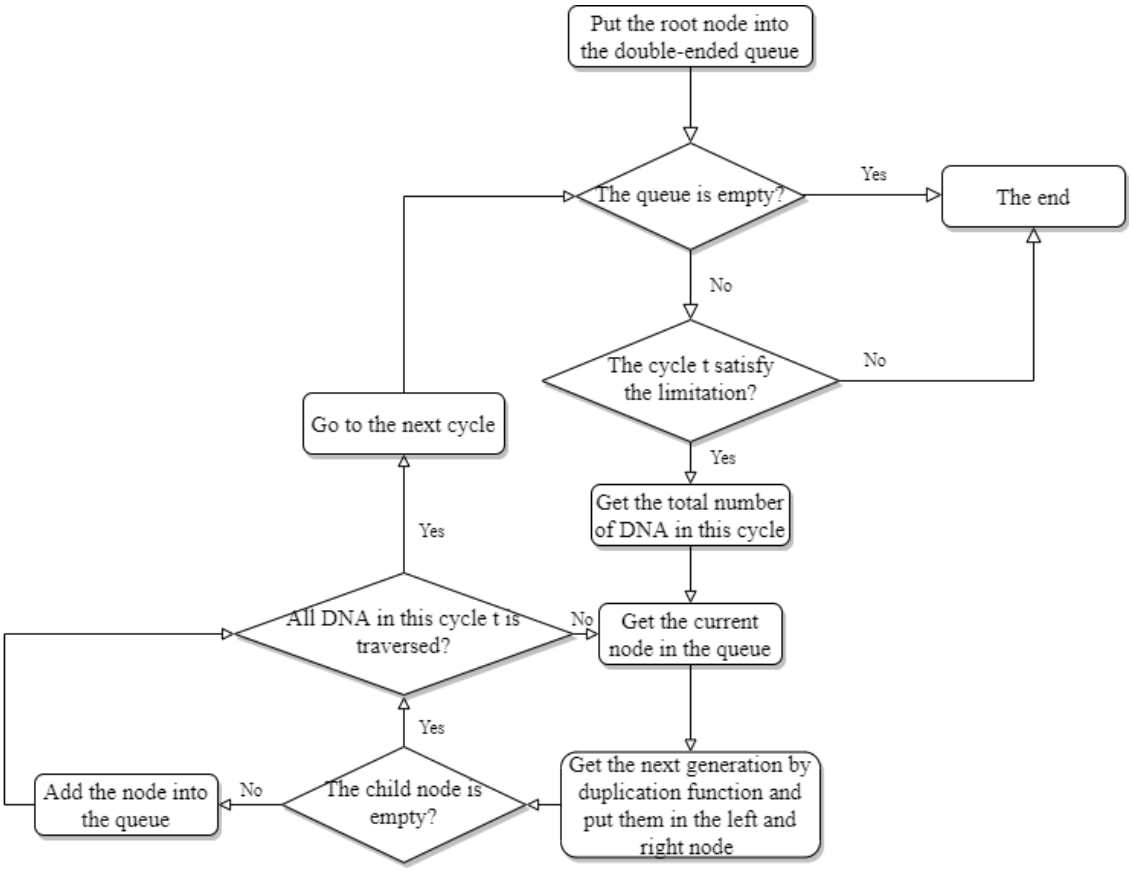

Figure 4.4: The process of Level Order Traversal

4.2.3 DNA barcoding rules

In this section, DNA barcoding rules will be introduced explicitly that only focus on counting UMIs on the complete strands according to the biological law. Each sense strand and nonsense strand are amplified according to different amplification efficiencies determined by the strand type. The strands with the same name mean containing the same UMI. Further judgment is made on the strand to give it a specific barcoding rule: the complete strands must be copied to obtain the complete strands. Therefore, the first thing needed to judge is whether the molecular strand input by the program is a complete strand or not. Secondly, since the position of the strand will also affect the type of the new strand, it is crucial to determine its position after judging whether it is a complete strand. Then a new strand containing UMI with a specific name and shape can be obtained according to the naming rule, numbers, lower or upper letters.

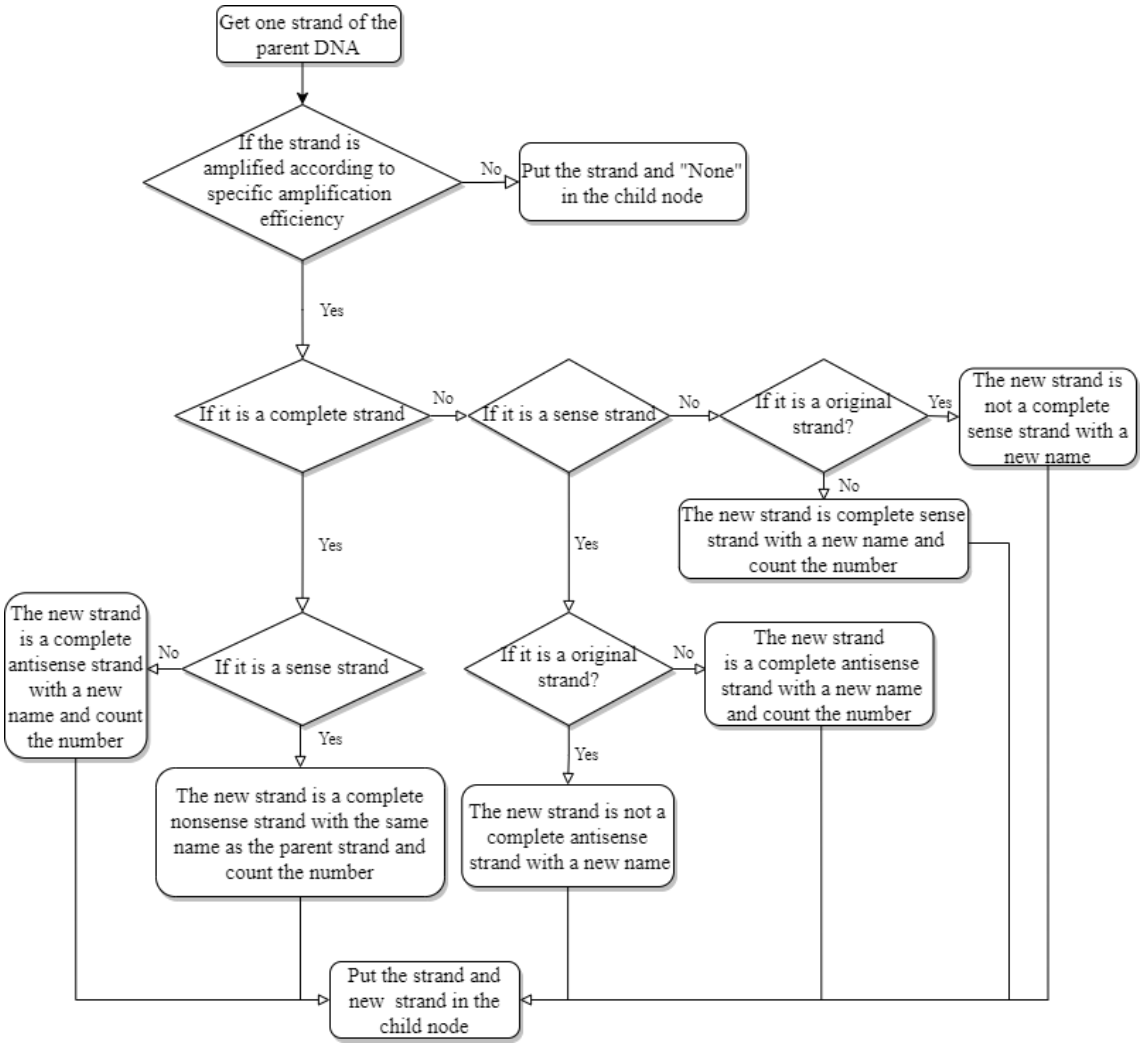

Figure 4.5: The process of DNA barcoding rules

4.3 DNA amplification simulation

Below is the whole process of the DNA amplification simulation. There is preliminary work and subsequent processing before and after amplification. They will be introduced together.

Figure 4.6: The Whole Amplification Process

In Figure 4.6, there are two stages of dilution. Cluster sizes after dilution are randomized based on equation 3.21.

4.3.1 Amplification rules

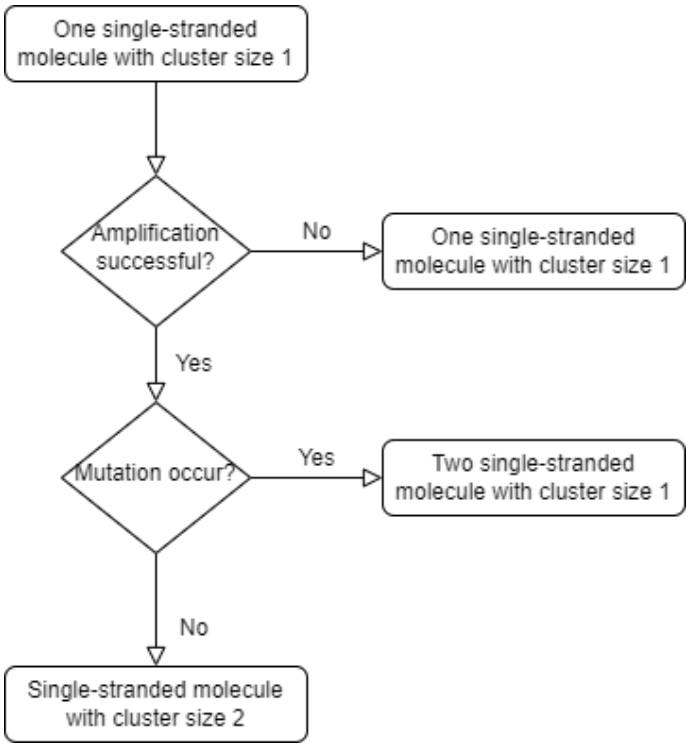

Figure 4.7: The simple process of amplification rules for one molecule

This section focus on amplification rules. Figure 4.7 illustrates the process for a single molecule. This is a simple demo diagram. The cluster size will be enormous in the actual simulation. For each cluster, in each round, we assign a different efficiency rate to evaluate whether the amplification is successful or not randomly. If amplification is unsuccessful, this molecule will not produce anything in this round. Otherwise, we will continue to determine if this round will generate an error randomly. If a mutation occurs, a new cluster with size one is created. If there is no mutation, a copy of the initial molecule is generated, so the size of the initial cluster will be doubled.

In the simulation, the amplification efficiency rate will keep the same in the beginning and then constantly change with the number of rounds and the environment. The beta distribution expresses the difference in the efficiency rate of different molecules. In the beginning, every cluster is arranged with a random number in $[0, 1)$ as their quantile over all the molecules. Before each round of amplification, the total number of molecules z in the environment is calculated.

If z is more diminutive than K, the distribution is fixed. Otherwise, $r = \frac{K}{K+z}$ is calculated as the mean of efficiency rate, where K is the Michaelis-Menten constant. At each round, a beta distribution with a mean equal r is used to generate different efficiency rates for different clusters based on their quantile generated in the very beginning.

4.4 DNA sequencing simulation

In order to maximize the restoration of the experimental process, we considered the sequencing error as well as the correction algorithm.

The process in the sequencer can be viewed as a single round PCR to read out the detailed sequence of molecules. We did not simulate this process but only introduced sequencing error. Since the PCR in the sequencer is used to read nucleotides, the error will be for each nucleotide and not for the whole molecule. The length of UMI in the experiment and simulation is 12. Hence each cluster is randomly assigned a specific sequence. Sequences are randomly generated from ATCG. Each nucleotide will have a 0.1% probability of changing into one of the other three, with the same probability of changing into each one to simulate sequencing error.

In bioinformatics, edit distance is used for the correction algorithm of the sequencing error and mutation error. The most common use in the lab is merging two clusters with an edit distance of 1. Furthermore, this is also used in the simulation.

5

Results

The results based on manual tuning hyperparameters are shown in this section. The comparison of the results of theory to simulation and the distribution of final data reproduced according to the actual data are depicted in Section 5.1 and Section 5.2 separately. Besides Section 5.2 also describes the suitable parameter values chosen.

5.1 The number of cluster size

If the results of the theoretical formula and the simulation results can be cross-verified, we have confidence that the simulation of the PCR barcoding process is reasonable. In turn, it can be considered that the data obtained by barcoding simulation with 7750 molecules as the original data are reasonable, guaranteeing the correctness of subsequent experiments. First, we will focus on the number of cluster sizes.

In Figure 5.1, we compare the sizes of each cluster size in theory and simulation of PCR barcoding cycle $t = 3$, PCR amplification cycle $x = 28$ on the left and PCR barcoding cycle $t = 5$, PCR amplification cycle $x = 26$ on the right. The solid line and the thicker dashed line show the simulated results and the theoretical results, respectively. Four different colour lines are drawn, representing the results in different efficiency rates (r_1, r_2, r_3, r_4): blue $(1, 1, 1, 1)$, green $(0.6, 0.7, 0.8, 0.9)$, red $(0.4, 0.6, 0.7, 0.9)$, black $(0.3, 0.4, 0.5, 0.9)$.

It is evident that the two lines almost overlap, illustrating that the theory and the simulation are mutually validated. In addition, for different efficiency rates, the larger the efficiency rate is, the larger the total number of cluster sizes. As the number of cycles increases, leading to more amplified molecules, the faster the total number of cluster sizes increases. Therefore, the PCR barcoding process simulation can be considered a rational experiment.

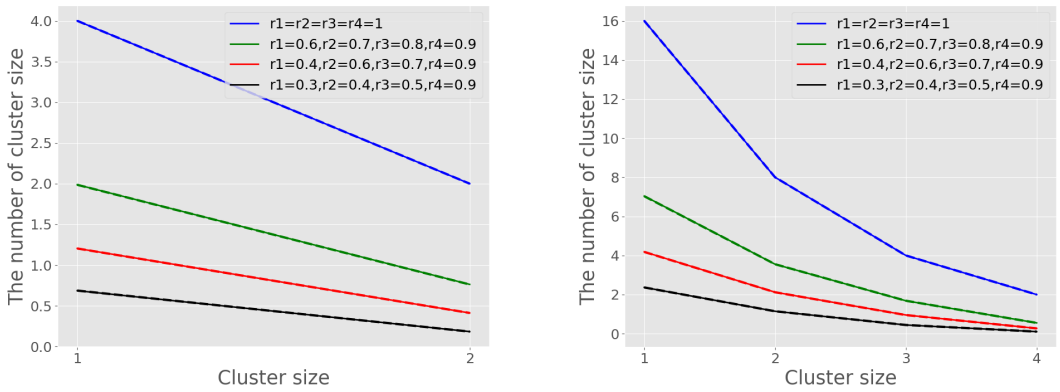

Figure 5.1: The comparison of theory results to simulation results in the number of cluster size.

5.1.1 The proportion of cluster size

As stated in the equation of section3, when t tends to infinity, $Z_t(m)/Z_t$ is an exponential distribution 2^{-m}. Therefore, we conjecture that this relationship applies to the incomplete duplication with four efficiency rates. Figure 5.2 shows the change of proportion of cluster size at $t = 3$, $x = 28$ and $t = 5$, $x = 26$ for the same four different efficiency rates above. We can notice that the curves converge as t becomes larger. This is in accordance with the above hypothesis. For $t = 5$, $x = 26$, it is remarked that the four solid lines almost overlap, and their distribution tends to be the same, which means the proportional distribution of cluster size does not have a big difference for these efficiency rates. Since our project has nine parameters, making the model tuning really difficult, our results mainly focus on the distribution of cluster size rather than the number itself. Then the influence of four barcoding efficiency rates r_1, r_2, r_3 and r_4 on the final results can be ignored for the time being and just select the appropriate efficiency rates and keep them constant. In this project, $r_1 = 0.6, r_2 = 0.7, r_3 = 0.7$ and $r_4 = 0.9$ according to the biological law and our own trails.

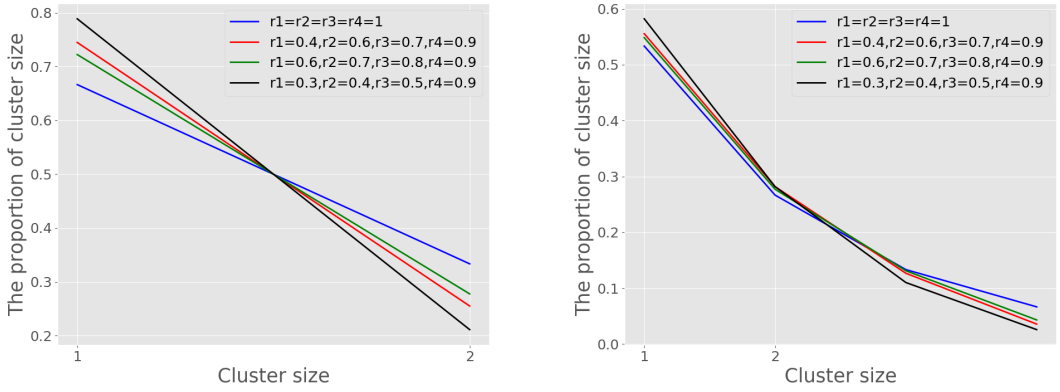

Figure 5.2: The proportion of cluster size with different efficiency rates.

5.1.2 The final DNA barcoding results

The molecules with a UMI are our primary focus. Therefore, the results of the barcoding simulation, that is, the numbers of each cluster size of 7750 molecules after barcoding, are saved to a CSV file and served as initial data in the following steps.

Table 5.1: An example of the final results in PCR barcoding process at $t = 3$.

Cluster size	1	2
$Molecule1$	3	2
$Molecule2$	1	1
$Molecule3$	4	2
⋮	⋮	⋮
$Molecular7750$	2	0

The data format is shown in Table 5.1. For example, the numbers of cluster size 1 and 2 of Molecule 1 are 3 and 2 separately.

5.2 Lab results vs Simulation results

Read the CSV file of the barcoding results and perform the simulation by the DNA amplification model for these molecules with UMI according to Section 4. In order to make the distribution of results similar to that of the actual experimental results, we obtain the following reasonable results by adjusting α in Beta distribution, The average DNA amplification rate for exponential growth r, the second dilution rate d_2, the environmental parameter K and mutation rate M these five parameters. The proportion distribution is shown in these figures: lab results in the first line and simulation results in the second line. The four sets of bar charts from left to right are the results of TP53_374_358, TP53_202_B, TP53_153_168 and TP53_098_109 separately.

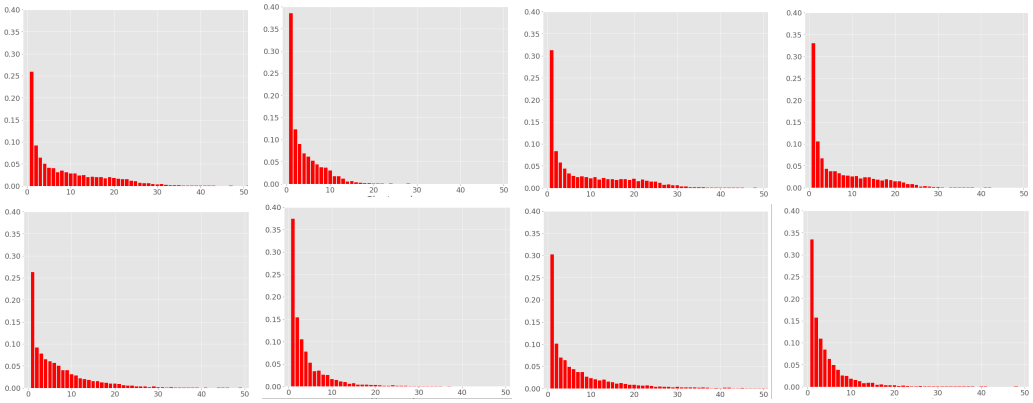

Figure 5.3: The proportion comparison of lab results to simulation results at $t = 3, x = 28$.

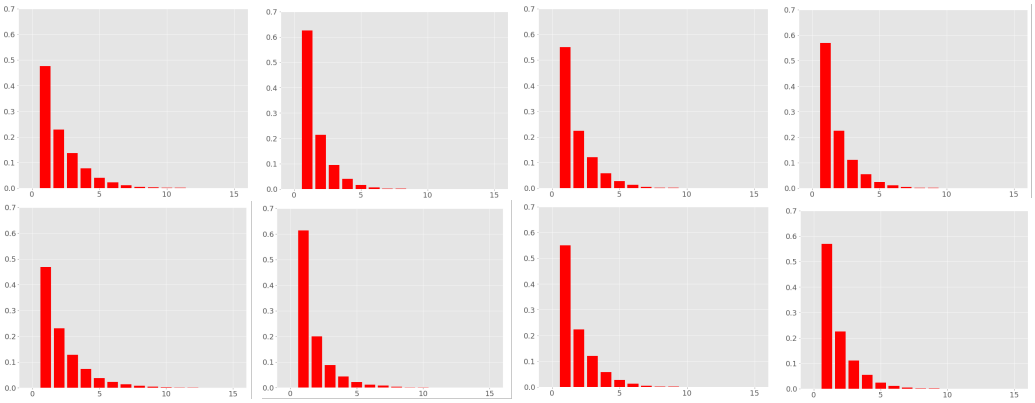

Figure 5.4: The proportion comparison of lab results to simulation results at $t = 5, x = 26$.

We can observe that the simulation results are similar to the real lab results. They should be kept constants for K and M according to the experimental setting, which is $K = 10^8$ and $M = 4 \times 10^{-4}$. Besides, α, r and d_2 should be the same for the same molecule in the two cases of PCR barcoding cycle $t = 3$, DNA amplification cycle $x = 28$ and PCR barcoding cycle $t = 5$, DNA amplification cycle $x = 26$. Therefore, the choice of other parameters for each molecule by tuning the model is shown in Table 5.2.

Table 5.2: The parameters for each molecule.

	α	r	d_2
$TP53\_374\_358$	12	0.9	24000
$TP53\_202\_B$	5	0.75	75000
$TP53\_153\_168$	10	0.8	35000
$TP53\_098\_109$	26	0.7	30000

6
Discussion

6.1 Choice of Parameters

The choice of the parameters is mainly based on the advice from laboratory researchers and our trials and errors.

6.1.1 α, r and d_2

The analysis focuses on how these three parameters change the singleton distribution and tail length, as shown in these three figures. The left y axis is the tail length, and the right y axis is the singleton distribution. Since the trend is the same, we will take $t = 3, x = 28$ as an example.

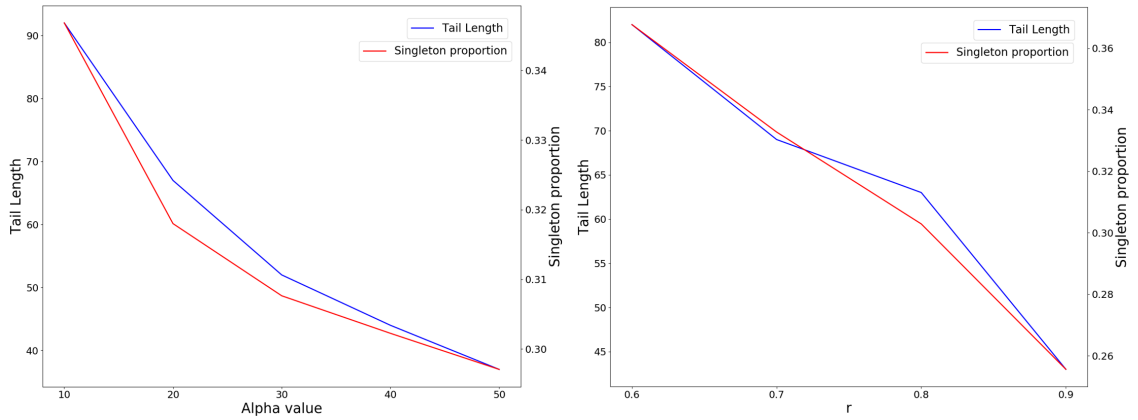

Figure 6.1: The influence on singletons distribution and tail length as α and r increases.

As α increases from 10 to 50 and other parameters do not change, the variance of the beta distribution will become smaller, leading to a decrease in the proportion of singleton and a shorter tail.

As r increases from 0.6 to 0.9 and other parameters remain the same, the amplification rate increases and more molecules with a UMI are produced. Under the same dilution rate, the cluster size of molecules gets bigger. Thus, the proportion of singleton gets smaller. At the same time, according to equation 3.29, when r becomes more significant, the variance of beta distribution becomes smaller, which

can also result in a shorter tail. The cluster size in the final result will show more aggregation.

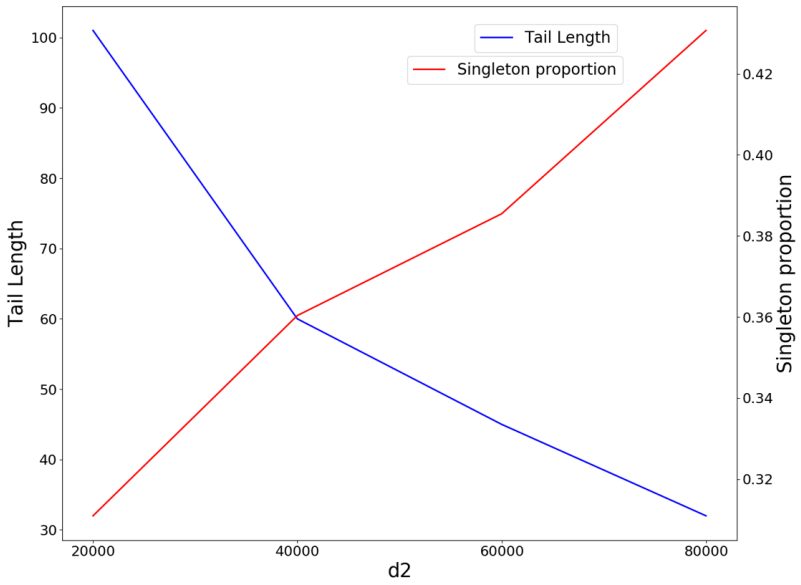

Figure 6.2: The influence on singletons distribution and tail length as d_2 increases.

Then as d_2 increases from 20000 to 80000 and other parameters remain unchanged, the probability of each molecule being selected becomes smaller. Thus it is easier to generate more small cluster sizes and more singletons. Since a large cluster size is less likely to be generated, the tail is shorter.

Table 6.1: The influence on singleton distribution and tail length by only increasing α, r or d_2.

	Singleton	*Tail*
α	*Decrease*	*Decrease*
r	*Decrease*	*Decrease*
d_2	*Iecrease*	*Decrease*

6.1.2 Environment parameter K

For K, it is a hidden parameter in the experiment. Although it represents the support of the environment for amplification, it is not a parameter that can be measured in reality with experimental instruments. It can only be speculated based on the final result. $K = 10^8$ is adopted throughout the simulation.

There is a possibility that it could be a more considerable number. 10^9 or even more significant numbers are all possible. However, the simulation needs to be judged whether amplification is successful and whether mutation will occur. As K increases, the number of operations in the program will grow exponentially, which

is a significant disadvantage for tuning the parameters afterwards. So larger K is not adopted in the simulation.

Besides, we think changing K will only make little difference to the final result. It is because since the dilution rate is also a hyperparameter, a more enormous value of dilution rate can be applied to eliminate the influence of larger K.

However, it is undeniable that K is also treated as the saturation point. It has some influence on the shape of the final distribution. However, due to the limit of computational power, this question is not studied.

As to the reason why K is fixed for all molecules, it represents the support from the environment. All molecules amplify in the same environment at the beginning. However, different molecules may consume the environmental resource at a different rate during amplification. Nevertheless, since it is very complex in biology and the exact sequence is not studied in the simulation, it is assumed to be the same.

6.1.3 Mutation rate M

It is hard to calculate in theory how large the mutation rate will cause how much error in the final result. In our simulation, the mutation rate is fixed as 4×10^{-3}.

The theoretical total error rate caused by mutation has been calculated in equation 3.34. The error under the choice of mutation rate in the simulation is slightly bigger than the theory.

Advice from the lab researchers is that the mutation rate should be around 10^{-6}, which is much smaller compared with our choice. This might be caused by choice of K. If K is more prominent in the simulation, a lower mutation rate may also create a similar result.

The mutation rate is all the same among all molecules. It is because that mutation is mainly caused by polymerase error. Furthermore, the same polymerase is used across all experiments. So we treat them the same as all molecules.

6.2 Difference between $t = 3$ and $t = 5$

It is easy to observe that for $t = 3$, the figure shows a longer tail compared with $t = 5$. The reason for this phenomenon is discussed below.

With the limit of the environment, the population size will grow slowly after reaching the saturation point. However, because of the fact that $t = 5$ has two more barcoding round, it has more clusters. The total number of molecules grows much slower than the number of clusters. So the cluster size for each cluster is minor than $t = 3$. Thus, for $t = 5$, under the same dilution rate, the result is more aggregation

in smaller cluster size.

For example, on average, $t = 3$ has 100 clusters and 10000 molecules. So the size for each cluster is 100. On the other hand, $t = 5$ has 200 clusters and 15000 molecules. Each cluster size has 75 on average. Under the same dilution rate, such as 10, $t = 3$ will retain 10 for each cluster size, and $t = 5$ will reserve 7.5 on average. As a result, cluster sizes after dilution will be smaller for $t = 5$, thus having a shorter tail.

6.3 The memory mechanism

Memory is the representation of molecule property. Since diversity in amplification comes from the different properties of molecules, the property is intrinsic and will not change across the amplification. The property's amplification efficiency is influenced and related to its previous performance. Therefore memory is introduced. It is natural because significant fluctuation for a particular molecule cluster is not realistic.

However, this memory mechanism needs proof from the biological experiment. Moreover, there could be a better idea to realize the thought of memory. For example, a simple Markov Chain is used in the simulation. It will be better if a more complicated one can be constructed. Nevertheless, the difficult point is how to maintain the whole distribution for all molecules, that is, the beta distribution. Another idea is that maybe every cluster has its own internal parameters for amplification, so they do not need to sample from the overall distribution.

6.4 Future work

In the first place, for the current model, we need to align the metric of the simulation results with the actual data. Here, due to the time limitation, the high complexity of biological experiments with invisible internal operations and many parameters, we only focus on the distribution of cluster size rather than the number itself. Besides, the parameters are chosen chiefly based on manual tuning. Afterwards, we need to find a better way to optimize the parameters rather than manually adjusting them to make the results more accurate. This will involve some knowledge of operations and optimization.

Secondly, our model involves many assumptions. Although all of them are natural and reasonable based on biological knowledge, evidence is needed to prove these assumptions. As a result, we hope to conduct some biological experiments like real-time fluorescence quantitative PCR. Conducting biological experiments can be very labour-intensive and costly, but it is the best way to understand the whole biological process.

Last but not least, some situations in the experiment could be more complicated than the simulation. Take K as an example; we assume K is the same for all clusters

and molecules. However, it is possible that although all molecules are in the same environment, the support provided is not the same for all molecules. An exaggerated example is that the same rainforest can support three monkeys but only one gorilla. The difference in the experiments will not be that great, but each cluster may have its own parameters. This will make the whole program much more complicated.

We believe that for this paper, there are still some features in the data that have not been reproduced. There are some data that we have excluded that the lab researchers tell us are normal. Future work should focus on how to include these cases.

7
Conclusion

With the premise that Unique Molecular Identifiers(UMI) has been shown to be effective in distinguishing sequencing errors from true variants in DNA sequencing, it makes sense to do the simulation for UMI counting in DNA sequencing to get the similar results to the real laboratory data by introducing a DNA barcoding algorithm and building a mathematical model for invisible DNA sequencing experiments to find reasonable parameter values.

In our project, we first introduce and implement a DNA barcoding algorithm based on Level Order Traversal with four parameters of DNA barcoding efficiency rates r_1, r_2, r_3 and r_4 according to different strand generations. The simulation results are similar to the theory results calculated by the equations in an early version of Serik paper [8]. It can be found that these four parameters have a less influence on the proportion of cluster size than the number of cluster size which makes us use the proportion as a metric for the result comparison to reduce the difficulty of tuning the model.

A mathematical model of DNA amplification assuming molecules with memory and based on the growth patterns of population and molecular diversity has been built with three parameters to tune, α, the α in the beta distribution of Equation 3.27, r, the average of DNA amplification rate for exponential growth and d_2, the second dilution rate and two parameters with constant values, K, the environment in DNA amplification and M, the mutation rate in DNA amplification. The appropriate parameter values for each molecule are found. Besides, it is observed that only increasing α or r makes singletons distribution and tail length both show a decreasing trend. However, only increasing d_2 makes tail length rise but singleton distribution fall.

In summary, the simulation based on the DNA barcoding algorithm and DNA amplification mathematical model can reproduce similar results to real laboratory results of TP53 molecular family with specific parameter values.

Bibliography

[1] Sanger, F., Nicklen, S., Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. Proceedings of the national academy of sciences, 74(12), 5463-5467.

[2] Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S., Taipale, J. (2012). Counting absolute numbers of molecules using unique molecular identifiers. Nature methods, 9(1), 72-74.

[3] Filges, S., Yamada, E., Ståhlberg, A., Godfrey, T. E. (2019). Impact of polymerase fidelity on background error rates in next-generation sequencing with unique molecular identifiers/barcodes. Scientific reports, 9(1), 1-7.

[4] Ståhlberg, A., Krzyzanowski, P. M., Jackson, J. B., Egyud, M., Stein, L., Godfrey, T. E. (2016). Simple, multiplexed, PCR-based barcoding of DNA enables sensitive mutation detection in liquid biopsies using sequencing. Nucleic acids research, 44(11), e105-e105.

[5] Haccou, P., Haccou, P., Jagers, P., Vatutin, V. A., Vatutin, V. (2005). Branching processes: variation, growth, and extinction of populations (No. 5). Cambridge university press.

[6] KlMMEL, M., Axelrod, D. E. (2002). Branching Processes in Biology. Interdisciplinary Applied Mathematics 19.

[7] Pflug, F. G., von Haeseler, A. (2018). TRUmiCount: correctly counting absolute numbers of molecules using unique molecular identifiers. Bioinformatics, 34(18), 3137-3144.

[8] Sagitov, S., Ståhlberg, A. (2022). Counting unique molecular identifiers in sequencing using a decomposable multitype branching process with immigration. arXiv preprint arXiv:2205.06405.

[9] Bertoin, J. (2009). The structure of the allelic partition of the total population for Galton–Watson processes with neutral mutations. The Annals of Probability, 37(4), 1502-1523.

[10] Weiss, G., von Haeseler, A. (1995). Modeling the polymerase chain reaction. Journal of computational biology, 2(1), 49-61.

[11] Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., Gilad, Y. (2008). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. Genome research, 18(9), 1509-1517.

[12] Jagers, Peter, and Fima Klebaner. "Random variation and concentration effects in PCR." Journal of Theoretical Biology 224.3 (2003): 299-304.

[13] Lalam, N., Jacob, C., Jagers, P. (2004). Modelling the PCR amplification process by a size-dependent branching process and estimation of the efficiency. Advances in Applied probability, 36(2), 602-615.

[14] Smith, T., Heger, A., Sudbery, I. (2017). UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. Genome research, 27(3), 491-499.

[15] Aird, Daniel, Michael G. Ross, Wei-Sheng Chen, Maxwell Danielsson, Timothy Fennell, Carsten Russ, David B. Jaffe, Chad Nusbaum, and Andreas Gnirke. (2011). Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. Genome biology, 12(2), 1-14.

[16] Fox, E. J., Reid-Bayliss, K. S., Emond, M. J., Loeb, L. A. (2014). Accuracy of next generation sequencing platforms. Next generation, sequencing applications, 1.

[17] Robasky, K., Lewis, N. E., Church, G. M. (2014). The role of replicates for error mitigation in next-generation sequencing. Nature Reviews Genetics, 15(1), 56-62.

[18] Brandariz-Fontes, C., Camacho-Sanchez, M., Vila, C., Vega-Pla, J. L., Rico, C., Leonard, J. A. (2015). Effect of the enzyme and PCR conditions on the quality of high-throughput DNA sequencing results. Scientific reports, 5(1), 1-5.

[19] Filges, S., Yamada, E., Ståhlberg, A., Godfrey, T. E. (2019). Impact of polymerase fidelity on background error rates in next-generation sequencing with unique molecular identifiers/barcodes. Scientific reports, 9(1), 1-7.

[20] Marnett, L. J., Plastaras, J. P. (2001). Endogenous DNA damage and mutation. TRENDS in Genetics, 17(4), 214-221.

[21] Mitchell, K., Brito, J. J., Mandric, I., Wu, Q., Knyazev, S., Chang, S., ... Mangul, S. (2020). Benchmarking of computational error-correction methods for next-generation sequencing data. Genome biology, 21(1), 1-13.

[22] Salk, J. J., Schmitt, M. W., Loeb, L. A. (2018). Enhancing the accuracy of next-generation sequencing for detecting rare and subclonal mutations. Nature Reviews Genetics, 19(5), 269-285.

[23] Saiki, R. K., Gelfand, D. H., Stoffel, S., Scharf, S. J., Higuchi, R., Horn, G. T., ... Erlich, H. A. (1988). Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. Science, 239(4839), 487-491.

A

Appendix 1

A.1 DNA barcoding code

```
from os import name
import random
import collections
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Store data in the structure of tree.
# [updown] indicates the two chains of DNA.
# Left and right are the new generation of DNA copied with up and down chains
#as templates respectively.
class TreeNode:
    def __init__(self, val, left=None, right=None):
        self.val=val
        self.left = left
        self.right = right

#102:the complete upper chain 201:the complete lower chain.
perfect_chain = ['102','201']
#Define the corresponding relationship of the chain, the key on the left is the type
#of template chain, and the value on the right is the type of new chain.
struct_table = {'000u':'001',
                '000d':'100',
                '100':'201',
                '102':'201',
                '001':'102',
                '201':'102',
                }

chain_num = {}

number_index = 1
letter_index = 'C'

```

```
35
36 #This function is used to get the new chain, including its name and type.
37 def createStruct(name,chain,position):
38     global number_index,letter_index
39
40     if chain in perfect_chain:
41         if position=='up':
42             new_down = str(name)+'_'+struct_table.get(chain)
43             chain_num[str(name)] = chain_num[str(name)]+1
44             return new_down
45         else:
46             new_up =  str(number_index)+'_'+
47                       struct_table.get(chain)
48             chain_num[str(number_index)] = 1
49             number_index+=1
50             return new_up
51
52     else:
53         if position=='up':
54             if str.isalpha(name):
55                 new_down = letter_index+'_'+
56                            struct_table.get(chain)
57                 letter_index = chr(ord(letter_index)+1)
58             else:
59                 new_down = name[:-1]+'_'+
60                            struct_table.get(chain)
61                 chain_num[name[:-1]] = 1+chain_num[name
                       [:-1]] if name[:-1] in chain_num.keys()
                       else 1
62             return new_down
63
64         else:
65             if struct_table.get(chain) not in perfect_chain:
66                 new_up = str(number_index)+'*'+'_'+
67                          struct_table.get(chain)
68             else:
69                 new_up =  str(number_index)+'_'+
70                           struct_table.get(chain)
71                 chain_num[str(number_index)] = 1
72             number_index += 1
73             return new_up
74
75 #This function generates two new DNAs based on the parent DNA,r1,r2,r3 and r4.
76 #Save the new DNAs to left and right node.
77 def duplicate(parent:TreeNode,r1,r2,r3,r4):
78     [up,down] = parent.val
```

```
79      if up == 'None':
80          down_name,down_chain = down.split('_')
81          r = random.random()
82          p = r1
83          if down_chain=='201':
84              p=r4
85
86          if down_chain=='001':
87              p=r2
88
89          if r <=p:
90              right = TreeNode([createStruct(
91                      down_name,down_chain,'down'),down])
92          else:
93              right = TreeNode(['None',down])
94
95          return None, right
96
97      elif down == 'None':
98          up_name,up_chain = up.split('_')
99          r = random.random()
100         p = r1
101         if up_chain=='102':
102             p=r4
103
104         if up_chain=='100':
105             p=r3
106
107         if r <=p:
108             left = TreeNode([up,createStruct(
109                     up_name,up_chain,'up')])
110         else:
111             left = TreeNode([up,'None'])
112
113         return left, None
114
115     else:
116         up_name,up_chain = up.split('_')
117         down_name,down_chain = down.split('_')
118
119         r = random.random()
120         p = r1
121         if up_chain=='102':
122             p=r4
123
124         if up_chain=='100':
```

```
125             p=r3
126
127         if r <=p:
128             left = TreeNode([up,createStruct(
129                     up_name,up_chain,'up')])
130         else:
131             left = TreeNode([up,'None'])
132
133         r = random.random()
134         p = r1
135         if down_chain=='201':
136             p=r4
137
138         if down_chain=='001':
139             p=r2
140
141
142         if r <=p:
143             right = TreeNode([createStruct(
144                     down_name,down_chain,'down'),down])
145         else:
146             right = TreeNode(['None',down])
147
148         return left, right
149
150 #This function uses the level order traversal framework, which is to traverse layer
151 #by layer according to the tree diagram.
152 def levelOrder(time,
153                root: TreeNode,r1=0.6,r2=0.7,r3=0.7,r4=0.9):
154     if not root:
155         return []
156
157     res = []
158     queue = collections.deque()
159     queue.append(root)
160
161     while queue and time>0:
162         time-=1
163         m = len(queue)
164         ans = []
165         for i in range(m):
166             tmp = queue.popleft()
167             ans.append(tmp.val)
168             tmp.left,tmp.right=duplicate(tmp,r1,r2,r3,r4)
169
170             if tmp.left:
```

```
                    #print(tmp.left.val,end=" ")
                    queue.append(tmp.left)
                if tmp.right:
                    #print(tmp.right.val,end=" ")
                    queue.append(tmp.right)
            res.append(ans)
        return res

#t=3
tmp = 7750
size = 7750

t = 3 #or t=5
Z_count = np.zeros(t-1)
#t=3
once_Z3 = np.zeros((size,2))
#t=5
#once_Z5 = np.zeros((size,4))

a = 0
for i in range(tmp):
    chain_num = {}
    number_index = 1
    letter_index = 'C'

    root = TreeNode(['A_000u','B_000d'])
    ans = levelOrder(t,root)

    b = 0
    for j in range(t-1):
        a = collections.Counter(chain_num.values())[j+1]
        Z_count[j] = Z_count[j] + a
        b = b + a
        #t=3
        once_Z3[i][j] = a
        #t=5
        #once_Z5[i][j] = a

#t=3
dataframe = pd.DataFrame(once_Z3)
dataframe.to_csv("new_once_Z3.csv",index=False,sep=',')


#t=5
#dataframe = pd.DataFrame(once_Z5)
#dataframe.to_csv("new_once_Z5.csv",index=False,sep=',')
```

A.2 DNA amplification code

```
import csv
import numpy as np
import matplotlib.pyplot as plt
from collections import defaultdict
from scipy.stats import beta
from scipy.stats import binom
import copy

#read the result of barcoding from csv file
def read_barcode(barcode_file):
    pcr = []
    with open(barcode_file)as f:
        f_csv = csv.reader(f)
        header = next(f_csv)
        for row in f_csv:
            pcr.append(row)
    pcr = [[float(x) for x in row] for row in pcr]
    pcr = np.sum(pcr, axis=0)
    pcr = [int(x) for x in pcr]
    return pcr

#The two stages dilution based on smapling from binomial distribution
def dilusion_1(barcoded, proportion):
    count = 1
    mol_dilution = [0] * 10
    for i in barcoded:
        success_flag = np.random.rand(i, count) <
                       1 / proportion
        success_num = np.bincount(np.sum(
                      success_flag, axis = 1))
        for i in range(1,len(success_num)):
            mol_dilution[i - 1] += success_num[i]
        count += 1
    return mol_dilution

def dilusion_2(after_amp, prop_1):
    mol_dilution = [0] * 200
    num1 = binom.rvs(after_amp, 1 / prop_1)
    for i in num1:
        if i != 0:
            mol_dilution[i - 1] += 1
    return mol_dilution

```

```
# Amplification simulation based on parameter in beta distribution, K, mutation rate and r
def amplify_beta_fixa(after_dil, a, x, K, mutation, mean):
    result = []
    count = 1
    for i in after_dil:
        for j in range(i):
            result.append(count)
        count += 1
    percentile = list(np.random.rand(len(result)))
    length = len(result)
    for i in range(x):
        summ = sum(result)
        if summ < K:
            b = (1 - mean) * a / mean
        else:
            b = summ / K * a
        means.append(a/(a+b))
        print(i + 1, b)
        bottle_neck = beta.ppf(percentile, a, b)
        r_1.append(bottle_neck[0])
        success_flag = binom.rvs(result, bottle_neck)
        error_flag = binom.rvs(success_flag, mutation)
        result = list(np.array(result) +
                      np.array(success_flag) -
                      np.array(error_flag))
        error_num = sum(error_flag)
        result.extend([1] * error_num)
        percentile_error = np.random.rand(error_num)
        percentile.extend(percentile_error)
    return result


pcr1 = read_barcode('new_once_Z3.csv')
pcr2 = read_barcode('new_once_Z5.csv')
d11 = dilusion_1(pcr1, 3)
d12 = dilusion_1(pcr2, 3)
alpha = 12
K = 10e7
mutation = 4 * 10e-4
mean = 0.9
r_1 = []
means = []
a = amplify_beta_fixa(d11, alpha, 28, K, mutation, mean)
# a = amplify_beta_fixa(d12, alpha, 26, K, mutation, mean)
d2 = dilusion_2(a, 40000)
```

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY