

MONASH UNIVERSITY

A Statistical Model to Explore Clonal Heterogeneity in Cancer

by

Zhaoxiang Cai

A thesis submitted in partial fulfillment for the
degree of Bachelor of Computer Science with Honours

in the
Faculty of Information Technology
Clayton School of Information Technology

June 2014

Declaration of Authorship

I, Zhaoxiang Cai, declare that this thesis titled, ‘A Statistical Model to Explore Clonal Heterogeneity in Cancer’ 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 thesis 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 thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis 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:

“If the facts don’t fit the theory, change the facts.”

Albert Einstein

MONASH UNIVERSITY

Abstract

Faculty of Information Technology
Clayton School of Information Technology

Honours Degree of Bachelor of Computer Science

by [Zhaoxiang Cai](#)

Cancer (tumor malignancy) has been seen as one of the most serious diseases for decades. Tumor contains different mutations among the tumor cells referred as tumor heterogeneity. Cancer treatment and therapy development are tumor heterogeneity dependent. For more accurate treatment and therapy, it is necessary to cluster the mutations of a tumor sample to find the subpopulations (a set of tumor cells having similar mutations). In recent years, many researches have done studies with the assumption that – mutations of a subpopulation do not overlap with others. But in the real world, mutations of a subpopulation may overlap with other. By considering overlapping mutations among different subpopulations, we have designed a statistical model: Het-FHMM, based on Factorial Hidden Markov Model (FHMM). We have carried out several experiments on our model with synthetic data and compared with an approximation of Pyclone, which is another state-of-the-art statistical model to analyze tumor heterogeneity. It has been found that our method outperformed Pyclone in accuracy. Our proposed method works on mutations overlapping among the subpopulations which help to identify subpopulation of a tumor sample more accurately, although the experiments were only done on the shortened version of synthetic genome data.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to Dr. Gholamreza Haffari and Professor Ann Nicholson, my supervisors, for their steady encouragement, patient guidance and enlightening discussions throughout my honours year. Without their help, the work presented here would have not been possible. I also wish to express my appreciation to my parents. You have been always support me on whatever I would like to pursue. I wish this thesis could be seen by my father in heaven and make him proud.

Contents

| | |
|---|-------------|
| Declaration of Authorship | i |
| Abstract | iii |
| Acknowledgements | iv |
| List of Figures | viii |
| List of Tables | ix |
| Abbreviations | x |
| Physical Constants | xi |
| Symbols | xii |
| | |
| 1 Introduction | 1 |
| 1.1 Preamble | 1 |
| 1.2 The Research Problem | 2 |
| 1.3 Objectives, Research Scope and Assumptions | 2 |
| 1.4 Thesis Organization | 4 |
| | |
| 2 Background | 5 |
| 2.1 Heterogeneity in Cancer | 5 |
| 2.2 Data | 8 |
| 2.2.1 Mutations | 9 |
| 2.2.2 DNA Sequencing | 10 |
| 2.2.3 Array Comparative Genomic Hybridization | 12 |
| 2.3 Graphical Models | 12 |
| 2.3.1 General Graphical Models | 13 |
| 2.3.2 Hidden Markov Model | 14 |
| 2.3.3 Factorial Hidden Markov Model | 15 |
| 2.4 Statistical Models for Studying Heterogeneity | 16 |
| 2.4.1 HMM based Models | 17 |
| 2.4.1.1 HMM-Mix: A Model for clustering aCGH data | 17 |

| | | |
|----------|--|-----------|
| 2.4.1.2 | HMMCNA: Copy Number Analysis based on aCGH Data | 19 |
| 2.4.1.3 | PennCNV: Copy Number Analysis based on B Allele Frequency and log R Ratio | 20 |
| 2.4.2 | Non-HMM based Models | 22 |
| 2.4.2.1 | THetA: Copy Number Analysis using Maximum Likelihood Mixture Decomposition Problem | 22 |
| 2.4.2.2 | PhyloSub: Hierarchical Bayesian Model Designed for Phylogeny Reconstruction | 23 |
| 2.4.2.3 | PyClone: Bayesian Model using Beta-Binomial Emission Densities | 25 |
| 2.4.3 | Summary for Related Work | 26 |
| 2.5 | Summary | 27 |
| 3 | Studying Heterogeneity in Cancer using FHMMs | 29 |
| 3.1 | Input and Output | 29 |
| 3.2 | Our Model: Het-FHMM | 30 |
| 3.3 | Inference | 35 |
| 3.3.1 | Exponentiated Gradient Descent | 36 |
| 3.3.2 | Gibbs Sampling | 37 |
| 3.4 | Time Complexity | 38 |
| 3.5 | Strengths, Limitations and Discussion | 38 |
| 4 | Experiments, Results & Discussion | 41 |
| 4.1 | Overview | 41 |
| 4.2 | Evaluation Mechanism | 43 |
| 4.2.1 | Percentage of Correctly Predicted Genotypes | 43 |
| 4.2.2 | Negative log-likelihood | 43 |
| 4.3 | Experiments on Synthetic Data | 44 |
| 4.3.1 | Generation of Synthetic Data | 44 |
| 4.3.2 | Finding the Best Configuration of the Inference Algorithm | 45 |
| 4.3.2.1 | Experiment 1: Find the best configuration for the convergence criteria | 45 |
| 4.3.2.2 | Experiment 2: Find the best configuration for the number of switches between Gibbs sampling and EG | 47 |
| 4.3.2.3 | Section Summary | 49 |
| 4.3.3 | Comparison between Different Number of Chains | 49 |
| 4.3.4 | Inference Performance on Different Numbers of Samples Considered | 51 |
| 4.3.5 | Comparison between Different Instantiations of the Model | 53 |
| 4.3.6 | Summary | 54 |
| 4.4 | Comparison with the baseline | 54 |
| 4.4.1 | Result of Baseline Comparison | 56 |
| 4.5 | Hardware Information & Computational Time | 57 |
| 4.6 | Summary of All Experiments | 57 |
| 5 | Conclusion and Future Work | 59 |
| 5.1 | Contribution | 59 |
| 5.2 | Future Work | 60 |

| | |
|----------------------------------|-----------|
| A Gibbs Sampling | 61 |
| B Full Experiment Results | 62 |
| Bibliography | 68 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Tumor Heterogeneity Comparison | 3 |
| 2.1 | Clonal heterogeneity illustration | 6 |
| 2.2 | Monoclonality abd polyclonality | 6 |
| 2.3 | Clonal heterogeneity illustration 2 | 7 |
| 2.4 | Impact of clonal heterogeneity on treatment | 8 |
| 2.5 | DNA structure | 9 |
| 2.6 | Mutations | 9 |
| 2.7 | Copy number variation | 10 |
| 2.8 | NGS data illustration | 11 |
| 2.9 | Array CGH data | 12 |
| 2.10 | Graphical model | 13 |
| 2.11 | Hidden Markov Model | 15 |
| 2.12 | HMM rain example | 15 |
| 2.13 | Factorial HMM | 16 |
| 2.14 | The graphical model of HMM-Mix | 18 |
| 2.15 | Illustration of HMMCNA | 20 |
| 2.16 | THetA algorithm overview | 23 |
| 2.17 | Phylogeny tree generated by PhyloSub model | 24 |
| 2.18 | The graphical model of PhyloSub | 24 |
| 2.19 | PyClone graphical model | 25 |
| 3.1 | Graphical model of Het-FHMM | 32 |
| 3.2 | Direction of the Chain | 39 |
| 3.3 | Second order HMM | 40 |
| 4.1 | The flow chart for the experiment | 42 |
| 4.2 | -LogLikelihood plots for Experiment 1 | 46 |
| 4.3 | -LogLikelihood plots for Experiment 2 | 48 |
| 4.4 | Line charts for different number of chains | 50 |
| 4.5 | Line chart of different number of samples | 51 |
| 4.6 | Line chart of different instantiations of the model | 53 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Six-state genotype space | 21 |
| 2.2 | Summary of related work | 27 |
| 3.1 | Sample input file | 30 |
| 3.2 | Sample output file | 30 |
| 3.3 | Summary of variables in Het-FHMM | 32 |
| 3.4 | Genotype variable state space | 33 |
| 4.1 | Data table for 20×1000 and 20×5000 | 47 |
| 4.2 | Data table for 20×1000 and 100×200 | 48 |
| 4.3 | Data table for different number of chains | 50 |
| 4.4 | Result table for different numbers of samples | 52 |
| 4.5 | Data table for different chains | 52 |
| 4.6 | Data table for different tissues | 54 |
| 4.7 | Result of Baseline Comparison | 56 |
| 4.8 | Computational time | 57 |
| B.1 | Data table 1 | 62 |
| B.2 | Data table 2 | 62 |
| B.3 | Data table 3 | 63 |
| B.4 | Data table 4 | 63 |
| B.5 | Data table 5 | 64 |
| B.6 | Data table 6 | 65 |
| B.7 | Data table 7 | 66 |
| B.8 | Data table 8 | 67 |

Abbreviations

| | |
|-------------|--|
| BAF | B A llele F requency |
| CNV | C opy N umber V ariation |
| NGS | N ext G eneration S equencing |
| aCGH | a rray C omparative G enome H ybridization |
| HMM | H idden M arkov M odel |
| FHMM | F actorial H idden M arkov M odel |
| GD | G radient D escent |
| EG | E xponentiated G radient descent |
| MCC | M onash C ampus C luster |

Physical Constants

Length of Human Genome $L = 3.2 \times 10^9$ bases

Symbols

| | |
|--------------------|---|
| a, d | reference count (matches) |
| b | B allele count (mismatches) |
| l | log ratio of tumor-normal read depth |
| N | read depth |
| G | genotype variable |
| X, Z | probe state |
| \mathbf{S}, ϕ | cellular prevalence |
| c | copy number |
| r | allele ratio |
| t | tumor purity |
| S, Y | observation variables |
| K | number of clones in the tumor |
| T | number of genotype variables in one chain |
| | |
| χ | Sample space |
| ψ | genotype combination |

*Dedicated to my parents,
who have supported me so long
and so far from home.*

Chapter 1

Introduction

1.1 Preamble

Cancer is the name for a set of diseases which describes the situation when cells lose their control on self-replication, and invade and transport to other tissues by the lymph system or blood [1]. If the spread of uncontrolled cells continues, cancer can lead to death with high probability. The incidence rate for all kinds of cancers in the United States is 550.7 (male) and 419.3 (female) per 100,000 people from 2005-2009 [2]. To show cancer is a worldwide issue, we can look at some local statistics for Australia. The mortality of cancer in 2012 is 221.7 and 137.6 per 100,000 people for male and female respectively, which is quite similar to the values in the U.S. [3].

Research studies have been focusing on various aspects of cancer, including the cause of cancer, the mechanism of cancer, how cancer evolves, different properties of cancer and how we could deal with cancer. With the aid of computational power, each field in cancer study has developed greatly. Computers have been playing one of the most important roles in modern cancer studies and in other relevant biology fields. For example, DNA sequencing data is usually used as input for study of cancer, and high-throughput DNA sequencing techniques such as next-generation sequencing (NGS) relies on computational algorithms in order to achieve reliable results. Details of sequencing techniques will be discussed separately in later sections in Chapter 2.

As the role of information technology becomes increasingly important in biology researches, one specific research field called “Bioinformatics” also emerged in 1970 [4]. This thesis falls within this area, as it aims at using computational power and statistical models to analyze a phenomenon in cancer called “clonal heterogeneity”.

1.2 The Research Problem

In order to discuss clonal heterogeneity in cancer, we first need to know how cancer works in a high level. Human body is composed of cells. Each cell contains the same DNA strand which carries genetic information of one specific person. This DNA strand not only carries the genetic information, but also controls the behavior of the cell in which the DNA strand resides. Therefore, once the DNA strand is mutated i.e. the content of DNA is changed, the behavior of the cell is also affected. Normal cells have a regulated cell cycle, which controls how the cell grows, splits and dies. Most normal cells have a fixed length of life and will be replaced by new cells.

When the DNA strand in a cell is mutated, the normal cell cycle process may be changed so that the cell gains extra advantages and lives longer than other cells. When there are many mutated cells that become uncontrollable in growth, they are then called “tumor cells”. A **tumor** can be benign or malignant. **Benign** tumors cells only grow locally but **malignant** tumor cells invade other tissues and transport to other sites as well. **Cancer** is the term that is used to describe the disease when malignant tumor cells invade other normal cells so that functionalities of normal cells are lost [1].

The process by which cells get mutated and accumulated is complex. There are various factors which may affect the probability of mutations’ occurring. Hence it is reasonable that different cells may acquire different mutations. Some cells in the tumor may share some common properties or the same mutations, and these cells are considered to be in the same **subpopulation** or **clone**. Since different clones react differently to drugs, we need to identify and target all the clones for a comprehensive treatment of cancer. Otherwise some clones still remain untreated, and they will generate more tumor cells and eventually malignant tumors grow again. **Clonal heterogeneity** refers to the phenomenon whereby different clones co-exist in a tumor.

The aim of this research project is to develop a computational model that automatically analyzes data from a tumor and identify clones that exist in the tumor. By identifying all the clones in the tumor, medical researchers can make the therapy more thorough and thus prevent the relapse of cancer.

1.3 Objectives, Research Scope and Assumptions

The objective of this thesis is to present a novel computational model that can be used to automatically identify what clones or cell subpopulations there are in a given tumor sample. Our model takes next-generation sequencing data as input, and outputs

the inferred clones. More details about the data will be discussed in Section 2.2. We evaluate the model based on synthetic data using different evaluation mechanisms. We compare this model with a baseline model to show the strengths and weaknesses of our new model.

With respect to the research scope, research on tumor heterogeneity consists of inter-tumor heterogeneity and intra-tumor heterogeneity lines of research. Figure 1.1 shows the difference. **Inter-tumor heterogeneity** research deals with the difference between the tumors from different people that grow from the same tissue site. For example, both person A and person B have tumors in the liver tissue, but the mutations acquired in the tumors may be different. On the other hand, **intra-tumor heterogeneity** focuses on the difference between mutations within one human body. As Figure 1.1 shows, three different parts of one tumor consists of cells from different clones. In this thesis, we focus our research scope on the intra-tumor heterogeneity only, and all the phrases “clonal heterogeneity” refer to intra-tumor heterogeneity.

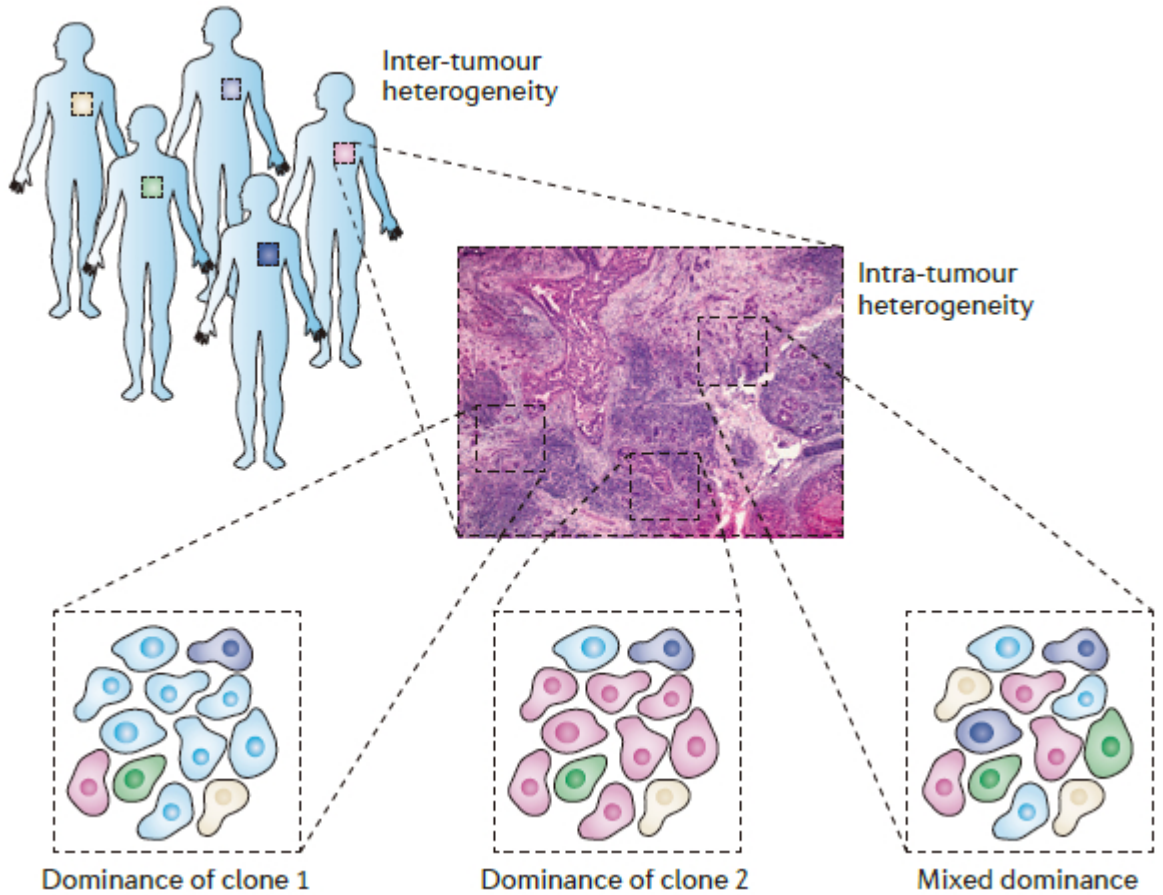


FIGURE 1.1: The difference between intra-tumor heterogeneity and inter-tumor heterogeneity [5]

We also limit our project to artificially generated data for all the experiments at this

stage. One reason is that although we have access to the sequencing data from 21 breast cancer patients, we do not have the ground truth of those data. The evaluation becomes extremely difficult without the ground truth. In addition, using synthetic data allows us to focus on the accuracy of the model for inferring the composition of the tumor. Although we consider the computational complexity to some extent, it is a secondary consideration behind investigating the potential of the model and we assume that there will be an inference algorithm that makes the model applicable to the real data. Therefore all experiments are done on a shortened genome (3×10^4 or 3×10^5 for different experiments).

1.4 Thesis Organization

This thesis is organized into another four chapters. In Chapter 2, we discuss the current status of researches on related topics involved in analyzing the clonal heterogeneity. We present our new computational model based on factorial Hidden Markov Model in Chapter 3. After discussing the new model, in Chapter 4 we present the experiments carried out for finding the best configuration as well as the evaluation of the new model. The comparison with a baseline model is also discussed in Chapter 4. Finally, in Chapter 5 we summarize our conclusions regarding the proposed new model and outline potential future work.

Chapter 2

Background

In this chapter, we present necessary background knowledge required to understand the research problem. We also survey some state-of-the-art statistical models that are proposed by researches for solving the problem of clonal heterogeneity. Section 2.1 provides a thorough illustration of the problem of clonal heterogeneity and some studies that support the existence of it. Various kinds of data that can be used for analyzing clonal heterogeneity are described in Section 2.2. In Section 2.3 we present a background on graphical models, which are used as a unified language in our literature review on this problem in Section 2.4.

2.1 Heterogeneity in Cancer

The basic mechanism of cancer was introduced in Section 1.2, so here we provide a more detailed description of clonal heterogeneity in cancer. Clonal heterogeneity refers to the idea that a solid tumor is composed of malignant cells with different genomic aberrations. In other words, not all tumor cells carry the same set of mutations, instead, the tumor includes different categories (aka clones, subpopulations or clusters) of cells where the cells belonging to each category carry similar mutations. Figure 2.1 provides a simplified illustration of what clonal heterogeneity looks like. In this figure, tumor cells that belong to the same clone have the same color, where these cells in the same clone share the same mutations.

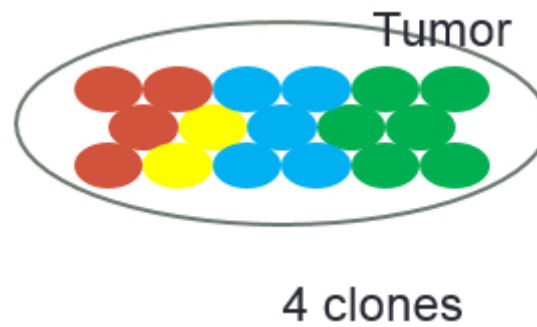


FIGURE 2.1: Simplified illustration of clonal heterogeneity in cancer, with 4 clones in a single tumor.

Intra-tumor heterogeneity is a fairly young topic in cancer research, first formally mentioned in 1978 by Heppner[6]. He found that different subpopulations of a tumor had variance in sensitivity to drugs, which was an indication of intra-tumor heterogeneity. Similar to many other brilliant new ideas, the concept of “intra-tumor heterogeneity” was not accepted at first, because cancer was believed to be “mono-clonal”. As Figure 2.2 shows, the “monoclonal model” is based on the assumption that all of the cancer cells originate from one single cell, whereas the “polyclonal model”, which is the basis for intra-tumor heterogeneity”, postulates that there are multiple origin cells so that there are different subpopulations existing in the tumor (represented by different colors in the figure).

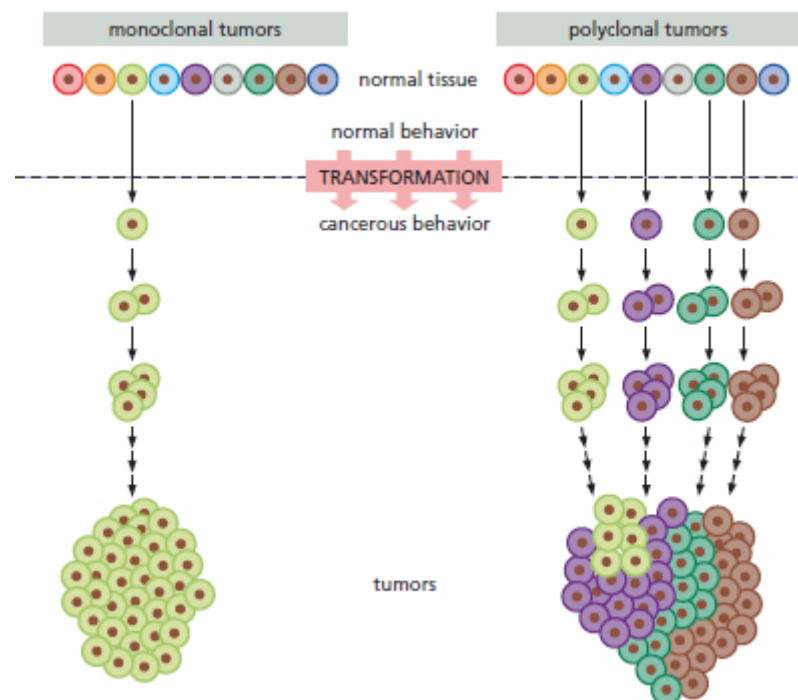


FIGURE 2.2: Illustration of monoclonality versus polyclonality of tumors [7]. In “monoclonal model”, all tumor cells are believed to originate from one single cell, whereas in “polyclonal model” there is a great extent of heterogeneity.

Figure 2.3 provides a more complex but more realistic illustration of the progression of cancer¹ in the monoclonal (2.3A) and polyclonal (2.3B) model. In the monoclonal model, mutations keep accumulating but the key is that the new cells will contain all the previous mutations. However in the polyclonal model, not all mutations are inherited by all the newer cells.

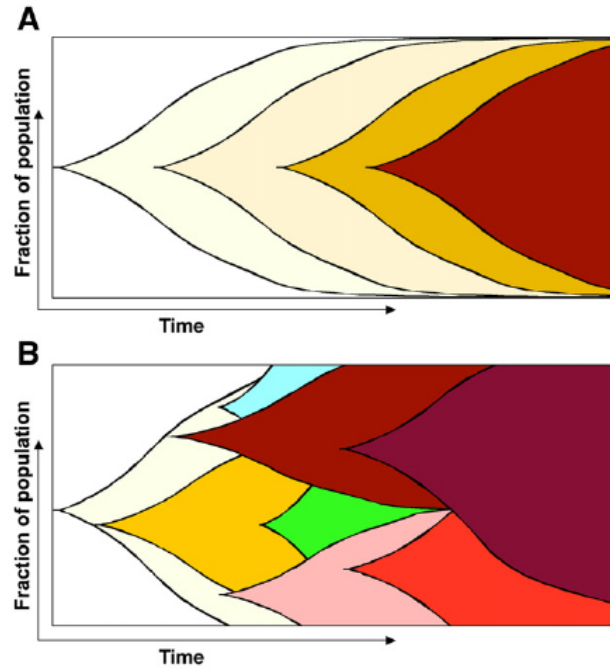


FIGURE 2.3: More Complex illustration of monoclonality versus polyclonality of tumors [7]

Although the truth about the mechanism remains controversial, there are several studies that strongly support the polyclonal model hypothesis [8–10]. In these studies, the core idea is to cut the whole tumor into different physical segments and analyze each of these segments, which is referred to as “multiregion sequencing”. All of the studies show that genomic aberrations acquired in different regions of the same tumor have some differences, thereby supporting the hypothesis of polyclonal model and the existence of clonal heterogeneity.

The importance of studying clonal heterogeneity can be seen in [11]. The study hypothesized that the drug resistance that appears during cancer treatment is caused by intra-tumor heterogeneity, supporting this hypothesis from the perspective of Darwinian evolution. Figure 2.4 illustrates the basic idea of this study. Genetic heterogeneity keeps increasing before the drug treatment starts. Once the drug is used, most of the tumor cells are killed. However, cells from one specific clone are immune to the drug and

¹The progression of cancer refers to the history of the development of a cancer.

therefore “selected” by the drug treatment. This clone then quickly reestablishes clonal heterogeneity again.

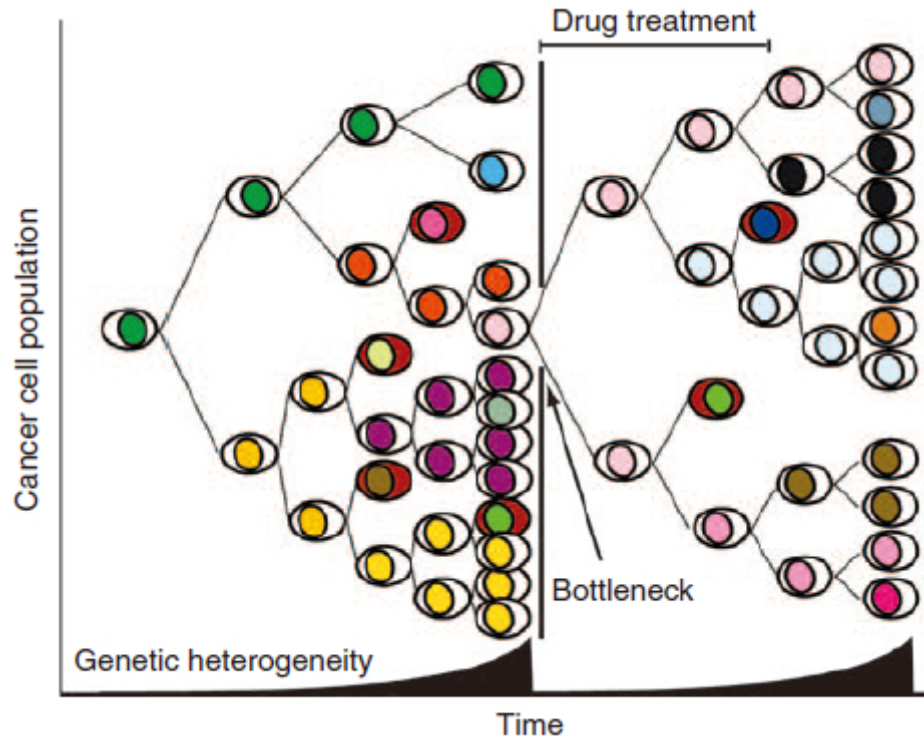


FIGURE 2.4: Schematic view of tumor heterogeneity during tumor progression and treatment [11]. It shows how clonal heterogeneity poses difficulties on therapy development.

Then the study provides several evidences supporting their hypothesis, including clone selection caused by cytotoxic chemotherapy and targeted drugs. To conquer this drug resistance problem, they emphasizes the importance of measuring heterogeneity using comprehensive tumor sampling techniques in order to identify all clones in the tumor.

Nowadays, researchers can make use of computational techniques to analyze heterogeneity [9, 12–16]. The increase in popularity of this research area comes from its crucial role in developing therapies. Since different clones have differences in sensitivity to drugs, in order to develop reliable and effective individualized therapies, clonal heterogeneity must be thoroughly analyzed.

2.2 Data

In this section, we describe various kinds of data that can be used to analyze heterogeneity in cancer, which include DNA sequencing data and array Comparative Genome

Hybridization data. First, we present some relevant background knowledge that is necessary to understand those data.

2.2.1 Mutations

In previous sections, we mentioned that cancer is believed to be caused by DNA mutations, but we have not yet formally introduced what DNA is composed of and what kinds of mutations there are. **DNA** is a molecule on which there is a sequence of nucleotides. Genetic information that is used for development and functioning of human body is encoded in DNA, and it contains two strands. There are four types of **nucleotides** or **bases** (A,T,C,G) in the helix structure of DNA, A always matches T and C always matches G (base pairing rules). Figure 2.5 shows the DNA structure. When a nucleotide gets mutated, there are three possible ways: one extra nucleotide may be added, one is deleted or the nucleotide is changed to another one. Figure 2.6 shows the three possibilities of mutations.

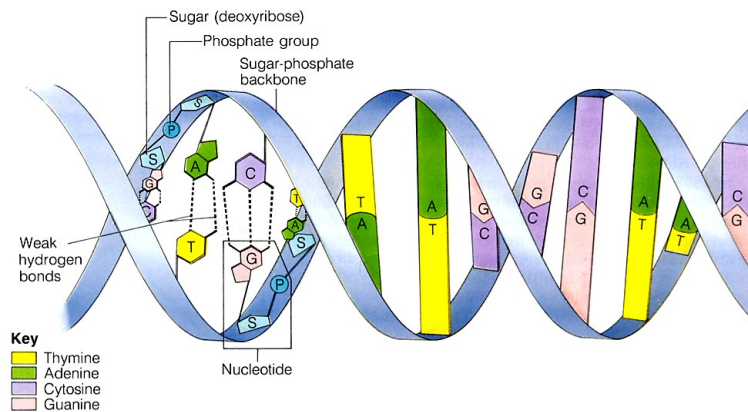


FIGURE 2.5: Each DNA molecule consists of two strands which are complement [17]. There are four types of nucleotides (A,C,T and G) which can sit at each position on the strand.

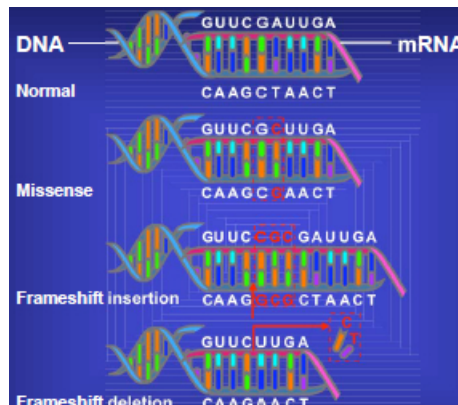


FIGURE 2.6: Three kinds of point mutations [18]. Missense means a single nucleotide is changed to another one.

The three kinds of mutations described above are also called **point mutation**. There is also another category of mutations named chromosome copy number variation (CNV). In contrast to point mutations where a single nucleotide base is altered, **CNV** describes the situation where a segment of DNA sequence is amplified or deleted. For example, a segment of the length 2,000 bases is amplified by having exactly the same sequence right after the normal segment, hence making an extra copy of that DNA segment. This concept is shown in Figure 2.7, where deletion and amplification of section C of DNA are presented. In normal cells, there are **two** copies for each genomic location and therefore for all of A,B,C,D there should be two copies in normal cells. By analyzing the CNV, we can analyze if there is any amplification or deletion of a small segment of DNA, thereby identifying the intra-tumor heterogeneity.

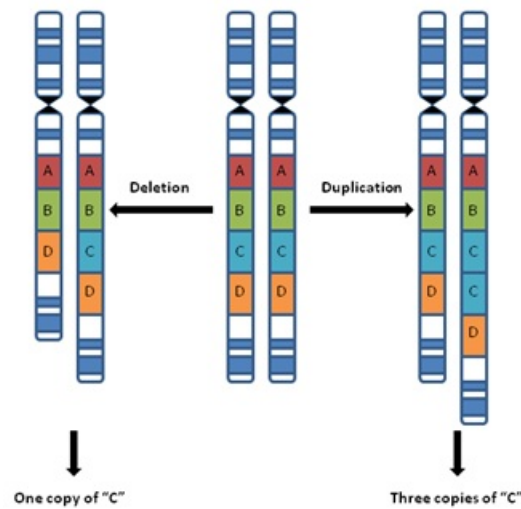


FIGURE 2.7: Copy Number Variation (CNV) illustration [19]. The segment of DNA “C” is deleted or amplified, i.e. the copy number of segment C is changed to 1 or 3 from 2 copies (normal) in this case.

2.2.2 DNA Sequencing

Since we determine clones based on the mutations acquired in cells, if we can determine the exact composition of the DNA strand we can then infer what clones exist in the tumor. The process of identifying the order of nucleotides on the DNA strand is called DNA sequencing. Our project also uses DNA sequencing data as the input to analyze clonal heterogeneity.

One early sequencing technique that is widely used is Sanger sequencing, also called the Chain-termination method [20]. This sequencing method also produced the first human genome in 2001. However, its cost is too much for cancer treatment for most people. After the mid-2000s, with the invention of so-called **next generation sequencing**

(NGS) techniques, the cost of sequencing per genome decreased dramatically from \$100 million in 2001 to \$10,000 in 2011 [21]. In this section, we only focus on NGS, because all the current studies that use DNA sequencing data as input use NGS data.

Next generation sequencing techniques focus on providing high-throughput and low-cost sequencing. Multiple copies of the sample genome are required for NGS and these copies are broken into fragments (or **reads**) of length ranging from 50 bases to 10,000 bases for different platforms. The second step is to determine the nucleotides on each read. Lastly, these sequenced short reads can be used to reconstruct the whole DNA sequence in two ways, either DNA alignment or DNA de novo assembly. In DNA alignment, the short reads are aligned to a reference genome, whereas in DNA assembly, the short reads are assembled together based on the common sequence section at the beginning and end of each read. DNA assembly and alignment are two major research areas that draw many researchers' interest, but since they are not directly related to our thesis, we do not review them in detail.

In this thesis, we rely on the "alignment" approach in putting the "reads" together to reconstruct the genome. Figure 2.8 illustrates how the data look like in detail. There are multiple reads aligned to each position. The total number of reads that cover a single position is called the read depth. In order to analyze the sample tumor, researchers usually compare the reads against a normal DNA sequence as a reference. **Reference count** refers to the number of reads that match the reference DNA. Some studies also use non-matches. Then if there are 2 out of 3 reads that do not match the reference, we say that the **B allele frequency** (BAF) for this position is 2/3. Usually B allele refers to the number of reads that does not match the reference and A allele refers to the number of matches. BAF is used as the input data for most of the studies which use NGS data. Another important data that is used is called **log-R ratio**. It is achieved by running the sequencing twice, one on the normal tissue and one on the normal tissue. Then the log ratio of the read depth of tumor tissue over normal tissue is taken. This ratio reflects the tumor/normal read depth of a single position and therefore it represents the copy number information.



FIGURE 2.8: NGS data illustration. In this picture, short reads are aligned to a reference DNA (DNA alignment).

2.2.3 Array Comparative Genomic Hybridization

There are also a number of studies [8, 22–24] using **Array Comparative Genomic Hybridization** (aCGH) data as the source of analyzing clonal heterogeneity. aCGH is an improved version of CGH which is developed to detect copy number variation. Since this thesis is not a biology thesis, we would not cover the mechanism of this technique, instead, we only introduce what the data look like.

aCGH is an array in which each element is a signal representing the copy number of a segment of the DNA sequence. Current aCGH techniques allows copy number variation at a level of 5-10 kilo bases (kb) to be detected [25]. Figure 2.9 is an example of aCGH data plot. The value of each element in the array is plotted in the chart.

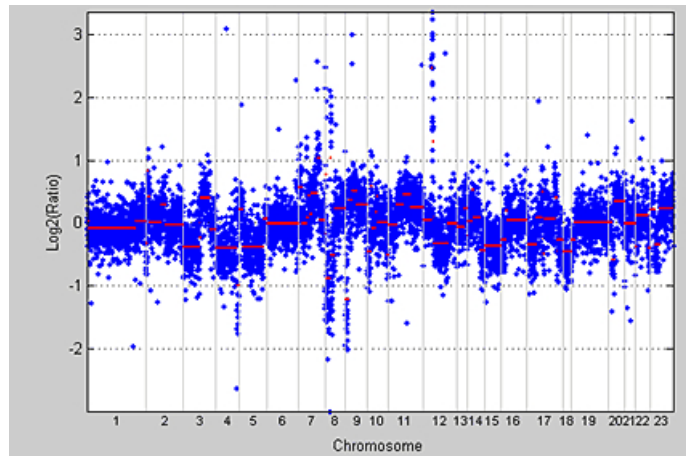


FIGURE 2.9: Example of aCGH data. X-axis represents the positions on DNA sequence and Y-axis is the \log_2 of the signal values, where 0 is normal because the normal copy number is 2. [26] The red lines indicate the segmentations of the data.

Since the resolution of aCGH is 5-10 kb and the total human genome has 3.2 billion bases, there are so many data points in the array that the data can be seen as continuous. Therefore studies using aCGH data mainly focus on how the continuous data can be segmented and how to assign a value for each segment. Once the data are segmented, such as the red line segments in Figure 2.9, segments with the same values belong to one clone.

2.3 Graphical Models

In above sections we have clarified the problem of clonal heterogeneity on the biological side, but we have not linked it to computer science. To summarize the problem in one sentence, we need to identify the clonal heterogeneity in cancer (hidden facts),

based on some observations (sequencing data), believing that the observations give some information about the clonal heterogeneity that exist in tumor. More concisely, in our project, we infer what mutations there are and how we can identify clones, given the data from NGS machine. In other words, we can say there are two sets of variables, one set for the observations and one for the representation of heterogeneity. We analyze the heterogeneity based on the **observations** and the **relationship** between observations and the clonal heterogeneity. Hence, using statistical models is a popular way for solving this problem, because it formalizes the relationship between random variables. In the problem of identifying intra-tumor heterogeneity, a statistical model can be used to quantify the relationship between the variables that represent the composition of the tumor and the variables representing the observations (sequencing data).

2.3.1 General Graphical Models

Graphical model is a framework to represent a statistical model where each node represents a random variable. An edge between nodes shows the relationship between the variables, which is usually quantified by conditional probabilities (in directed graphs). Linking to the problem of clonal heterogeneity, for example, if we know the sequencing data or the aCGH signal for a position as observations, we can infer the composition of the tumor, thereby identifying the heterogeneity.

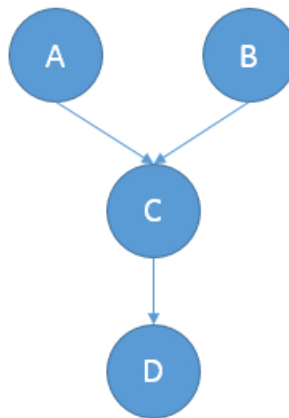


FIGURE 2.10: An example of graphical models. Variable C is dependent on both A and B, variable D is dependent on C.

A graphical can be either directed or undirected. The undirected graph model is also called “Markov Random Fields”, and we only focus on directed graph models because all the models related to this thesis are directed graph models. Directed graphical model is also called “Bayesian Network” and Figure 2.10 gives a simple illustration of directed graphical models. In the figure there are four random variables representing four

different events. The relationship between these events are represented by the links (or arcs) between the nodes. Conditional probabilities are used to quantify this relationship. For example, the arc $A \rightarrow C$ represents $P(C|A)$, which means the probability of getting any value of C given the value of A .

Overall, using graphical models we aim to infer the most likely values of all the random variables. In “Maximum Likelihood”, we determine the values which give the highest joint probability (likelihood function) of the model to be the most likely ones. The joint probability is the product of all the conditional probabilities in the model. For example in Figure 2.10, the likelihood function of the model is calculated as $P(D|C) \cdot P(C|A, B) \cdot P(A) \cdot P(B)$.

Below we describe two specific graphical models that are related to our project in detail.

2.3.2 Hidden Markov Model

A Hidden Markov Model (HMM) is a graphical model where there is a chain of hidden random variables, which are the same random variables whose value may change over time. They are called hidden because their values cannot be directly observed. However, there are some other variables whose values can be observed, and these variables are called “observations” or “evidence” which gives some information about the hidden variables. If we know the values of the observations, we can make inference on the values of hidden variables. HMM is useful when we model a problem where one random variable appears again and again but the values change over time or locations, and the previous variable gives some information about the value of the variable at next position. The chain that links all the hidden variables quantifies this relationship, which is also called “transition probability”. Figure 2.11 gives an example of HMM. Each x_t is dependent on the previous variable x_{t-1} and thus all the horizontal edges form the chain. To infer these hidden variables, we model the relationship between the hidden variables \mathbf{x} and observations \mathbf{y} . More precisely, the hidden variable at each position gives some information about the value we can observe. This relationship is represented by the vertical edges, and it is quantified by the “observation probability”, $p(y_t|x_t)$.

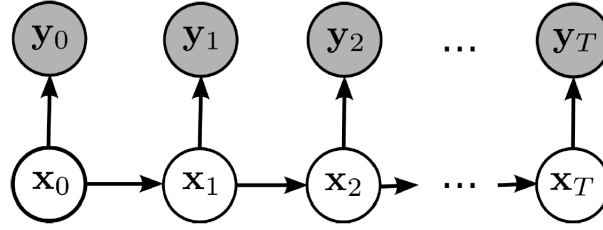


FIGURE 2.11: An example of HMM. $x_0 \dots x_T$ represents the hidden variable chain. x_n is dependent on x_{n-1} and this condition probability is also called “transition probability”. $y_0 \dots y_T$ are the observations, and the conditional probability representing this is called “observation probability”. Each observation is dependent on one hidden variable.

To give a simple example of the application of HMM, suppose we want to infer whether it rains or not each day based on the observations as to whether a person takes an umbrella for that day. Figure 2.12 represents the problem as an HMM. If it rains on day t , the probability of rain on day $t + 1$ is 0.7. Also, if it rains on day t , the probability that the person takes the umbrella is 0.9. Based on these conditional probabilities and the observed values for *Umbrella*, we can make inference on whether it rains or not each day, $\argmax_{x_{0:T}} P(x_{0:T} | y_{0:T})$.

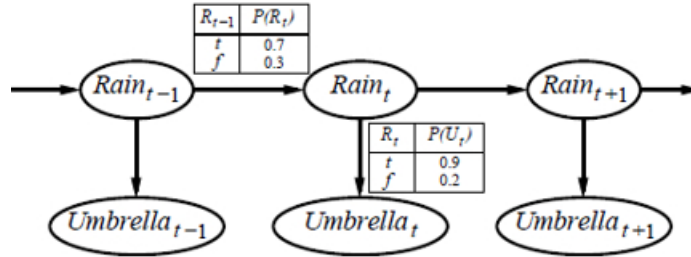


FIGURE 2.12: Rain example of an HMM.

2.3.3 Factorial Hidden Markov Model

Factorial Hidden Markov Model (FHMM) is a generalization of conventional HMM developed by Jordan and Ghahramani [27]. Instead of having only one chain of hidden variables, there are multiple chains of hidden variables. All hidden variables at one position share the same observation. However, there is no direct dependency between hidden variables from different chains. In other words, all the chains in the model are independent of each other. Figure 2.13 shows the graphical structure of an FHMM, with three chains of hidden variables $S^{(1)}, S^{(2)}, S^{(3)}$, and one observation variable Y_t for all the three hidden variables at the same position. Vertical arcs $S'_t \rightarrow Y_t$ represent conditional probabilities which describe the dependencies between hidden variables and observations.

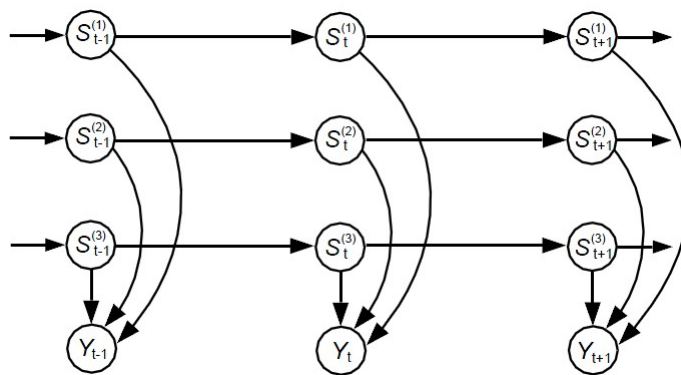


FIGURE 2.13: The structure of Factorial Hidden Markov Model

Note that the hidden variables at the same positions on all chains can actually be combined to form one hidden variable. Therefore, FHMM can be converted into a conventional HMM. The reason we split the repeated hidden variables ($S^{(1)}, S^{(2)}, S^{(3)}$) is that it may make the inference task more tractable. Furthermore, it may naturally make more sense to have multiple chains when modeling a given problem.

2.4 Statistical Models for Studying Heterogeneity

In section 2.3 we have discussed what graphical models are and how they can be used. We have also introduced the well-known Hidden Markov Model, which has been widely applied in many research studies in bioinformatics. In this section, we provide a current status of research that focuses on using statistical models to analyze clonal heterogeneity. The whole section is divided into two parts, models based on HMM and non-HMM based models. The difference will be discussed below in detail.

As mentioned in Section 2.3, the main idea for analyzing intra-tumor heterogeneity using computational techniques is to use statistical models. In statistical models, there is a set of variables representing the input data (observation variables), and another set of variables representing the real composition of the tumor (hidden variables, because we cannot directly observe their values). The following models all focus on faithfully quantifying the relationship between hidden variables and observation variables, and the relationship between hidden variables. Having faithfully quantified these dependencies, analyzing the clonal heterogeneity amounts inferring the values of hidden variables. These models differ in assuming what relationships actually exist and by which probability distributions they are quantified. The pros and cons of each model will be discussed below.

Although different models have different notations for the variables, most of them actually use similar variables. A list of symbols can be found at the beginning of this thesis and same variables with different notations are grouped together. We do not introduce a universal notation for all them for the convenience of referring back to their original papers.

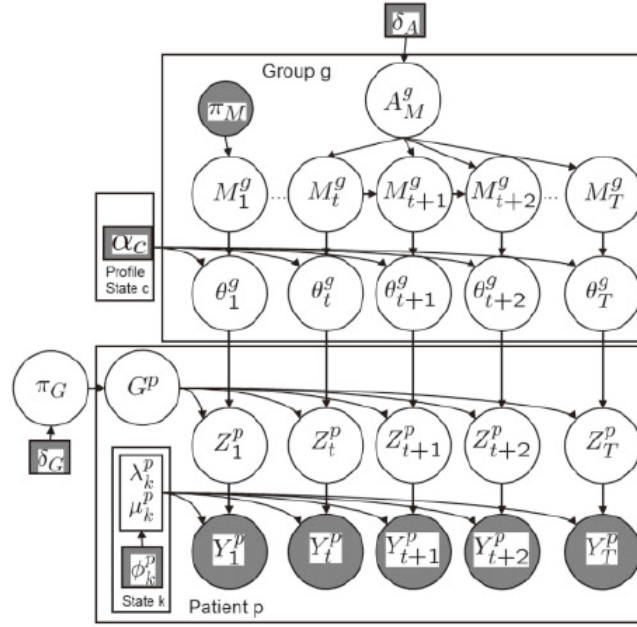
2.4.1 HMM based Models

We firstly review the models based on HMM. The key characteristic of HMM based models that differs from non-HMM based models is that in HMM, the hidden states are dependent on other hidden states, whereas in non-HMM based models, independence between hidden states is assumed. In studies of clonal heterogeneity, the hidden variables usually represent the composition of nucleotides on the DNA sequence of a single base or a segment of bases. Using HMM, we assume that the composition of nucleotides on one position gives some information of the composition on the neighboring nucleotides. These positions may not be actually consecutive on the DNA strand if only some parts of the whole genome are selected to be analyzed.

Three HMM based models will be discussed in this section. HMM-Mix [23] takes aCGH data as input and does the clustering task, i.e. identifying the clones that exist in the tumor. HMMCNA [22] also uses aCGH data as input but in a way of finding clusters that is optimized for constructing the progress history of the whole tumor. PennCNV [28] uses next generation sequencing data as input, and clusters the mutations into different clones.

2.4.1.1 HMM-Mix: A Model for clustering aCGH data

HMM-Mix [23] is a mixture of HMM models where there are multiple HMMs, each representing a clone. This model is also capable of taking multi-sample data, where each patient provides one sample. In the model, the log ratios of raw CGH data is represented by Y_t^p for probe $t \in (1, \dots, T)$ patient $p \in (1, \dots, P)$. Thus the input data is represented by a matrix $Y_{1:T}^{1:P}$, which is also the observation of the HMM. Hidden variables $Z \in \{L, N, G\}$ reflects copy number loss, normal and gain.

FIGURE 2.14: The graphical model of HMM-Mix²[23]

To summarize, their target is to based on observed data $Y_{1:T}^{1:P}$, infer what group G^p the patient p belongs to, and a posterior distribution of M^g s in each group g for a given profile. The graphical model is shown in Figure 2.14. The shaded variables Y s are observed and treated as evidence. In addition to Y s, all shaded nodes are observed. δ, π, α and ϕ are fixed parameters for the priors or distribution parameters. A_M^g is the transition matrix, whose values are learned by fitting the data. M_t^g is the hidden state on the chain. $M_t^g \in \{G, B, L\}$ representing gain, background and loss. This variable is very similar to Z but separated from Z because in this way, hidden states of M can be shared among different patients and each patient has their own Z s to represent their probe signals. θ^g is a parameter which is dependent on M^g and a fixed parameter α_c to determine Z^p . Finally $G^p \in \{1 \dots G\}$ represents the group number that a patient belongs to. On the inference side, the iterative conditional modes (ICM) algorithm is used. ICM can be seen as a deterministic version of Gibbs sampling. ICM always choose the most probable value for a random variable based on its neighbors in the graphical model.

Before discussing the pros and cons of HMM-Mix, there is one major problem with the model – it is clustering patients rather than clustering mutations or CNVs, although in [22] it is seen as dealing with intra-tumor heterogeneity issues. Finding subgroups among a patient cohort can be useful, but it does not really contribute to the problem of intra-tumor clonal heterogeneity, which deals with the subpopulations within one patient.

²Not all variables are explained in this thesis. For more details please refer to the original paper.

In terms of the model itself, it has its own strengths and limitations. As an HMM-based model, it considers the possibility that the state of one position may affect the next position. In addition, the transition probability A^g is learned by fitting the training data instead of making it a fixed value, such as in [22]. A^g is calculated as $p(A^g|M_{1:T}^g, \delta_A)$ where δ_A is the fixed Dirichlet hyper-parameter. The other advantage of this model is the inference algorithm. Both MCMC and EM algorithms were considered but they both turned out to be too slow. The ICM algorithm adopted in the study gives satisfactory results and performs well in terms of time taken. On the other hand, one important limitation of this study is that the state space is limited to gain, neutral and loss. The copy number can be amplified to 3,4 or even 5 copies, but in this model they will all be classified as “gain”, which loses information of the original data.

2.4.1.2 HMMCNA: Copy Number Analysis based on aCGH Data

Subramanian et al. developed a model that is based on HMM to identify subpopulations [22]. They use aCGH data which is introduced in Section 2.2 as input, and then use the model named Hidden Markov Model Copy Number Analysis (HMMCNA) to group the whole DNA sequence into different segments according to copy number. Then these different segments can then be considered as clones or subpopulations. For example in Figure 2.9, the red solid line segments represent the segmentation and the segments with the same log ratio value are grouped into one clone.

First, the probes in copy number profile (aCGH data) are combined into groups so that one single state variable in the model can represent a set of probes (e.g. 500 probes). Then Figure 2.15 illustrates what this study is trying to do, i.e. for each set, we want to determine its “true” copy number value, since the data is noisy. In HMM, each state represents if a set of probes is categorized as amplified(1) or normal(0). In the example of Figure 2.15 there are two samples under analysis, and 00 means both of the samples are normal in one set of probes. The state of one position gives some information about the next state as well as the observation, which is the actual data from aCGH. Once we achieve these segments, we can divide the whole DNA sequence into different clones. The inference task is that given the noisy data as evidence, we need to compute the value (0 or 1) for each state that represents a set of probes. The observation matrix is believed to follow an additive Gaussian noisy $X_{ij} = S_{ij} + \mathcal{N}(0, \sigma^2)$, where X_{ij} is the hidden state and S_{ij} is the observed signal. In this study, the inference is done using Viterbi algorithm [29].

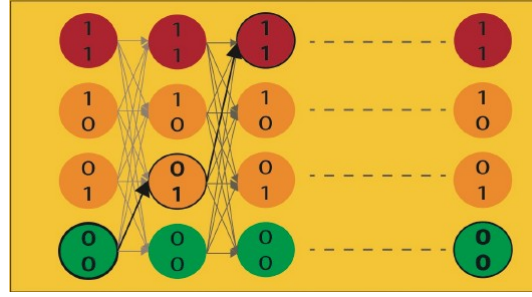


FIGURE 2.15: Hidden Markov Model Copy Number Analysis [22]

The strongest advantage of this model in addition to the previous HMM-Mix is that HMMCNA tries to find the clones that are optimized for building the phylogeny of cancer. Cancer phylogeny refers to the history of cancer and how different clones emerged from one single clone. This problem can be seen as the next step of analyzing clonal heterogeneity because once we understand how cancer develops, we can develop effective therapies. Many studies also combine the two problems, such as [8, 10, 15]. HMMCNA takes “multiple samples” as input whereas many other models can only take a single sample as observation. The number of samples is crucial for accuracy in inference task.

However, one important disadvantage of this model is that Subramanian et al. treated the copy number amplification and deletion in the same way, i.e. they only had normal/amplified state. The accuracy of the result may be doubtful since the deletion and amplification may disguise each other’s effect. Also, unlike the previous model, the transition probabilities are fixed and therefore not learned from the data. The key probabilities are p_{NA} and p_{AA} , representing the probability of going from normal to aberrant state and staying aberrant, and other transition probabilities are just uniform. Prior assumptions are made on p_{NA} and p_{AA} , which is open to doubt. The details of these probabilities can be found in their original paper.

2.4.1.3 PennCNV: Copy Number Analysis based on B Allele Frequency and log R Ratio

PennCNV is a statistical model based on HMM developed by Kai Wang et al.[28]. Instead of using aCGH data, it uses DNA high intensity sequencing data from Illumina platform. The output of the model is very similar to the previous HMMCNA model [22]. It finds the most likely state for each hidden variable, which represents the CNV genotype of each position. **Genotype** here refers to the composition of the bases on the DNA sequence. In this study, they adopted a six-state definition for the hidden states, which is shown in Table 2.1.

| Copy no. state | Total copy no. | Description (for autosome) | CNV genotypes |
|----------------|----------------|----------------------------|------------------------------|
| 1 | 0 | Deletion of two copies | Null |
| 2 | 1 | Deletion of one copy | A, B |
| 3 | 2 | Normal state | AA, AB, BB |
| 4 | 2 | Copy-neutral with LOH | AA, BB |
| 5 | 3 | Single copy duplication | AAA, AAB, ABB, BBB |
| 6 | 4 | Double copy duplication | AAAA, AAAB, AABB, ABBB, BBBB |

TABLE 2.1: The six-state definition adopted in [28]. For each genotype, letter A represents that the data from sequencing matches the reference genome and B means mismatch. The relationship between match and mismatch in term of copy number is coded as the genotype representation.

B Allele Frequency (BAF) and log R Ratio (LRR) are the two main observations that are used as input in this model. Both of them have been introduced in Section 2.2. The hidden variables in the hidden chain are the genotypes for each base, and there are two random variables for observations since we have two types of observations, BAF and LRR. In this study, r_i, b_i, z_i are the notations for the LRR, BAF and copy number state (genotypes) at position i respectively. The transition probability is calculated as

$$P(Z_i = l | Z_{i-1} = j) = \begin{cases} 1 - \sum_{k=2}^6 p_{j,k-1}(1 - e^{-d_i/D}), & \text{if } l = j \\ p_{j,l-1}(1 - e^{-d_i/D}), & \text{if } l \neq j \end{cases}$$

where d_i denotes the distance between two positions and D is a constant. p is an unknown parameter and estimated using EM algorithm. The observation probability follows the Illumina BAF calculation procedure and models the “boundary truncation” event. More details can be found in [28]. The inference task in this study is also achieved by the classical Viterbi algorithm [29].

Compared to the previous two related work, this model does have many advantages. First, this model adopts a larger state space for the hidden variables. In HMM-Mix and HMMCNA, only “gain”, “neutral” and “loss” three cases are included in the state space, while PennCNV uses six states (copy number from 0 to 4, where there are 2 states for 2 copies). This allows the copy number CNV events to be modeled more precisely. Second, the parameter p in the transition probability calculation is learned from data, and not fixed beforehand. Third, the observation probability is not an arbitrary distribution but a calculation that considers the “boundary truncation” event, which is a normalized measure of BAF and makes the results from different samples more comparable.

However, there are still limitations for this model. Including HMM-Mix and HMMCNA, all three models share one important assumption, which is that the mutations from different clones do not overlap. Other limitations of this study are mainly related to the

data they use. Since they are using the SNP array³ and SNP is not equally distributed across the whole genome, the CNV may not be captured if they happen between two SNP positions that are far from each other.

2.4.2 Non-HMM based Models

In the last section we presented three statistical models that are based on HMM to analyze clonal heterogeneity using different kinds of input. The strength of HMM based model is apparent – it considers the relationship between neighboring positions on the DNA sequence. However there are still a number of studies focusing on models that assume independence between the positions. In this section we discuss three models that are not HMM-based. The first model to be discussed, THetA, does not involve graphical models but develops an algorithm based on maximum likelihood theory. PhyloSub is a hierarchical method, which infers the structure of whole phylogeny tree first and then infer the cellular prevalence for each clone. The last model discussed in this section is very similar to our proposed model but is not based on HMM.

2.4.2.1 THetA: Copy Number Analysis using Maximum Likelihood Mixture Decomposition Problem

The algorithm named Tumor Heterogeneity Analysis (THetA), designed by Oesper et al.[13], is based on Maximum Likelihood Mixture Decomposition Problem (MLMDP). The input data that is used as input in this study is high-throughput DNA sequencing data. More specifically, it uses the total read depth of each position as observation to infer different clones. The overview of how the algorithm is described in Figure 2.16. Firstly suppose we have 3 cells, 2 normal cells and 1 aberrant cell. In the aberrant cell, the copy number of middle segment is amplified to 3. Then after aligning all the reads to the reference genome, we can get a distribution of reads over segments. Then this distribution is used as the input to the MLMDP problem solving method. The last part shows the result returned from the algorithm. μ denotes the cellular prevalence of the corresponding clone and \mathbf{C} is the segment count matrix, where each row represents a segment and each column represent its copy number in each of the clones.

This algorithm was shown to outperform three other methods on simulated data, which are ASCAT, CNAnorm and ABSOLUTE. The result of the study was also used to compare with the result in [9], as both of the study use the same data. While Navin et al. went through a huge amount of manual analysis, the result from THetA almost

³SNP stands for Single Nucleotide Polymorphism. It is a subset of bases of the whole human genome and it only contains the positions that are believed to be more likely to have mutations.

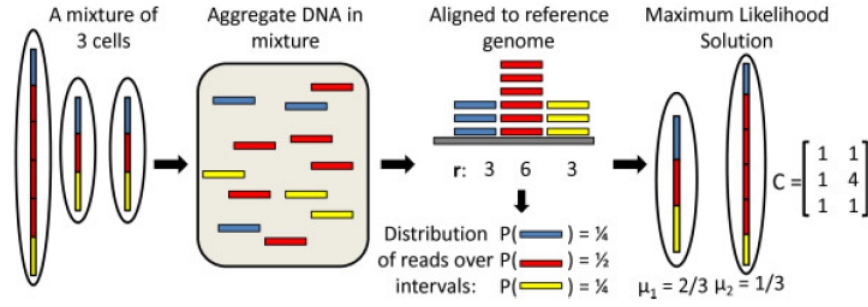


FIGURE 2.16: THetA algorithm overview.[13]

resembled the result from Navin et al.'s study automatically. Secondly, this method automatically reports the clones found from the sample, whereas all the methods we have been discussed above only segment the data. The output of cellular prevalence of each clone and the copy number matrix are more meaningful and easier for medical researchers to use.

The shortages of this method also cannot be ignored. Firstly, it excludes all the possible relationship between different positions. In other words, the genotype in one position does not tell us any information about which genotype is likely to appear in the next position. Secondly, THetA only considers read depth as observation, without taking the frequency of mutations into account. This means subpopulations without CNV would not be identified by THetA such as point mutation. Although CNV is ubiquitous in tumors, there are also point mutations.

2.4.2.2 PhyloSub: Hierarchical Bayesian Model Designed for Phylogeny Reconstruction

PhyloSub is a graphical model developed by Jiao et al. that is used to reconstruct the phylogeny of cancer [16]. As mentioned in section 2.4.1.2, phylogeny reconstruction is the next step of clonal heterogeneity analysis. This model infer the tree structure and cellular prevalence of each mutation using Gibbs sampling. Since the scope of this thesis only deals with clonal heterogeneity study, we omit the part of phylogeny construction but focus on clones identification.

The model uses read depth and reference count from DNA sequencing data as input, and generates a phylogeny tree with cellular prevalence of each node in the tree. Figure 2.17 gives an example of the phylogeny tree.

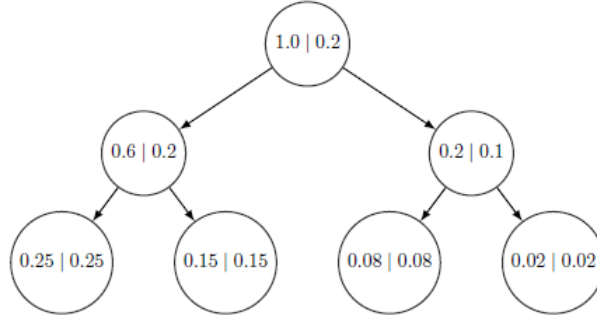


FIGURE 2.17: Phylogeny tree generated by PhyloSub model. The label of the nodes are the corresponding cellular prevalence and weights, where $\sum_{v \in V} \eta_v = 1$ and $\phi_v \geq \sum_{w \in \mathcal{C}(v)} \phi_w$ [16]

Figure 2.18 gives the details of the graphical model. All shaded variables represent observations, while others are hidden variables we need to infer. $i = 1 \dots N$ represents location 1 to N on the DNA sequence. G_i is the genotype in position i and evidence used include read depth d_i , reference count a_i , probabilities μ of sampling a given genotype. Dirichlet parameters $H, \alpha, \gamma, \delta$ are used to determine the structure of the tree. μ_i^r and μ_i^v are the probabilities of getting a reference allele from reference and variant population respectively. The important variable $\tilde{\phi}_i$ denotes the cellular prevalence from the variant population at position i . The notations and parameters are very similar to PennCNV model, discussed in section 2.4.1.3.

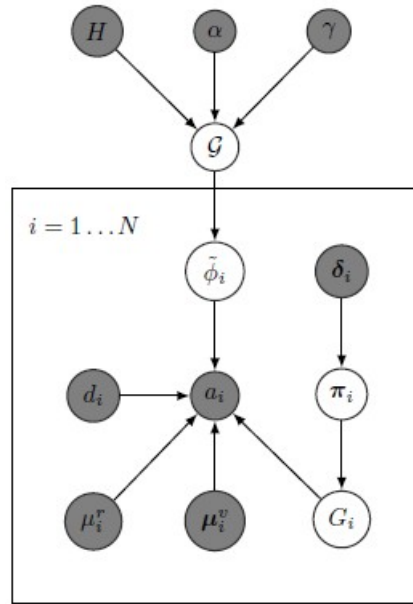


FIGURE 2.18: The graphical model of PhyloSub⁴[16].

⁴Not all variables are explained in this thesis. For more details please refer to the original paper.

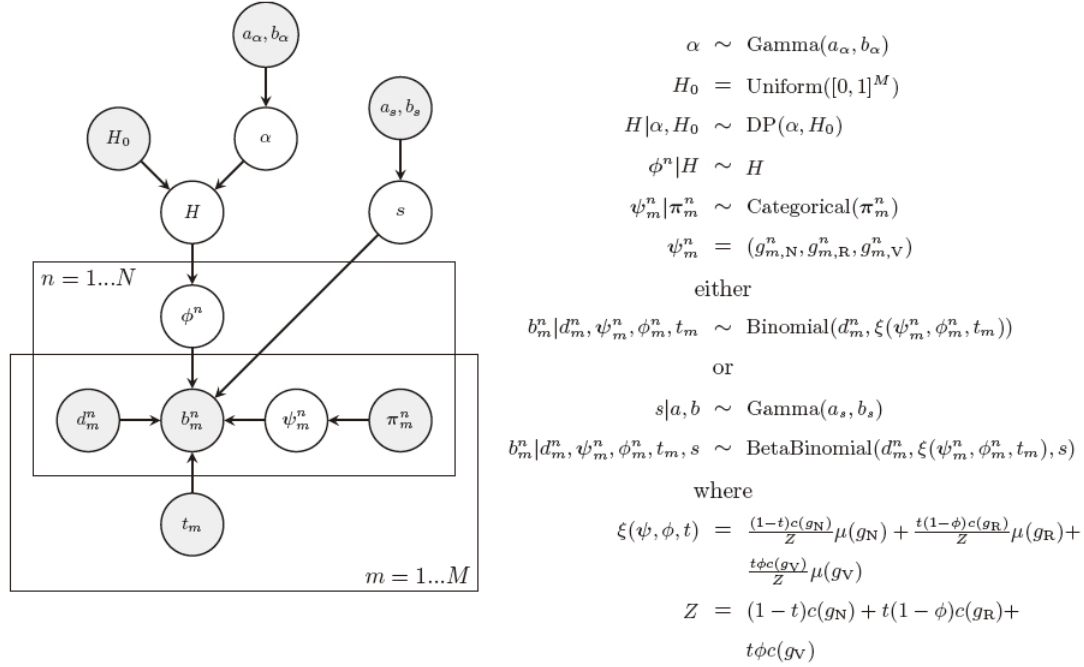


FIGURE 2.19: The graphical model of PyClone and related conditional probabilities⁵[31].

The advantage of this model is that it takes a good number of factors into account, so that the model is more faithful to the real world. However, as any other non-HMM based model, the relationship between neighboring positions are not modeled.

2.4.2.3 PyClone: Bayesian Model using Beta-Binomial Emission Densities

The last model that is to be discussed is the PyClone model developed by Roth et al. [31]. It uses deep DNA sequencing data as input. More specifically, they use deeply sequenced mutations whose coverage (total read depth) is larger than 100. The model clusters the mutations according to their cellular prevalence. The graphical model of PyClone is very similar to the graphical model of PhyloSub, but uses a different observation probability distribution and an another method for clustering. Figure 2.19 shows the model in detail and conditional probabilities are listed on the right, where $n = 1 \dots N$ represents N different clones and $m = 1 \dots M$ represents M samples from one tumor.

As in any other graphical models, shaded variables are observed. $H_0, a_\alpha, a_s, b_\alpha, b_s$ and π are priors that are used for non-parametric clustering, Gamma distribution and prior for genotypes. d denotes the total read depth and b represents the B allele count (non-matches against reference). t_m shows the purity of sample m , which is the proportion of tumor cells in the sample. The two important hidden variables that are to be inferred are ϕ^n and ψ_m^n , which denote cellular prevalence of a mutation in the sample and the

genotypes of a mutation, respectively. The inference task is done by MCMC sampling, which is a widely adopted method for inference in graphical models.

Compared to the two models discussed above, this model is more faithful in terms of analyzing clonal heterogeneity. Firstly, the beta-binomial distribution is used for the link between allele count and genotypes, which is more robust in modeling data that has a higher variance in allelic measurement. All other models discussed above, except for PennCNV [28], use binomial distribution for this probability, which is reasonable because for each read, it is either a match (A allele) or mismatch (B allele). Due to the noise of data, proportion of normal cells and the copy number of the mutation, the allelic prevalence is not directly related to cellular prevalence. In order to account for this issue, beta-binomial is shown to be more robust than the binomial model. Secondly, this model also follows a non-parametric clustering method, which does not limit the number of clones that can be found.

This model does have many limitations, but most of these limitations are shared with all the methods discussed above, which is what motivates our study. First, the model assumes mutations persist in the clone and each base can only have one kind of mutations. This assumption is made in all of the studies discussed in this section, because without this assumption, the inference task usually becomes intractable. This is also the main motivation for our study. Second, they cluster the mutations according to their cellular prevalence instead of actual compositions. Having the same cellular prevalence is a necessary condition for two mutations to be in one clone but not a sufficient condition.

2.4.3 Summary for Related Work

Having discussed all the models, a summary of strengths and limitations of each model is presented in Table 2.2.

⁵Not all variables are explained in this thesis. For more details please refer to the original paper.

| | Input | Advantages | Disadvantages | HMM-based? |
|----------|-------|--|--|------------|
| HMM-Mix | CGH | Transition probabilities learned from data. Advanced inference algorithm | Limited state space. Group patients instead of mutations | Yes |
| HMMCNA | aCGH | Optimized for phylogeny construction. Multi-sample. | Fixed transition probability. Limited state space. | Yes |
| PennCNV | NGS | Six-state space. Transition probabilities learned from data. Sophisticated observation probability | Only used SNP array as input | Yes |
| THetA | NGS | Automatically report clusters. Accurate performance. | Assume independence between positions. Only consider read depth. | No |
| PhyloSub | NGS | Model is used to build phylogeny tree. | Assume independence between positions. Only consider read depth. | No |
| PyClone | NGS | More robust observation probability. | Assume independence between positions. Only cluster mutations on cellular prevalence | No |

TABLE 2.2: Summary of related work

2.5 Summary

In this section, we provided the background of relevant knowledge in the area of studying clonal heterogeneity. We first explained the phenomenon of clonal heterogeneity in cancer and the practical problems associated with this phenomenon. Possible causes are also mentioned. Then we introduced different kinds of data that can be used as input when analyzing clonal heterogeneity. Knowledge of computational side with respect to graphical models were presented and discussed in Section 2.3, where we also introduced

two well-known models, HMM and FHMM. In the last section, we described six statistical models that were developed by other researchers, three of which are HMM-based and three are not. We believe the relationship between mutations must be captured, because the processes that cause mutations do not simply affect one nucleotide. Instead, it works on a segment of the genome. In addition, although the above models all have their own strengths and limitations, there is one major assumption made by all of the studies, which is that they assume mutations do not overlap on the DNA sequence. In other words, one specific mutation can only appear in one clone. However, there is no evidence for this assumption. Therefore, modeling the relationship between mutations and allowing mutations to overlap among clones are the two motivations to our proposed model.

Chapter 3

Studying Heterogeneity in Cancer using FHMMs

In the previous chapter, we have discussed several state-of-the-art computational models for analyzing clonal heterogeneity, and we have identified the two issues that motivate our study, which are the relationship between mutations and the overlap of mutations among clones. In this chapter, we focus on the details of how we construct our novel model, which is based on Factorial Hidden Markov Model (see Section 2.3.3). The model was originally postulated by my supervisor Dr. Gholamreza Haffari. The chapter begins with a description of the input and output of the model. The details of the model Het-FHMM are given in Section 3.2. The inference challenges and algorithms used are presented in Section 3.3, and the related time complexity is discussed in Section 3.4. In the last section of this chapter, we discuss the theoretical strengths and limitations of the proposed model.

3.1 Input and Output

Table 3.1 gives a sample of the input file to our model. As mentioned before, it is the alignment file of the next generation sequencing (NGS) data. Three observations are used, which include the reference count a , total read depth N and tumor-normal ratio l . The last column shows the position on the DNA sequence. The distance between locations is used for transition probability calculation. Figure 2.8 shows how a_t and N_t are calculated from the alignment file. l_t is computed by doing the sequencing twice, one for a tumor sample and one for a normal sample. Then l_t is calculated as $\log \frac{N_{tumor}}{N_{normal}}$.

| a | N | l | location |
|----------|----------|----------|----------|
| 235 | 439 | 1.7718 | 0 |
| 244 | 481 | 0.1 | 1996 |
| \vdots | \vdots | \vdots | \vdots |

TABLE 3.1: Sample input file

| iteration | -logLikelihood |
|-----------|----------------|
| 1 | 47200.1 |
| 2 | 47194.6 |
| 3 | 47187.2 |
| 4 | 47183 |
| \vdots | \vdots |

| clone 0 | clone 1 |
|----------|----------|
| AB | AABB |
| AB | AABB |
| AB | AABB |
| AB | AABB |
| \vdots | \vdots |

TABLE 3.2: Sample output file

There are two outputs from our model. One of them contains the likelihood of the whole model after each iteration of Gibbs sampling, which is an inference algorithm we use. The other output shows the genotype at each position returned from the inference algorithm, as well as the proportion of each clone (cellular prevalence). Details of the meaning of the genotype and cellular prevalence will be described in next section. Table 3.2 presents the format of the two outputs (cellular prevalence is not shown).

3.2 Our Model: Het-FHMM

The model is named **Het-FHMM**, which means “a model based on Factorial Hidden Markov Model (**FHMM**) to explore **H**eterogeneity”. We choose to base our model on FHMM for three reasons. In the last section in Chapter 2, I mentioned that the two key motivations to this project are that the relationship between mutations should be captured and other models assume mutations do not overlap among clones. Therefore first, the transition probabilities model and quantify the relationship between genomic positions. Second, having multiple chains in the model with each chain representing a clone, we do not need to assume mutations do not overlap among clones. These two things make our model more faithful than other models, hence having more representational power. In addition, existing powerful inference algorithms developed for FHMM is also a reason we choose FHMM.

In the general FHMM model, as described in Section 2.3.3, there are multiple hidden chains. Each hidden variable in one chain is dependent on the previous variable. The variables at the same position also share the same observation variables. In our FHMM based model which we call Het-FHMM, we have multiple chains of hidden variables for genotypes. The genotype of a position represents the composition of nucleotides at each position from all the cells in the sequencing sample, in terms of matching or mismatching the reference genome. In Het-FHMM, the value of a genotype at one position is dependent on the genotype at its previous position (quantified by the transition probability). Different chains in Het-FHMM represent different clones (or subpopulations) that exist in the tumor. In other words, each clone is modeled by one complete chain of genotype variables in the model. The link between genotype variables and the next generation sequencing data (used as the input) is quantified by observation probabilities.

Table 3.3 lists all the variables in the model and the graphical model of Het-FHMM is shown in Figure 3.1. The genotypes are denoted by $G_{t,k}$ where $t \in [0, T]$ is the position on the chain and $k \in [0, K]$ denotes which chain or clone the variable belongs to. T and K denote the length of the genome and the number of chains in the model respectively. The concept of a genotype was introduced in Section 2.4.1.2, but compared to PennCNV [28] which considered copy number up to 4, we consider copy number to 5, which gives a state space of 21 possible values for the genotype variable. Thus the state space for all the genotype variables is $G \in \{NA, A, B, AA, AB, BB, AAA, \dots BBBBA, BBBBB\}$. Each “A” represents a copy that matches the reference genome and “B” represents mismatch (recall that there are two copies of DNA in normal cells). “NA”, not available, describes the case when there is no copy at all. The first chain in Het-FHMM ($k=0$) represents the normal cells, since even in the tumor sample there is also a portion of normal cells. As we only consider heterozygous positions¹ on the genome, all $G_{t,0}$ are set to genotype **AB**².

¹Heterozygous positions are those where one of two copies is different from the reference genome. We only focus on heterozygous positions because if there are two matching copies and one of them mutates, there is still one functional copy. But for heterozygous positions, although people may stay healthy with only one functional copy, this copy may get mutated so that the function is lost.

²This is equivalent to reduce the state space of $G_{t,0}$ to a single value AB.

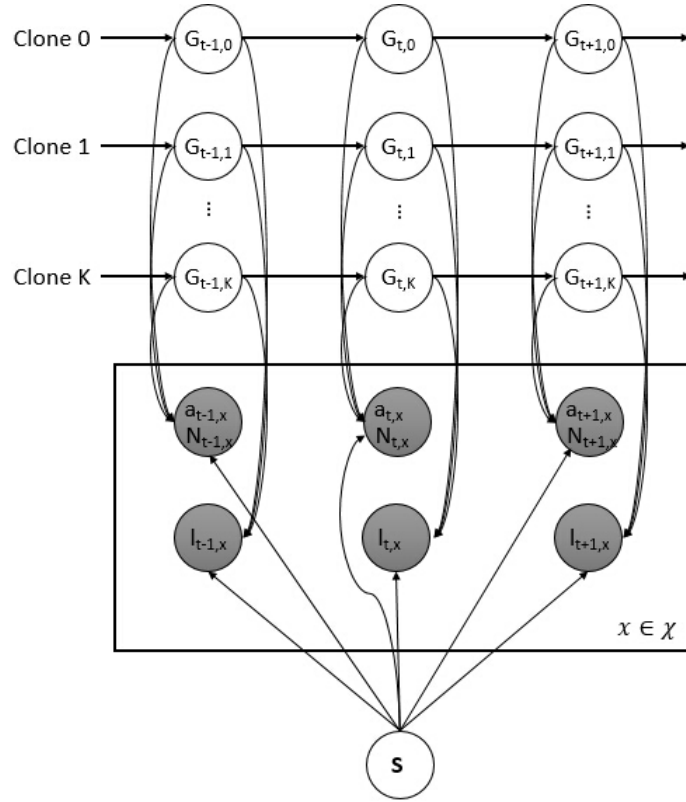


FIGURE 3.1: Graphical model of Het-FHMM

| Symbol | Meaning |
|--|--|
| $G_{t,k} \in \{NA, A, B, \dots, BBBBB\}$ | The genotype at position t in chain k |
| $a_{t,x}$ | The reference count from sample x at position t |
| $N_{t,x}$ | The total read depth from sample x at position t |
| $l_{t,x}$ | The log ratio of tumor-normal read depth from sample x at position t |
| \mathbf{S} | The vector that represents the cellular prevalence of each clone, i.e. the proportion of cells. $\sum_k S_k = 1$ |
| χ | The set of all samples of input |

TABLE 3.3: Summary of variables in Het-FHMM

For each chain k , there is a hidden variable S_k representing the cellular prevalence of that clone. In our Het-FHMM, they are represented with a single node, which is a vector $\mathbf{S} = (S_0, S_1, \dots, S_K)$. The elements of vector \mathbf{S} must sum to 1. For example, if $\mathbf{S} = (0.4, 0.3, 0.3)$, it means there are 3 chains (one clone for normal cells and two clones for tumor cells) in the model, representing 40%, 30%, 30% of all the cells respectively.

The final components of the model are the observations, which connect the genotype variables and the cellular prevalence vector \mathbf{S} . The three kinds of shaded variables a , N and l are the observations which are inputs to our model, whose meanings have been described in the previous section. In Figure 3.1 a and N are put together in one node because they are closely related. Since our model supports multiple samples as input, $a_{t,x}$ and $N_{t,x}$ denote the reference count and total read depth for position t from sample x , and $l_{t,x}$ denotes the log R ratio. χ represents the set of all samples for a given tissue. For each position t , there could be several samples given as input.

Overall, we aim to use this model to identify the composition of DNA in different cells in a tumor, thereby identifying clonal heterogeneity. The input of read depth N_t , reference count a_t and the log ratio l_t are used as observations on which we do the inference. The genotype at each position is dependent on the genotype of its previous position, because there is a relationship between neighboring positions. All the genotype variables at one position determine the observations, that is a_t, N_t and l_t are dependent on $G_{t,0:K}$.

| Copy Number | Genotype |
|-------------|---------------------------------|
| 0 | NA |
| 1 | A,B |
| 2 | AA,AB,BB |
| 3 | AAA,AAB,ABA,ABB,BAA,BAB,BBA,BBB |
| 4 | AAAA,AAAB,...,BBBB |
| 5 | AAAAA,AAAAB,...,BBBBB |

TABLE 3.4: Genotype variable space. We consider the case when the copy number is amplified up to 5. A means the nucleotide matches the reference and B means there is a mismatch.

The FHMM contains both transition probabilities and observation probabilities. Firstly, the transition probabilities are expressed in terms of matrices $A_t(i, j)$, where i and j range over possible genotypes; the space of genotypes for each site contains 21 possible states as shown in Table 3.4. The transition probability is defined as:

$$\underbrace{P(G_t = i | G_{t-1} = j)}_{A_t(i,j)} = \begin{cases} \rho_t & \text{if } i = j \\ \frac{1-\rho_t}{D-1} & \text{otherwise} \end{cases} \quad (3.1)$$

where

$$\rho_t = 1 - \frac{1}{2}[1 - e^{\frac{-d_t}{2L}}]$$

in which L is the average length of segment³ and D is the dimension of the state space, which is 21 in this case. d_t denotes the distance between position t and $t - 1$, recalling that these are heterozygous positions and thus not adjacent in the genome. We employ this transition probability from the study done by Colella et al. [33], because satisfactory results have already been achieved from it. The intuition behind the calculation of ρ_t is that the closer the two positions, the more likely they have the same genotype. When d_t is 0, ρ_t is 1, which is reasonable because for the same position there is only one genotype. As d_t increases, i.e. the distance between two heterozygous positions increases, ρ_t , the probability of getting the same genotype at the next position, decreases following the exponential function and it is always larger than $\frac{1}{2}$. The probability of getting one specific different genotype is the probability of not getting the same genotype, $1 - \rho_t$, divided by the number of all possible different genotypes, $D - 1$.

Secondly, the likelihood models for generating the observations based on the hidden variables are as follows:

$$P(a_t|N_t, \mathbf{G}_t, \mathbf{S}) = \text{Bin}(a_t|N_t, \mu_t(\mathbf{G}_t, \mathbf{S})) \quad (3.2)$$

$$P(l_t|\sigma, \mathbf{G}_t, \mathbf{S}, \phi) = \log(\mathcal{N}(l_t|m_t(\mathbf{G}_t, \mathbf{S}, \phi), \sigma)) \quad (3.3)$$

$$= -\log(\sigma) - \frac{1}{2} \cdot \log(2\pi) - \frac{1}{2} \cdot \frac{(l_t - m_t)^2}{\sigma^2} \quad (3.4)$$

$$\mu_t(\mathbf{G}_t, \mathbf{S}) = \frac{\sum_{k=0}^K S_k \cdot r_{g_t,k} \cdot c_{g_t,k}}{\sum_{k=0}^K S_k \cdot c_{g_t,k}} \quad (3.5)$$

$$m_t(\mathbf{G}_t, \mathbf{S}, \phi) = \frac{\sum_{k=0}^K S_k \cdot c_{g_t,k}}{S_0 \cdot c_{g_t,0} + \sum_{k=1}^K S_k \cdot \phi} \quad (3.6)$$

where $c_{g_t,k}$ is the copy number of a given genotype and $r_{g_t,k}$ is the allele ratio. The allele ratio is the ratio of matches over the copy number of a genotype. For example, if G_t is AAB, then c_{g_t} is 3 and r_{g_t} is 2/3. ϕ is a tumor ploidy parameter fixed to 3 in our model and σ is fixed to 2 for the standard deviation of the normal distribution. These two parameters are chosen arbitrarily, because we do experiments on synthetic data, they will not affect the inference results.

The intuition behind the Binomial distribution and Gaussian distribution are as follows. For the reference count a_t , it is easy to see that it is either a match or a mismatch for any read. So if there are N_t number of reads in total, the average number of matches μ_t is calculated by formula 3.5. The formula is straightforward to understand, where the numerator is the number of As (match) in all the chains at one position, and the

³It was observed to be 2 Megabases (2×10^6 bases) in 104 breast tumors (rounded to the nearest Mb.) [32]

denominator is the total copy number. S_k is the proportion of each clone, which is to give different weights to different chains. The tumor-normal depth ratio is modeled by a Gaussian distribution where the mean m_t is calculated by formula 3.6. We use Gaussian distribution because the ratio is continuous. The length of human genome is about 3.2 billion and since the copy number variation reflects the change of a segment of the DNA, l_t is not supposed to change dramatically and abruptly at each position. Therefore the value can be seen as continuous. In formula 3.6, the numerator reflects the total read depth from the tumor sample and the denominator reflects the total read depth from the normal sample.

In order to infer the hidden variables \mathbf{G} and \mathbf{S} , we use Maximum Likelihood in which we pursue \mathbf{G} and \mathbf{S} which give the highest joint probability of the model. In other words, we infer variables \mathbf{G} and \mathbf{S} which maximize the likelihood function of the model:

$$P(\mathbf{G}, \mathbf{l}, \mathbf{a}, \mathbf{S} | \mathbf{N}, \phi) = \prod_{k=0}^K \prod_{t=0}^T P(G_{t,k} | G_{t-1,k}) \quad (3.7)$$

$$\begin{aligned} & \prod_{x \in \mathcal{X}} \prod_t P(a_{t,x} | N_{t,x}, \mu_{t,x}) P(l_{t,x} | m_{t,x}, \sigma) \\ &= \prod_{k=0}^K \prod_{t=0}^T A_t(G_{t,k}, G_{t-1,k}) \\ & \prod_{x \in \mathcal{X}} \prod_t \text{Bin}(a_{t,x} | N_{t,x}, \mu_{t,x}) \mathcal{N}(l_{t,x} | m_{t,x}, \sigma) \end{aligned} \quad (3.8)$$

Because this joint probability involves a large number of variables, its values tends to be very small. Therefore, in the inference we use negative logarithm of the likelihood for calculation. We also take the negative for convenience, because the value of the logarithm of a probability is always small. The target of all the inferences is to minimize the value of the negative logLikelihood function (equivalent to maximize the original likelihood function). Details of negative logLikelihood will be introduced in Section 4.2.

3.3 Inference

In this section, we discuss inference algorithms which find \mathbf{G} and \mathbf{S} so that the likelihood function is maximized. During inference, we need to (i) Infer the latent variables \mathbf{G} given fixed parameters \mathbf{S} , and (ii) Learn the parameters \mathbf{S} given fixed setting for latent variables \mathbf{G} . Overall, the strategy we design to do these two tasks is to fix the value of \mathbf{S} and infer the most probable \mathbf{G} , and then fix the values of \mathbf{G} and infer the best \mathbf{S} .

We use Gibbs Sampling to infer \mathbf{G} and Exponentiated Gradient Descent(EG) to infer \mathbf{S} . The process is as follows:

1. Fixing \mathbf{S} , use Gibbs sample to infer \mathbf{G} .
2. Fixing \mathbf{G} , use EG to infer \mathbf{S} .
3. Repeat step 1 and 2 till we choose to stop.

3.3.1 Exponentiated Gradient Descent

Having fixed all the values for genotype variables \mathbf{G} , the negative logLikelihood can be minimized using Exponentiated Gradient Descent, which is similar to normal Gradient Descent. Gradient Descent is an algorithm to find the minimum of a given objective function (target function) by gradually approaching the minimum along the direction of the gradient function.

Exponentiated Gradient (EG) [34] Algorithm is a variant of normal Gradient Descent. The difference is that the update for Gradient Descent is to subtract the gradient of a target function, where as in EG the update is done by multiplying the exponents of the negative gradient. One reason we use have to use EG instead of normal Gradient Descent is the simplex constraint of EG, which means all the elements of the vector \mathbf{S} sum to one and there is no negative element. So in EG we pursue:

$$\max_{\mathbf{S} \in \Delta} -\mathcal{L}(\mathbf{S})$$

that is $\sum_k S_k = 1$ and $S_k \geq 0$, where \mathcal{L} denotes the objective function. In addition, it is proved to perform better when the target is sparse. In other words, it allows us to identify clones even if it contains only a very small proportion of cells.

In our case, the objective function is the logLikelihood function of the model, i.e. $\mathcal{L}(\mathbf{S}) = \log P(\mathbf{G}, \mathbf{l}, \mathbf{a}, \mathbf{S} | \mathbf{N}, \phi)$. To solve the above maximization problem, the EG updates are as follows:

$$S_k^{new} = S_k \exp [-\eta \nabla_{S_k} \mathcal{L}(\mathbf{S})]$$

where η is the learning rate. After updating each component of the parameter vector \mathbf{S} , the values are normalized so that they sum to one. The above updates are repeated until convergence.

In our model, for the EG updates, we need the derivatives which are derived using the chain rule as follows:

$$\mathcal{L}(\mathbf{S}) = \sum_t \log \left[\binom{N_t}{a_t} \cdot \mu_t^{a_t} \cdot (1 - \mu_t)^{N_t - a_t} \cdot \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(l_t - m_t)^2}{2\sigma^2}} \right] \quad (3.9)$$

$$+ Const$$

$$= \sum_t \log \binom{N_t}{a_t} + a_t \log \mu_t + (N_t - a_t) \log(1 - \mu_t) + \log\left(\frac{1}{\sigma \sqrt{2\pi}}\right) - \frac{(l_t - m_t)^2}{2\sigma^2} + Const \quad (3.10)$$

$$\nabla \mathcal{L}(\mathbf{S}) = \sum_t \left[\left(\frac{a_t}{\mu_t} - \frac{N_t - a_t}{1 - \mu_t} \right) \cdot \nabla \mu_t + \frac{l_t - m_t}{\sigma^2} \cdot \nabla m_t \right] \quad (3.11)$$

And:

$$\frac{d\mu_t}{dS_k} = \frac{c_{t,k} \cdot (r_{t,k} - \mu_t)}{\sum_{i=0}^K S_i \cdot c_{t,i}} \quad (3.12)$$

$$\frac{dm_t}{dS_0} = \frac{c_{t,0} \cdot (1 - m_t)}{S_0 \cdot c_{t,0} + \sum_i S_i \cdot \phi} \quad (3.13)$$

$$\frac{dm_t}{dS_k} = \frac{c_{t,k} - \phi m_t}{S_0 \cdot c_{t,0} + \sum_i S_i \cdot \phi} \quad (3.14)$$

Substitute $\frac{d\mu_t}{dS_k}$ and $\frac{dm_t}{dS_k}$ back to $\nabla \mathcal{L}(\mathbf{S})$, we can get the gradient of the objective function with respect to variable \mathbf{S} .

Now that we have described how to maximize the log likelihood in terms of the clones cellular frequencies \mathbf{S} , next we present a way of updating the clones genotypes \mathbf{G} given fixed clone proportions \mathbf{S} .

3.3.2 Gibbs Sampling

Since the exact inference for FHMM is intractable [27], we use a sampling method to get an approximation. The complete algorithm is provided in the Appendix A. As mentioned in Section 2.4, Markov chain Monte Carlo (MCMC) sampling is widely adopted for this task and Gibbs sample in one simple sampling scheme among MCMC methods. In order to do Gibbs sampling, we need to start with an initial model with all genotype variables determined. Except for the normal chain, we randomly choose a genotype for each variable based on the uniform distribution. Then, each hidden variable is sampled given the current state of rest of the variables. In our case, the probability of each genotype

for a hidden variable $G_{t,k}$ is :

$$P(G_{t,k}) \propto P(G_{t,k}|G_{t-1,k})P(G_{t+1,k}|G_{t,k})P(l_t|m_t, \sigma)P(a_t|N_t, \mu_t) \quad (3.15)$$

$$\propto A_t(G_{t,k}, G_{t-1,k})A_{t+1}(G_{t+1,k}, G_{t,k})\text{Bin}(a_t|N_t, \mu_t)\mathcal{N}(l_t|m_t, \sigma) \quad (3.16)$$

In other words, we sample each current state given another four variables, the previous genotype G_{t-1} , the next genotype G_{t+1} and two observations l_t and a_t . These are the four neighboring variables. The Gibbs stops when the convergence criteria is met. In our case, we define the number of times that each variable is sampled to be the convergence criteria. So for example, if the criteria is 1000 iterations, it means each of the random genotype variables are sampled 1000 times.

3.4 Time Complexity

Since there is no fixed time complexity for EG part of the inference, we assume the time taken for EG is constant E . For Gibbs sampling, there are $T \cdot K$ states in the model, and each state has 21 possible values. Therefore assuming the convergence criteria of Gibbs Sampling is I , the total time complexity is $O(21TKI) = C \cdot O(TKI)$.

3.5 Strengths, Limitations and Discussion

The biggest strength of this Het-FHMM is that it is more faithful to reality than other current models in that it allows mutations to overlap among clones and models the relationship between mutations in the same clone. More specifically, having multiple chains in the model allows the mutations to exist in more than one clone. Suppose we have only one chain (conventional HMM). At one position, there is only one genotype variable, i.e. only one mutation can be reflected at one position. Thus when clustering the mutations, this one variable can only be clustered into one group, which indicates that the mutation can only belong to one clone. This issue can be resolved by having multiple chains with each chain representing a clone. The model has more representational power than other models since it is not restrained by the assumption of non-overlapping. Furthermore, it models the effect of one mutation on its neighboring positions, i.e the positions that are closer to one mutation are more likely to have mutations than positions that are far away. In addition to the faithfulness of the model, existing powerful inference techniques are also one of the strength of our model.

Allowing mutations to overlap among clones usually lead to an intractable inference problem, but our model uses FHMM to separate the mutations into different chains.

Using sampling methods, the problem of computational time is alleviated. However, sampling methods are only to make an approximation. The accuracy of the inference still has room for improvement.

In terms of the model itself, there is also a limitation, which relates to the relationship between clones. In Het-FHMM, there is no direct link at all between the variables from different chains, which means all mutations in one clone do not have any relationship between the mutations in other clones. However, this is usually not the case in the real world. According to the tumor progression model described in Section 2.1, mutations in clones that appear later inherit some mutations from the older clones. The reason this relationship is omitted in most of research in the area is that it makes the inference intractable.

Another limitation of our model is that we fix the number of clones. Hierarchical methods can dynamically determine how many clones there are, but at this stage we can only use traditional search algorithms to find the number of clones that exist in the tumor. When doing the search, the negative likelihood function is used as the heuristic function. In this project we have not addressed the problem of searching for the most likely number of clones; this is a topic for future research.

There are another two issues worth discussing. The first one is the applicability of HMM and FHMM in recovering the DNA compositions. Normally HMM is used for modeling a time series of data, when t actually means the time. The variable X_{t-1} always happens before the variable X_t . However, the positions in the DNA strand do not appear in time but only have the relationship of relative locations. The genotype variable at position $t - 1$ does not appear before the genotype variable t in time but appears next to t physically. Therefore, G_t is not only dependent on the previous base G_{t-1} , but also dependent on its next neighbor G_{t+1} . Thus, the direction of the chain is arbitrary, which means inference with the reverse direction gives the same output result as the normal direction. Also, the two cases shown in Figure 3.2 have the same inference results, because the transition probability we define only depends on the distance between the two locations (refer back to equation 3.1), thus is symmetric.

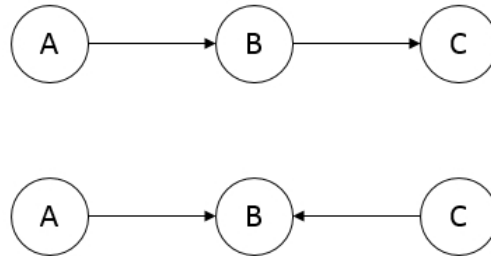


FIGURE 3.2: The two situations where the inference results are the same

Another issue is the order of the FHMM. The **order** of a HMM model means the number of previous variables which the current variable is dependent on. Het-FHMM is based on the first order FHMM, which means one genotype variable is dependent on one previous genotype. Figure 3.3 gives an example of second order HMM, in which each current random variable is dependent on two previous variables.

We believe there is a relationship between the distance between the locations and the genotype at each location, but there is no evidence showing the genotype is only dependent on the nearest neighbor. It could also be dependent on nearest two neighbors or dependent on the locations within a certain distance. Although the higher the order, the more faithful the model can be, inference is already a challenge for our task and increasing the order of the model will make the inference even more difficult.

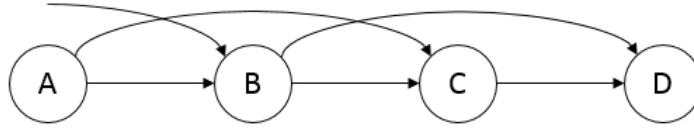


FIGURE 3.3: An example of second order HMM

Overall, our model has more representational power than other state-of-the-art models by allowing mutations to overlap between clones and modeling the relationship between the relationship between neighboring genotypes. The challenge is to find a good trade-off between the faithfulness of the model and the computational complexity. In other models, the above two factors make the inference intractable, whereas Het-FHMM models those two factors with a linear time complexity, although in experiments we only did experiments on shortened versions of the whole sequence.

Chapter 4

Experiments, Results & Discussion

In this chapter, we present several different experiments carried out for our novel model based on FHMM. First we describe the format of the input and output in detail. Then we describe the evaluation criteria we use, which includes the percentage of correct predictions, negative log-likelihood. We design a set of experiments to find the best configuration for the inference algorithm (Gibbs sampling). Then we analyze if the model is stable across different number of chains and different tissues. Lastly, the comparison between our model and a baseline model will be discussed.

4.1 Overview

The overall experimental design is shown in a flow chart in Figure 4.1. Firstly we generate synthetic data on which we carry out our experiments. Originally we plan to evaluate our model on the real data, but unfortunately we do not have a ground truth of real data, which makes the evaluation extremely difficult. The only way we can evaluate the model is to calculate the log-likelihood, but it is more for evaluating the inference algorithm but not the actual model. Therefore we decided to focus on the synthetic data. When generating the synthetic data, there are three parameters which we can change. First the data can be generated from different models. For the following experiments, except for the baseline comparison section, all the synthetic data is generated from Het-FHMM. Then another parameter included is the number of chains, which is equal the number of clones in the tumor. We can also adjust the proportion of each clone, which is the vector \mathbf{S} .

In addition to the three parameters mentioned above, the length of the whole genome can also be varied. The actual length of the human genome is about 3.2 billion bases, but in our experiment, we shortened it to 3×10^5 or 3×10^4 in order to run the experiments faster. Details on how the data is generated will be discussed in detail later.

After generating the synthetic data, we use it as input to run the inference algorithm based on our model. For inferring the proportion of clones, we use exponentiated gradient descent (EG) algorithm and we use Gibbs sampling for inferring the genotypes at each position. Since we are not doing a non-parametric clustering, we fix the number of clones. We also use a fixed number for the number of iterations as the convergence criteria of Gibbs sampling, so we need to run experiments on different number of iterations to find a reasonable convergence criteria. The last parameter we need to investigate is the number of times that we need to switch between Gibbs sampling and EG algorithm, trading off between accuracy and computational time.

Furthermore, as with any other machine learning algorithms, the more samples we include for observations, the more accurate the results should be. Therefore we also design experiments to observe the relationship between the performance and the number of samples we include. If we can show that the results do not vary too much between having only a small number of samples and having many samples, we can conclude that our model is robust even when the data is sparse. This is a particularly important aspect of this study, because usually researches only have one sequencing result for one patient. At last, results we get from running the inference are evaluated on different measures. Details about evaluation measures will also be discussed below.

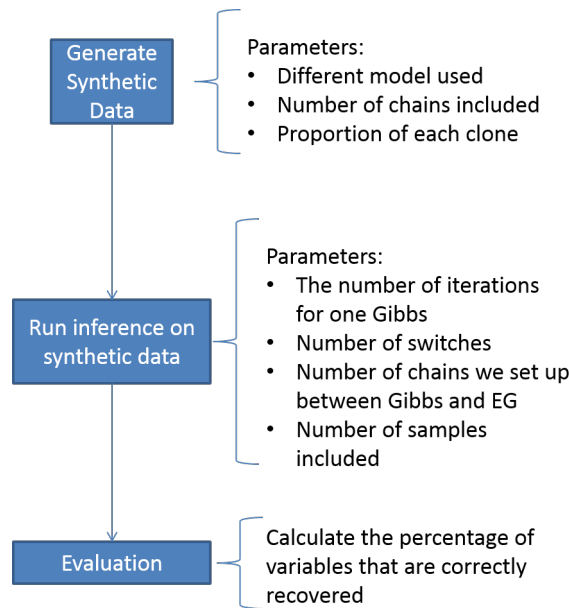


FIGURE 4.1: The flow chart for the experiment

The following experiments will be designed based on the four parameters for inference we mentioned above. In addition, we also did experiments on different tissues with the same property to show the stability of the model on different tissues from different patients.

4.2 Evaluation Mechanism

4.2.1 Percentage of Correctly Predicted Genotypes

To evaluate how well the inference performs on synthetic data, the most straightforward evaluation method is to calculate the how much the origin model/tissue which we use to generate the data is recovered. Therefore the first evaluation mechanism is to compute the percentage of correctly predicted genotypes. The accuracy is defined as

$$Accuracy = \frac{R}{T \cdot K} \quad (4.1)$$

where R is the number of genotype variables from the output that match the original model/tissue, and $T \cdot K$ is the total number of genotype variables in the model (K is the number of chains and T is the length of the genome). This evaluation mechanism is also referred as **Accuracy** in the rest of this thesis.

One issue is that when the model generates the output, the chains may be in any order. This means we need to compare each pair of the output chain and the original chain. This issue is resolved by considering all permutations of the output chains and comparing them with the original model. The permutation with the highest accuracy is selected.

4.2.2 Negative log-likelihood

The likelihood function (formula 3.8,3.9) is the probability of an assignment to all variables of the model. The inference algorithm maximizes the likelihood to find the most probable configuration. Since there are a huge number of variables in the model, the joint probability tends to be a very small number between 0 and 1. Therefore, we take the logarithm of the probability so that the value can be easier to evaluate. Because the log of a number between 0 and 1 is always negative, we take the negative log of the probability so that the value is always positive. The lower the negative log-likelihood is, the more probable an assignment to variables of the model is.

4.3 Experiments on Synthetic Data

4.3.1 Generation of Synthetic Data

We artificially generate the observed data, which include a_t (reference count), N_t (total read depth), l_t (log ratio of tumor/normal read depth) and the location of the genotypes, pretending the model/tissue genotypes are already known, i.e. the value of each random variable is known. The hidden variables of the first genomic location ($G_{0,1} \dots G_{0,k}$) are assumed to be the normal genotype AB. Having achieved the values for the first variables in each chain, we sample the values of other hidden states based on the transition probabilities, which considers the previous genotype and the distance between the two locations. Once the distribution of the variable at the next position is determined, we then sample the value according to this distribution. All the variables are generated by this mechanism. Note that all the genotypes in chain 0 are guaranteed to be AB, which is the normal genotype. The algorithm for generating synthetic data is shown below. Before running the data generation program, the proportion of each clone \mathbf{S} and the total number of clones that exist in the tumor must be specified as input.

Synthetic Data Generation Algorithm

Input: \mathbf{S} , number of chains

- 1: Assign $G_{0,0}, G_{0,1} \dots G_{0,K}$ to be AB
 - 2: Generate a_0 and l_0 based on the observation matrix
 - 3: $t = \text{next mutation position}$ //The next mutation is generated based on an average distance between mutations with random variance
 - 4: **while** $t \leq \text{genomeLength}$ **do**
 - 5: let $G_{t,0} = AB$
 - 6: **for** $i = 1 \dots K$ **do**
 - 7: sample $G_{t,i}$ from the transition probability based on $G_{t-1,i}$
 - 8: **end for**
 - 9: Generate a_t and l_t based on the observation matrix
 - 10: $t = \text{next mutation position}$
 - 11: **end while**
-

There are also some other fixed parameters involved in generating the synthetic data. First, the coverage of each base is set to between 400 and 500. Although the deeper the sequencing is, the more accurate the result should be, it has no impact on the inference process. The tumor ploidy parameter ϕ is set to 3 and the standard deviation for calculating l_t is set to 2.

The model we use to generate the synthetic data is also referred as the **original model/tissue** for the rest of experiments.

4.3.2 Finding the Best Configuration of the Inference Algorithm

In the previous section, we described how we generate a known model/tissue in a probabilistic and generate the synthetic data. Having obtained all the synthetic model/tissue data, we run the inference algorithm with different setups on the data to see if the original model/tissue can be recovered. We run Gibbs sampling for genotype variables and exponentiated gradient descent (EG) for the latent vector **S** representing the proportion of each clone (or chain). For Gibbs sampling, a fixed number of **iterations** of going through all the variables is considered as a parameter and we vary this value to find the minimum number of iterations that gives satisfactory results. In each iteration, each hidden variable is sampled exactly once.

The number of **switches** between Gibbs and EG is another parameter we aim to optimize. Similarly, we want to see if a large number of switches between Gibbs and EG is needed or a large number of iterations within Gibbs is more important. In EG, the learning rate is the parameter we need to adjust. If the learning rate is too large, the result may actually get increasingly worse, while if the rate is too small it may take too long to the converge. In our experiment, the learning rate is set to be adaptive, which means it gets smaller after each iteration in EG. More specifically, the learning rate η is set to $1/(\text{initial value} + w \cdot t)$, where t is the count of iterations which starts with 0 and w is the factor we need to adjust. Therefore we have two parameters here, one for the initial value and one for the change of the learning rate. These two parameters for EG are learned by manually running the program with different parameters, because if the learning rate is not appropriate, most of the elements in vector **S** will quickly become 0.

For the following experiments in this section, in order to make the results more reliable we do the experiments on one single sample case and one multi-sample (3 samples) case.

4.3.2.1 Experiment 1: Find the best configuration for the convergence criteria

The overall idea is that we run the inference algorithm with two different values as the convergence criteria and compare the plots of their negative log-likelihood function. If the negative log-likelihood does not improve too much from the smaller number of iterations to the large number of iterations, we would choose the smaller one because

we want the inference to finish as quickly as possible. From other studies such as [31], we know that for the sampling methods roughly at least 1000 iterations are necessary.

So for the first run we use 1000 as the number of iterations, and we carry out another experiment of 5000 iterations. Other parameters are set as following for both experiments: 20 times of switches between Gibbs sampling and EG, and 3 hidden chains in the model. The synthetic data used in these experiments has: 3 chains with the proportion of 40% normal cells and two tumor clones with cellular prevalence of 30% and 30%, with the genome length of 3×10^5 .

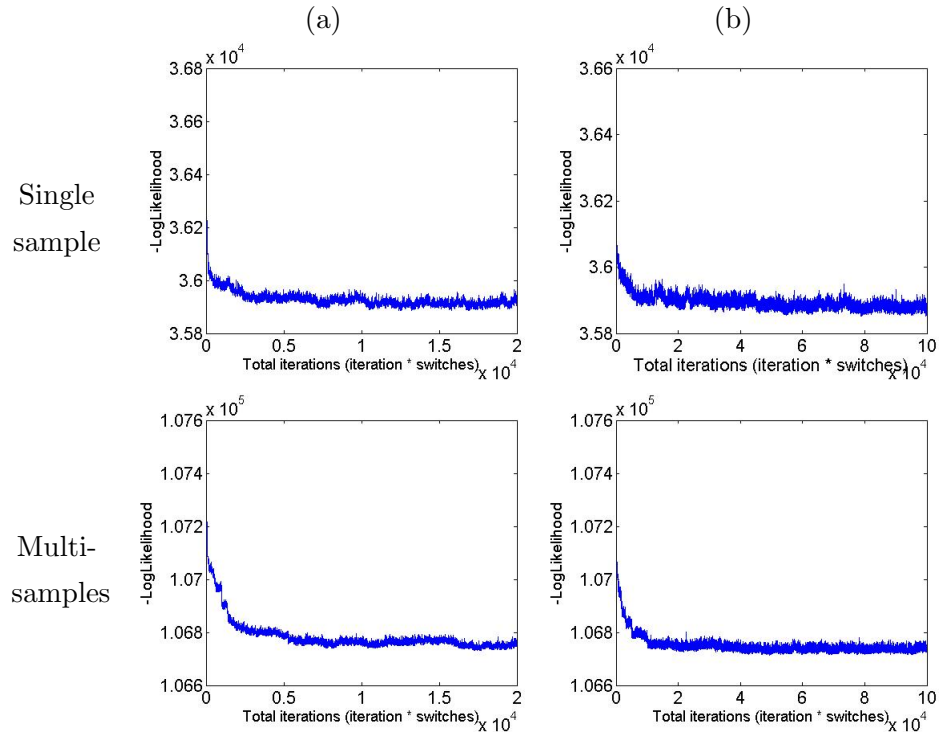


FIGURE 4.2: Negative log-likelihood of (a) running 1000 iterations per Gibbs \times 20 switches and (b) running 5000 iterations per Gibbs \times 20 switches. The X-axis represent the total number of iterations done, and Y-axis shows the -LogLikelihood value.

Figure 4.2 shows the plots of negative log-likelihood for using 1000 iterations (column (a)) and 5000 iterations (column (b)) as convergence criteria respectively. From Figure 4.2(b) of single sample (top right), we can see that the -LogLikelihood decreases mainly in the first 10000 iterations (5000 iterations \times 2 switches). After these 10000 iterations, there is no drastic improvement. This also indicates that the -LogLikelihood is improved mostly by the updates of Gibbs sampling compared to EG. The detail of this will be discussed in next section. Combined with the plots for multi-sample experiments (the bottom two plots), it confirms that the first 10000 iterations are the most important part of the inference. The effect of EG is also mostly obvious in the first 10000 iterations.

In Figure 4.2(b) of the multi-sample (bottom right), we can see two dramatic decrease that happen around iteration 5000 and 10000, where EG happens.

Apart from the -LogLikelihood comparison, we also check the accuracy of the model. Table 4.1 shows the results of all above four experiments averaged from 5 runs. Surprisingly, the accuracy of $20 \text{ switches} \times 1000 \text{ iterations}$ is even better than 5000 iterations on single sample. The Student's t-test shows that for single sample case, 20×1000 iterations is significantly more accurate than the 5000 iteration with the confidence of 80% (full results for each run can be found in Appendix B). For multi-sample case, the t test shows there is no significant difference. Although the standard deviation is lower for 5000 iterations, considering that the computational time taken is much longer for 5000 iterations, we conclude that 20 switches with 1000 iterations being the convergence criteria is more appropriate than 5000 iterations.

| Accuracy | Average (Single Sample) | Standard deviation | Average (Multi- sample) | Standard deviation |
|--------------------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|
| 20×1000 iterations | 0.46 | 0.083 | 0.43 | 0.044 |
| 20×5000 iterations | 0.39 | 0.048 | 0.44 | 0.027 |

TABLE 4.1: Percentage of random variables correctly inferred, averaging from 5 runs

4.3.2.2 Experiment 2: Find the best configuration for the number of switches between Gibbs sampling and EG

In Experiment 1 we found that the impact of Gibbs sampling is larger than EG. In this section we describe the experiment we carried out to further investigate this hypothesis. This experiment is very similar to Experiment 1, in which we compare the plots of -LogLikelihood and the accuracy.

The top two plots in Figure 4.3 show the plots of -LogLikelihood of two setups - $20 \text{ switches} \times 1000 \text{ iterations}$ (Figure 4.3(a)) and $100 \text{ switches} \times 200 \text{ iterations}$ (Figure 4.3(b)) for single sample case. The two plots are very similar in terms of both the trend and the final results. The shapes of the two plots are alike to each other and the -LogLikelihood decreases to around 3×10^4 for both of the two setups. This shows there is no obvious difference in inference outcome between the two setups.

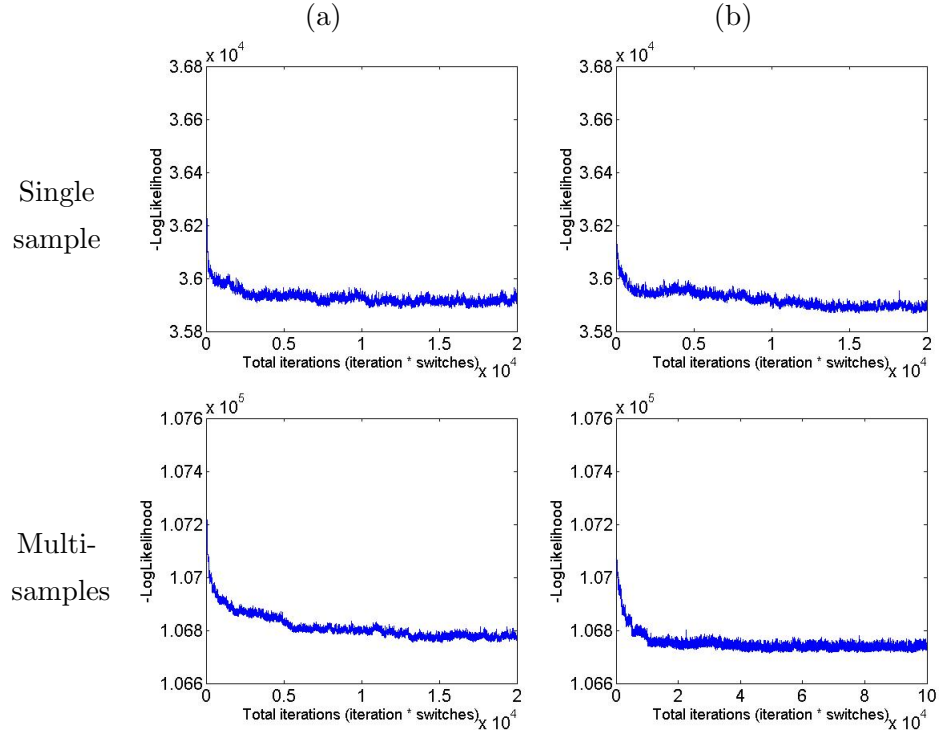


FIGURE 4.3: Negative log-likelihood of (a)running 1000 iterations per Gibbs \times 20 switches and (b)running 200 iterations per Gibbs \times 100 switches. The X-axis represent the total number of iterations done, and Y-axis shows the -LogLikelihood value.

The plots for multi-sample experiments (bottom two plots in Figure 4.3) also show that the improvement made on -LogLikelihood are the same for the two setups. They both decrease to around 1.0677×10^5 at the end.

Table 4.2 presents the accuracy of the two setups for running on the same data for 5 times. The average of accuracy and the standard deviation are included in the table. Looking at the average, 20×1000 iterations is better than 100×200 iterations because the average is higher. Student's t test further confirms the combination of 20×1000 is better with 60%(single sample) and 80%(multi-sample) confidence.

| Accuracy | Average (Single Sample) | Standard deviation | Average (Multi- sample) | Standard deviation |
|--------------------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|
| 20 \times 1000 iterations | 0.46 | 0.083 | 0.43 | 0.044 |
| 100 \times 200 iterations | 0.41 | 0.039 | 0.41 | 0.034 |

TABLE 4.2: Percentage of random variables correctly inferred, averaging from 5 runs

4.3.2.3 Section Summary

We have presented two experiments to find the best combination of number of iterations for Gibbs sampling and the number of switches between Gibbs sampling and EG. The analysis was done by comparing the plots of negative log likelihood of the model and the accuracy of recovering the original genotypes. Student's t test was also used to determine if there is a significant difference between the accuracy of different setups. The conclusion of the analysis is that the combination of 20 switches \times 1000 iterations being the convergence is significantly more accurate than the other two combinations, and from the plot we can also determine that the setup of 20 switches \times 1000 iterations is the optimal combination, considering that we want the inference as fast as possible¹. All following experiments use this setup for the inference.

4.3.3 Comparison between Different Number of Chains

In this experiment, we evaluate the performance of the inference when we have different number of clones. The experiment is designed as follows, and the percentages represent the cellular prevalence.

Four setups:

1. 2 chains: 40%(normal cells), 60%
2. 3 chains: 20%(normal cells), 30%, 50%
3. 3 chains: 40%(normal cells), 30%, 30%
4. 5 chains: 20%(normal cells), 10%, 20%, 30%, 20%

Genome length: 3×10^5 bases.

Note that the cellular prevalence for each setup is arbitrarily chosen, because we are not focusing on evaluating the cellular prevalence vector \mathbf{S} in this project. Therefore, we do not use an exhaustive set of cellular prevalence, but only make sure they are not uniform.

The experiment is run for 5 times for each setup. In order to make results more reliable, we do the experiments on one single sample case and one multi-sample case. Since we fix the number of chains before we start the inference, at this stage we always set the number of chains equal to the number of chains in the original model/tissue. Otherwise, the accuracy of the inference cannot be evaluated directly.

¹In this experiment, the time taken for 20 \times 1000 is about 1 day while it needs around 5 days to complete the 20 \times 5000 inference.

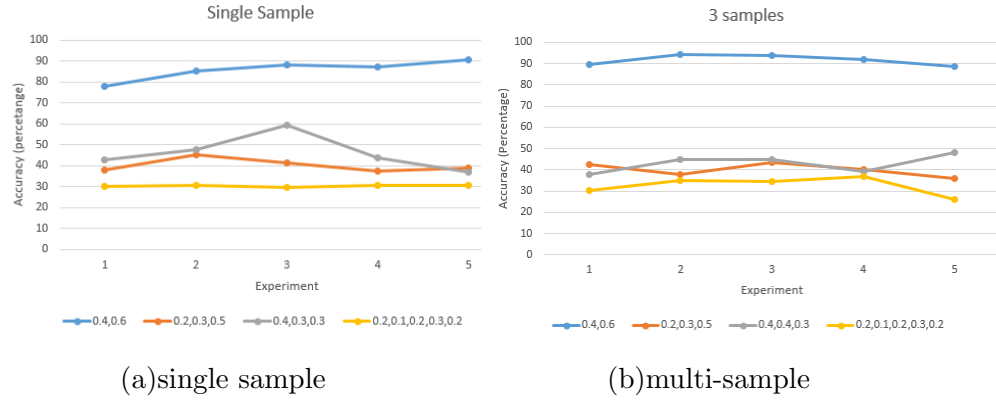


FIGURE 4.4: Line chart of running on different chains for five times. Each setup is represented by one color.

Figure 4.4 presents the line charts of the accuracy of the inference for all of the four setups for both single sample and multi-sample cases. The X-axis is the experiment number, and in total there are five experiments. Y-axis is the accuracy of the inference. Each line represents the accuracy of each setup. From these two line charts we can see the accuracy is very high (around 90%) when there are only two chains, but it drops dramatically when there are more than two chains. But with respect to each single line, there is no dramatic change, which means the inference results are stable and the standard deviation is low. Focusing on the two 3 chains experiments which have different **S** vector, we find that our model is stable when the cellular prevalence is different. The average accuracy for the two setups are 0.42 (20%,30%,50%) and 0.44 (40%,30%,30%). Table 4.3 gives the detailed statistics of the experiments.

| Accuracy | Average (Single sample) | Standard Deviation | Average (Multi- sample) | Standard Deviation |
|----------------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|
| 40%, 60% | 0.86 | 0.049 | 0.92 | 0.026 |
| 20%, 30%, 50% | 0.42 | 0.032 | 0.40 | 0.033 |
| 40%, 30%, 30% | 0.44 | 0.056 | 0.42 | 0.038 |
| 20%, 10%, 20%, 30%, 20% | 0.30 | 0.0046 | 0.32 | 0.040 |

TABLE 4.3: Result table of running the inference on different number of chains.

To summarize, we found that the accuracy for two chains situation is satisfactory, but when the number of chains increases, the accuracy deteriorates quickly. However, our model is robust on different cellular prevalence setups.

4.3.4 Inference Performance on Different Numbers of Samples Considered

In this section, we discuss the experiments for analyzing the relationship between the accuracy and the number of samples included. As stated above, if the difference of accuracy between accuracy of models with different number of samples is small, we can conclude that our model is robust even when the data is sparse. In order to be more inclusive, we carry out the experiments on each of the 2 chains, 3 chains and 5 chains setup. Also the results are taken from running the inference for five times to take the average, but since we are interested in the change when the number of samples included, all results below are already the average results. The genome length is shortened to 3×10^4 due to time limitation.

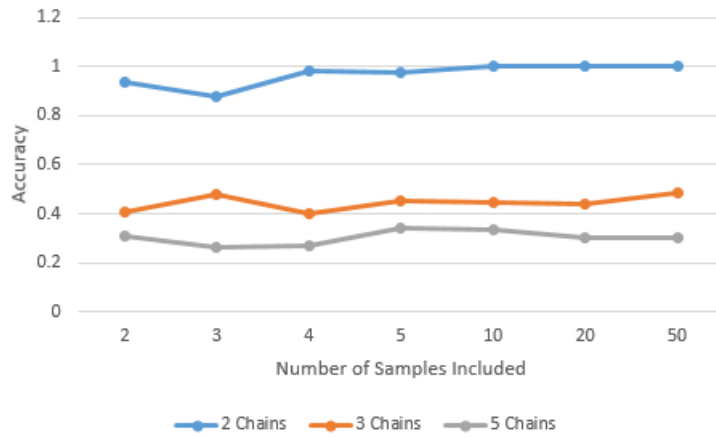


FIGURE 4.5: Line chart of running on number of samples included averaged from five times running. Each setup is represented by one color.

Figure 4.5 is the line chart for this experiment, which shows the change of accuracy when the number of samples included varies from 2 to 50. Each point on the lines is the accuracy result which is already on average from 5 runs. When there are two chains (blue line), we can see that as number of samples included increases, the accuracy also goes up, which is what we expected. However, for the experiment of 3 chains and 5 chains, the change of accuracy is not obvious. Therefore we look at the actual results to find the optimal number of samples to be included.

Table 4.4 is the data table for Figure 4.5, and the last row shows the average accuracy among all numbers of samples included for each of the 2, 3 and 5 chains. Then we compute the difference between the accuracy of each row and the last row. Table 4.5 shows the result of this calculation. In this table, the sum over 2, 3 and 5 chains of the difference for each number of samples included is calculated in the last column. Ranking over this sum from high to low, we can find the number of samples included which gives

the most accurate inference overall. As expected, 50 samples has the highest accuracy, with 0.08 higher than the three averages in total.

Again considering the computational time issue, we choose 5 samples included to be the optimal setup, trading off between computational complexity and accuracy. Another issue we cannot choose a large number is that usually in practical researchers would not have multiple sequencing on the same tissue in the same patient. The results show that our model and inference perform well even when the samples are sparse. All three lines are almost flat, and the standard deviation for the three chains are 0.05(2 chains), 0.03(3 chains) and 0.03(5 chains).

| Accuracy | 2 chains(40%,60%) | 3 chains(40%,30%,30%) | 5 chains(20%,10%,20%,30%,20%) |
|------------|-------------------|-----------------------|-------------------------------|
| 2 samples | 0.93 | 0.40 | 0.31 |
| 3 samples | 0.88 | 0.48 | 0.26 |
| 4 samples | 0.98 | 0.40 | 0.27 |
| 5 samples | 0.98 | 0.45 | 0.34 |
| 10 samples | 1.00 | 0.45 | 0.33 |
| 20 samples | 1.00 | 0.44 | 0.30 |
| 50 samples | 1.00 | 0.49 | 0.31 |
| Average | 0.97 | 0.44 | 0.30 |

TABLE 4.4: Result table of running the inference when different number of samples included

| | 2chains | 3chains | 5chains | sum |
|------------|---------|---------|---------|--------|
| 50 samples | 0.033 | 0.041 | 0.002 | 0.076 |
| 10 samples | 0.033 | 0.003 | 0.029 | 0.065 |
| 5 samples | 0.008 | 0.007 | 0.038 | 0.053 |
| 20 samples | 0.033 | -0.002 | 0.000 | 0.031 |
| 4 samples | 0.017 | -0.045 | -0.034 | -0.062 |
| 2 samples | -0.033 | -0.040 | 0.008 | -0.065 |
| 3 samples | -0.092 | 0.036 | -0.042 | -0.098 |

TABLE 4.5: The difference between each result in Table 4.4 and its corresponding average in the last row. The sum is calculated and ranked to find the most accurate setup.

4.3.5 Comparison between Different Instantiations of the Model

In this section, we show the results for analyzing the performance of the model on different instantiations of the model. Different instantiations of the model can be seen as equivalent to the different instantiations in reality. Therefore the aim of this experiment is to test the model to see whether the model would be stable across different kinds of instantiations when it is applied to real data. More specifically, we use the setup of 3 chains with proportion of 40% normal cells, 30% tumor cells of clone 1 and 30% tumor cells of clone 2. Five different instantiations are generated by running the synthetic data generation algorithm five times. Since 5 samples were found to be the optimal number of samples, we only use 5 samples as input for this experiment, and we run the inference five times to take the average. The genome length is also set to 3×10^4 for this experiment.

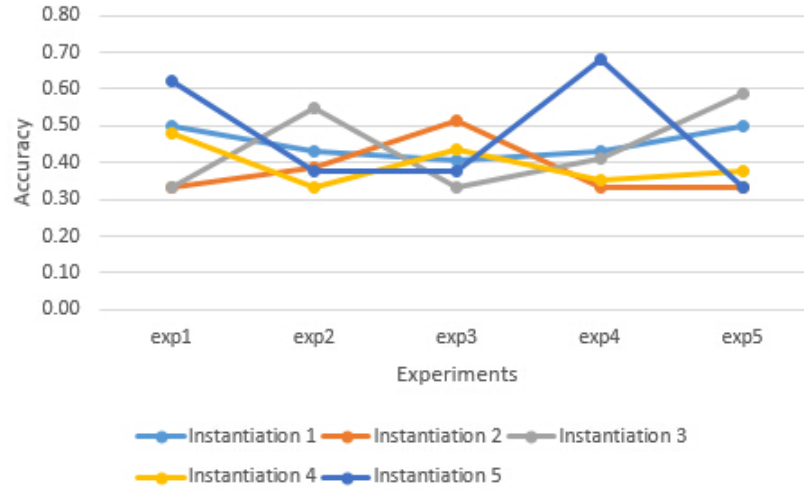


FIGURE 4.6: Line chart of running on different instantiations of the model for five times. Each instantiation is represented by one color. (5 samples used for input)

In Figure 4.6 above, each point represents the accuracy of one of five runs for one instantiation. Although results vary from experiment 1 to experiment 5, there is no significant difference among the results for different instantiations. This indicates that our model and inference is stable across different instantiations. To support this further, Table 4.6 shows the average accuracy of the five runs for each instantiation and the standard deviation is calculated in the last row. The standard deviation across different instantiations is even smaller than all of the standard deviations within each instantiation across the five runs.

| Accuracy | Average (5 samples) | Standard Deviation |
|--------------------|---------------------|--------------------|
| Instantiation 1 | 0.45 | 0.04 |
| Instantiation 2 | 0.37 | 0.08 |
| Instantiation 3 | 0.44 | 0.12 |
| Instantiation 4 | 0.39 | 0.06 |
| Instantiation 5 | 0.47 | 0.16 |
| Standard Deviation | 0.04 | |

TABLE 4.6: Result table of running the inference on different instantiations with the same setup (3 chains, 40%, 30%, 30%, 3×10^4 bases)

4.3.6 Summary

In Section 4.3, we have presented the experiments designed to evaluate our model and the inference algorithms on the synthetic data. Initially, we described how we had generated the synthetic data, and then four experiments that we carried out. Firstly, we determined the best combination for the inference parameters - number of iterations as the convergence criteria for Gibbs sampling. With two comparisons on the plots as well as the accuracy, 20 switches \times 1000 iterations is chosen. Secondly, we evaluated the accuracy of our model with respect to the number of chains. Result showed that our model had around 90% accuracy when there were two chains, but when the chain number increased to 3 and 5, the accuracy went down quickly to around 43% and 30% respectively. Moving on to the multi-sample inference, we concluded that 5 samples was the best choice considering the trade-off between computational time and accuracy. Lastly, we demonstrated that our model and inference algorithm were stable in terms of accuracy among different instantiations.

4.4 Comparison with the baseline

In this section, we compare the performance of Het-FHMM with another recent model that is published in Nature Methods. Originally we tried to use PyClone model[31] as a baseline comparison, but there are two issues. First, the input that our model uses is not exactly the same as what PyClone uses, according to the sample input file they provide on their websites. Second, since they assume mutations do not overlap between

different clones, there is no way to directly compare the two models on real data. Only when we have access to the ground truth of the real data, we can do the real baseline comparison between models.

Although we cannot compare PyClone and our model directly, we implemented another model to be an approximation to PyClone. PyClone and our model are very similar except that we assume the mutations at neighboring positions are not independent. Therefore, we developed a variant of our model such that all the transition probabilities between hidden states are removed. Then we compare our model with this variant to see which model performs better on the synthetic data.

In this experiment, we focus on three setups of 2(40%, 60%), 3(40%, 30%, 30%) and 5(20%, 10%, 20%, 30%, 20%) chains with the genome length of 3×10^4 . The experiments are repeated 5 times and we take the average. Since the best number of samples to be included is found to be 5 samples in previous experiments, we only make the model to consider 5 samples when doing inference.

However, one problem of comparing models based on synthetic data is that how these data are generated. As mentioned above, the synthetic data that have been used for experiments are generated from the our proposed model. More specifically, the probability of generating a genotype is dependent on the previous genotype (or referred as “dependent” model). Therefore, FHMM is expected outperform PyClone, because FHMM models the transition probability.

This becomes a bias when comparing models. In order to avoid this, we generate another set of synthetic data that is generated from a “independent model” where there is no transition probability. This set will favor the PyClone model in theory because PyClone assumes there is no relationship between positions. Then we run both of the models on both sets of data. More specifically, there are four combinations, Het-FHMM model inference on “dependent” model-generated data, PyClone approximation model inference on “dependent” model-generated data, Het-FHMM model inference on “independent” model-generated data and PyClone approximation model inference on “independent” model-generated data. If we can show our model is better on both sets of data, we can conclude that our model is better.

4.4.1 Result of Baseline Comparison

| | Chain Number | Het-FHMM | PyClone | t -test |
|------------------|--------------|----------|---------|-----------|
| dependent data | 2 | 0.92 | 0.78 | 0.159 |
| dependent data | 3 | 0.46 | 0.41 | 0.346 |
| dependent data | 5 | 0.25 | 0.26 | 0.679 |
| independent data | 2 | 0.83 | 0.76 | 0.033 |
| independent data | 3 | 0.42 | 0.40 | 0.298 |
| independent data | 5 | 0.30 | 0.30 | 0.784 |
| Average | | 0.45 | 0.42 | 0.046 |

TABLE 4.7: Comparison between FHMM based model and approximate PyClone model averaged from 5 runnings of all 3 setups. The last column shows the t -test value for checking the significant difference between two models for each setup. The average performance of FHMM model outperforms the PyClone approximation with confidence of **95.4%** under Student’s t test.

The result of the baseline comparison is summarized in Table 4.7. As mentioned in the previous section, there are 4 combinations between the data and models. The first three rows use the data generated from the “dependent” model, and the last three rows use the data generated from the “independent” model. The column “Het-FHMM” and “PyClone” mean the model that is used to do the inference. The values are averaged from 5 runs.

The first thing we can observe is that our model is more accurate than PyClone approximation for most of the cases. More precisely, we can see that the difference is significant when the number of chains is small. When there are two chains in the model, the confidence of saying there is a significant difference between the accuracy of the two models is around 97% for independent data, and 84% for dependent data. But as the number of chains increases, the difference becomes less significant. One reason may be because that both models are having low accuracy and therefore the difference is not apparent any more.

Looking at the overall accuracy, the average accuracy of our model for all 6 experiments is 0.45, versus 0.42 for the PyClone approximation. The Student’s t test shows that our model performs better than the PyClone approximate model on overall “dependent” and “independent” data, with the confidence of 95.4%.

4.5 Hardware Information & Computational Time

All above experiments were run on Monash Campus Cluster (MCC). MCC is Linux computer cluster for general-purpose use for many faculties in Monash University [35]. Intel Xeon and AMD Opteron are used as CPUs in the cluster, and memory from 4GB to 1TB can be allocated to a task. According to the wiki page of MCC, the nodes allocated to our tasks use up to 16 GB and eight CPU cores [36]. Further exact information about what CPUs are allocated to our tasks is not available.

Although we did not do a complete set of experiments on the aspect of computational time, we provide some rough results here for future reference. The time taken with respect to different setups is listed in Table 4.8.

| Genome length | #iterations | #switches | #chains | #samples | Time taken |
|-----------------|-------------|-----------|---------|----------|---------------|
| 3×10^4 | 1000 | 20 | 2 | 5 | 12 mins |
| 3×10^4 | 1000 | 20 | 3 | 5 | 20 mins |
| 3×10^4 | 1000 | 20 | 5 | 5 | 33 mins |
| 3×10^5 | 1000 | 20 | 2 | 1 | 29 mins |
| 3×10^5 | 1000 | 20 | 3 | 1 | 36 mins |
| 3×10^5 | 1000 | 20 | 5 | 1 | 58 mins |
| 3×10^5 | 5000 | 20 | 2 | 1 | 2 hrs 25 mins |
| 3×10^5 | 5000 | 20 | 3 | 1 | 4 hrs 45 mins |
| 3×10^5 | 5000 | 20 | 5 | 1 | 6 hrs 55 mins |

TABLE 4.8: Computational time recorded for different setups. The time is recorded by MCC automatically.

4.6 Summary of All Experiments

Two major categories of experiments were carried out in this project to evaluate our model and inference. All experiments used synthetic data as input, so we described how to generate the synthetic data first. Then the two major experiments include evaluating the model by itself with different setups, and the comparison between our model and a model implemented as an approximation to PyClone model. Results showed that our model performed well when two clones existed in the tumor (90% accuracy), but the accuracy decreased while the number of clones increases. As any machine learning model, the more samples included as observation, the more accurate the results are supposed to be. Therefore we evaluated how many samples the model needed to have a good trade-off between the accuracy and computational time. Results showed that our model

performed well even when the data was sparse and we determined five samples was the optimal number of samples. We also evaluated the model on different tissues (data generated from different assignments of genotypes in the model). We found that our model stable across different tissues and the model can be considered reliable when it was applied to different types of cancers in practical use.

To compare our model with another state-of-the-art computational model, we implemented a model that can be used as an approximation to the PyClone model. We could not use PyClone model directly because the input data and assumption made are different from our model. Results show that our model outperformed the approximation on average with over 95.4% confidence using Student's t test.

Chapter 5

Conclusion and Future Work

5.1 Contribution

This thesis presents a novel statistical model, which we call Het-FHMM for identifying and quantifying the intra-tumor heterogeneity in tumor. The model is based on the factorial hidden Markov model which is a variation of the well-known hidden Markov model. It models the information relationship between neighboring mutations (transition probabilities), and the relationship between the hidden genotype of the mutation and the sequencing data for that position (observation probabilities). The input to the model is the alignment file from next generation sequencing data, and the model outputs the genotypes of mutations in all clones and cellular prevalence of each clone. The original model was postulated by my supervisor Dr. Gholamreza Haffari. The inference of our statistical model is done by Gibbs sampling and exponentiated gradient descent (EG), pursuing the maximum likelihood of the whole model. The inference task is to find the most probable genotype for each genotype variable and the cellular prevalence S for each clone.

Our model contributes to the existing researches by resolving the assumption of non-overlapping of mutations among clones. In previous studies, mutations are assumed to not exist in more than one clone, but our model allows different mutations to be shared by different clones. Our model also captures the relationship between neighboring mutations, which is omitted by many other studies.

Having evaluated our model, we found that our model has a high accuracy when the number of chains is low. More specifically, the accuracy is around 90% when there are two chains. When the number of chains increases to 3 or 5, the result becomes less satisfactory. We also compared our model with an approximate version of another model

called PyClone¹. The result shows that our model is more accurate than PyClone with 95.4% of confidence, again with the improvement at lower number of clones.

5.2 Future Work

Although our model provides reliable results and overcomes the issue that arises from assuming mutations cannot be shared between different clones, there is much work to be done in the future. Firstly, the aim of the model is to find different clones in the tumor, but at the moment we have to fix the number of clones before we start the inference. The first thing that needs to be done is to extend the approach to do non-parametric inference so that the number of clones (chains in the model) can be decided dynamically. Secondly, as presented in Chapter 4, both of our two evaluation methods focus on the accuracy of genotypes. However, inferring the proportions of clones is also one of the aims of the project. We need to design an appropriate evaluation mechanism for checking the cellular prevalence in the future. Another thing that we plan to do next step is to find a better inference algorithm to replace Gibbs sampling, because the accuracy of Gibbs sampling is just a simple inference to start with and more advanced inference techniques aided by Beam sampling for infinite HMM [37], linear programming algorithms and column generation[38] may give improved performance. Fourth, we employed the transition probability (formula 3.1) in the model directly from [33], but it may not capture the truth accurately. In the future, we could make it more robust by modifying some parameters in the formula. Experiments could be done with respect to different modifications to evaluate the performance.

Lastly, we have to carry out the experiments on the real data. To evaluate how the model performs on real data, we also need to find another appropriate evaluation mechanism, because we do not have the ground truth of the real data.

¹Since we could not directly use PyClone for comparison due to different input and assumptions, we implemented another model which can be seen as an approximation to PyClone.

Appendix A

Gibbs Sampling

Gibbs Sampling Algorithm

```
1: for  $i = 1$  to  $K$  do
2:   for  $t = 1$  to  $T$  do
3:      $G_{t,i}$  = pick random genotype
4:   end for
5: end for
6: while not Converged do
7:   for  $t = 1$  to  $T$  do
8:     for  $i = 1$  to  $K$  do
9:       for  $g = 1$  to 18 do
10:        //To get the posterior we need for probabilities, which are calculated as
        follows
11:         $Probability\_at = P(a_t | G_{t,0}, G_{t,1}, \dots, G_{t,K})$ 
12:         $Probability\_lt = P(l_t | G_{t,0}, G_{t,1}, \dots, G_{t,K})$ 
13:        calculate  $P(G_{t,i}[g] | P(G_{t-1,i}))$ 
14:        calculate  $P(G_{t+1,i} | P(G_{t,i}[g]))$ 
15:         $Probability\_trans[g] = P(G_{t-1,i}[g]) * P(G_{t+1,i}[g])$  //put together
16:         $Probability\_posterior[g] = P\_trans[g] \times P\_at \times P\_lt$ 
17:      end for
18:       $normalize(P\_posterior)$ 
19:       $G_{t,i} = pick\_genotype(P\_posterior)$  //randomly sample one genotype from
      the distribution
20:    end for
21:  end for
22: end while
```

Appendix B

Full Experiment Results

| 1 sample | 100;200 | 20;1000 | 20;5000 | 3 samples | 100;200 | 20;1000 | 20;5000 |
|----------|---------|---------|---------|-----------|---------|---------|---------|
| 1 | 77.9 | 77.9 | 77.2 | | 89.65 | 89.65 | 88.62 |
| 2 | 90 | 85.2 | 88.3 | | 94.5 | 94.5 | 93.1 |
| 3 | 87.2 | 88.3 | 95.5 | | 94.8 | 93.8 | 82.4 |
| 4 | 86.3 | 87.4 | 90.2 | | 92.2 | 92.2 | 87.5 |
| 5 | 90.7 | 90.7 | 71 | | 87.6 | 88.6 | 94.8 |
| Average | 86.42 | 85.9 | 84.44 | | 91.75 | 91.75 | 89.284 |

TABLE B.1: Genome length= 3×10^5 Cellular prevalence: 40%,60%. Values are in percentage.

| 1 sample | 100;200 | 20;1000 | 20;5000 | 3 samples | 100;200 | 20;1000 | 20;5000 |
|----------|---------|---------|---------|-----------|---------|---------|---------|
| 1 | 38.1 | 41.8 | 49.1 | | 43.5 | 42.7 | 42.4 |
| 2 | 37.3 | 41.1 | 39.5 | | 39.7 | 37.6 | 40.8 |
| 3 | 38.7 | 46.7 | 38.7 | | 44.5 | 43.7 | 40 |
| 4 | 46.9 | 39.2 | 37.1 | | 45.3 | 40.3 | 43.5 |
| 5 | 38.4 | 38.7 | 36.8 | | 45.1 | 36 | 36.8 |
| Average | 39.88 | 41.5 | 40.24 | | 43.62 | 40.06 | 40.7 |

TABLE B.2: Genome length= 3×10^5 Cellular prevalence: 20%,30%,50%. Values are in percentage.

| 1 sample | 100;200 | 20;1000 | 20;5000 | 3 samples | 100;200 | 20;1000 | 20;5000 |
|----------|---------|---------|---------|-----------|---------|---------|---------|
| 1 | 44 | 43 | 37.2 | | 38.2 | 37.7 | 40.4 |
| 2 | 38.1 | 47.5 | 43.2 | | 40.5 | 45.1 | 46.3 |
| 3 | 36.7 | 59.5 | 35.3 | | 40.5 | 45.1 | 46.3 |
| 4 | 40.4 | 44 | 45.4 | | 40.2 | 39.3 | 41.8 |
| 5 | 46.1 | 37 | 34.9 | | 47.2 | 48.4 | 44.9 |
| Average | 41.06 | 46.2 | 39.2 | | 41.32 | 43.12 | 43.94 |

TABLE B.3: Genome length= 3×10^5 Cellular prevalence: 40%,30%,30%. Values are in percentage.

| 1 sample | 100;200 | 20;1000 | 20;5000 | 3 samples | 100;200 | 20;1000 | 20;5000 |
|----------|---------|---------|---------|-----------|---------|---------|---------|
| 1 | 29.4 | 25.2 | 30.4 | | 38 | 30.2 | 34.3 |
| 2 | 32.9 | 31.9 | 25.7 | | 42 | 34.8 | 32.3 |
| 3 | 28.5 | 32.9 | 41.1 | | 42 | 34.6 | 32.3 |
| 4 | 27.7 | 27.5 | 30.2 | | 24.8 | 36.7 | 38 |
| 5 | 32.1 | 33.8 | 25.6 | | 30.2 | 26.2 | 36.2 |
| Average | 30.12 | 30.26 | 30.6 | | 35.4 | 32.5 | 34.62 |

TABLE B.4: Genome length= 3×10^5 Cellular prevalence: 20%,10%,20%,30%,20%. Values are in percentage.

| | #Sample | exp1 | exp2 | exp3 | exp4 | exp5 | Average |
|-------------------------|---------|------|------|------|------|------|---------|
| 2chains/0.4_0.6/tissue1 | 2 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 0.80 |
| 2chains/0.4_0.6/tissue1 | 3 | 1.00 | 1.00 | 0.50 | 0.88 | 0.50 | 0.78 |
| 2chains/0.4_0.6/tissue1 | 4 | 0.50 | 0.71 | 1.00 | 1.00 | 0.50 | 0.74 |
| 2chains/0.4_0.6/tissue1 | 5 | 0.50 | 0.50 | 1.00 | 0.50 | 0.50 | 0.60 |
| 2chains/0.4_0.6/tissue1 | 10 | 0.54 | 0.50 | 0.50 | 1.00 | 0.50 | 0.61 |
| 2chains/0.4_0.6/tissue1 | 20 | 0.96 | 0.50 | 0.54 | 0.50 | 0.83 | 0.67 |
| 2chains/0.4_0.6/tissue1 | 50 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 2chains/0.4_0.6/tissue2 | 2 | 0.50 | 0.50 | 0.67 | 0.50 | 0.69 | 0.57 |
| 2chains/0.4_0.6/tissue2 | 3 | 0.69 | 0.69 | 0.50 | 0.69 | 0.69 | 0.66 |
| 2chains/0.4_0.6/tissue2 | 4 | 0.69 | 0.69 | 0.69 | 0.64 | 0.69 | 0.68 |
| 2chains/0.4_0.6/tissue2 | 5 | 0.69 | 0.69 | 0.69 | 0.50 | 0.69 | 0.66 |
| 2chains/0.4_0.6/tissue2 | 10 | 0.69 | 0.50 | 0.69 | 0.69 | 0.69 | 0.66 |
| 2chains/0.4_0.6/tissue2 | 20 | 0.69 | 0.69 | 0.69 | 0.69 | 0.69 | 0.69 |
| 2chains/0.4_0.6/tissue2 | 50 | 0.69 | 0.69 | 0.69 | 0.69 | 0.69 | 0.69 |
| 2chains/0.4_0.6/tissue3 | 2 | 0.68 | 0.74 | 0.74 | 0.74 | 0.68 | 0.71 |
| 2chains/0.4_0.6/tissue3 | 3 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 |
| 2chains/0.4_0.6/tissue3 | 4 | 0.74 | 0.76 | 0.76 | 0.71 | 0.74 | 0.74 |
| 2chains/0.4_0.6/tissue3 | 5 | 0.68 | 0.68 | 0.76 | 0.76 | 0.68 | 0.71 |
| 2chains/0.4_0.6/tissue3 | 10 | 0.71 | 0.76 | 0.76 | 0.71 | 0.76 | 0.74 |
| 2chains/0.4_0.6/tissue3 | 20 | 0.76 | 0.71 | 0.71 | 0.71 | 0.82 | 0.74 |
| 2chains/0.4_0.6/tissue3 | 50 | 0.74 | 0.74 | 0.74 | 0.74 | 0.74 | 0.74 |
| 2chains/0.4_0.6/tissue4 | 2 | 1.00 | 1.00 | 1.00 | 0.79 | 0.88 | 0.93 |
| 2chains/0.4_0.6/tissue4 | 3 | 1.00 | 0.58 | 1.00 | 0.79 | 1.00 | 0.88 |
| 2chains/0.4_0.6/tissue4 | 4 | 0.92 | 1.00 | 1.00 | 1.00 | 1.00 | 0.98 |
| 2chains/0.4_0.6/tissue4 | 5 | 1.00 | 1.00 | 1.00 | 1.00 | 0.88 | 0.98 |
| 2chains/0.4_0.6/tissue4 | 10 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 2chains/0.4_0.6/tissue4 | 20 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 2chains/0.4_0.6/tissue4 | 50 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 2chains/0.4_0.6/tissue5 | 2 | 0.97 | 0.97 | 0.87 | 0.87 | 0.87 | 0.91 |
| 2chains/0.4_0.6/tissue5 | 3 | 0.87 | 0.87 | 0.87 | 1.00 | 0.87 | 0.89 |
| 2chains/0.4_0.6/tissue5 | 4 | 1.00 | 0.97 | 0.97 | 0.97 | 0.87 | 0.95 |
| 2chains/0.4_0.6/tissue5 | 5 | 1.00 | 0.83 | 0.87 | 0.87 | 1.00 | 0.91 |
| 2chains/0.4_0.6/tissue5 | 10 | 0.87 | 1.00 | 1.00 | 0.87 | 0.87 | 0.92 |
| 2chains/0.4_0.6/tissue5 | 20 | 1.00 | 1.00 | 1.00 | 0.90 | 1.00 | 0.98 |
| 2chains/0.4_0.6/tissue5 | 50 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

TABLE B.5: Genome length= 3×10^4 Cellular prevalence: 40%,60%. 20×1000 iterations.

| | #Sample | exp1 | exp2 | exp3 | exp4 | exp5 | Average |
|-----------------------------|---------|------|------|------|------|------|---------|
| 3chains/0.4_0.3_0.3/tissue1 | 2 | 0.50 | 0.33 | 0.48 | 0.33 | 0.38 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue1 | 3 | 0.40 | 0.79 | 0.36 | 0.45 | 0.40 | 0.48 |
| 3chains/0.4_0.3_0.3/tissue1 | 4 | 0.40 | 0.48 | 0.40 | 0.36 | 0.36 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue1 | 5 | 0.50 | 0.43 | 0.40 | 0.43 | 0.50 | 0.45 |
| 3chains/0.4_0.3_0.3/tissue1 | 10 | 0.38 | 0.55 | 0.52 | 0.40 | 0.38 | 0.45 |
| 3chains/0.4_0.3_0.3/tissue1 | 20 | 0.43 | 0.43 | 0.52 | 0.45 | 0.38 | 0.44 |
| 3chains/0.4_0.3_0.3/tissue1 | 50 | 0.52 | 0.48 | 0.48 | 0.43 | 0.52 | 0.49 |
| 3chains/0.4_0.3_0.3/tissue2 | 2 | 0.33 | 0.77 | 0.33 | 0.33 | 0.33 | 0.42 |
| 3chains/0.4_0.3_0.3/tissue2 | 3 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 |
| 3chains/0.4_0.3_0.3/tissue2 | 4 | 0.36 | 0.33 | 0.33 | 0.33 | 0.33 | 0.34 |
| 3chains/0.4_0.3_0.3/tissue2 | 5 | 0.33 | 0.38 | 0.51 | 0.33 | 0.33 | 0.38 |
| 3chains/0.4_0.3_0.3/tissue2 | 10 | 0.33 | 0.36 | 0.41 | 0.46 | 0.33 | 0.38 |
| 3chains/0.4_0.3_0.3/tissue2 | 20 | 0.44 | 0.38 | 0.33 | 0.38 | 0.33 | 0.37 |
| 3chains/0.4_0.3_0.3/tissue2 | 50 | 0.36 | 0.36 | 0.38 | 0.38 | 0.33 | 0.36 |
| 3chains/0.4_0.3_0.3/tissue3 | 2 | 0.33 | 0.37 | 0.57 | 0.33 | 0.49 | 0.42 |
| 3chains/0.4_0.3_0.3/tissue3 | 3 | 0.55 | 0.37 | 0.41 | 0.65 | 0.33 | 0.46 |
| 3chains/0.4_0.3_0.3/tissue3 | 4 | 0.33 | 0.41 | 0.33 | 0.45 | 0.39 | 0.38 |
| 3chains/0.4_0.3_0.3/tissue3 | 5 | 0.33 | 0.55 | 0.33 | 0.41 | 0.59 | 0.44 |
| 3chains/0.4_0.3_0.3/tissue3 | 10 | 0.51 | 0.47 | 0.53 | 0.33 | 0.61 | 0.49 |
| 3chains/0.4_0.3_0.3/tissue3 | 20 | 0.39 | 0.37 | 0.43 | 0.41 | 0.39 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue3 | 50 | 0.43 | 0.43 | 0.39 | 0.33 | 0.39 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue4 | 2 | 0.46 | 0.46 | 0.42 | 0.35 | 0.33 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue4 | 3 | 0.38 | 0.67 | 0.54 | 0.33 | 0.40 | 0.46 |
| 3chains/0.4_0.3_0.3/tissue4 | 4 | 0.33 | 0.42 | 0.50 | 0.50 | 0.65 | 0.48 |
| 3chains/0.4_0.3_0.3/tissue4 | 5 | 0.48 | 0.33 | 0.44 | 0.35 | 0.38 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue4 | 10 | 0.42 | 0.33 | 0.46 | 0.42 | 0.56 | 0.44 |
| 3chains/0.4_0.3_0.3/tissue4 | 20 | 0.44 | 0.50 | 0.50 | 0.38 | 0.42 | 0.45 |
| 3chains/0.4_0.3_0.3/tissue4 | 50 | 0.40 | 0.46 | 0.38 | 0.50 | 0.46 | 0.44 |
| 3chains/0.4_0.3_0.3/tissue5 | 2 | 0.64 | 0.33 | 0.33 | 0.58 | 0.47 | 0.47 |
| 3chains/0.4_0.3_0.3/tissue5 | 3 | 0.55 | 0.58 | 0.41 | 0.39 | 0.39 | 0.46 |
| 3chains/0.4_0.3_0.3/tissue5 | 4 | 0.50 | 0.58 | 0.33 | 0.50 | 0.64 | 0.51 |
| 3chains/0.4_0.3_0.3/tissue5 | 5 | 0.62 | 0.38 | 0.38 | 0.68 | 0.33 | 0.48 |
| 3chains/0.4_0.3_0.3/tissue5 | 10 | 0.36 | 0.36 | 0.39 | 0.45 | 0.41 | 0.40 |
| 3chains/0.4_0.3_0.3/tissue5 | 20 | 0.41 | 0.39 | 0.42 | 0.39 | 0.41 | 0.41 |
| 3chains/0.4_0.3_0.3/tissue5 | 50 | 0.41 | 0.42 | 0.38 | 0.41 | 0.41 | 0.41 |

TABLE B.6: Genome length= 3×10^4 Cellular prevalence: 40%,30%,30%. 20×1000 iterations.

| | #Sample | exp1 | exp2 | exp3 | exp4 | exp5 | Average |
|-------------------------------------|---------|------|------|------|------|------|---------|
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 2 | 0.28 | 0.21 | 0.39 | 0.24 | 0.43 | 0.31 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 3 | 0.35 | 0.24 | 0.29 | 0.22 | 0.20 | 0.26 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 4 | 0.26 | 0.33 | 0.24 | 0.28 | 0.23 | 0.27 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 5 | 0.28 | 0.36 | 0.38 | 0.41 | 0.27 | 0.34 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 10 | 0.29 | 0.39 | 0.36 | 0.35 | 0.27 | 0.33 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 20 | 0.29 | 0.27 | 0.36 | 0.33 | 0.26 | 0.30 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue1 | 50 | 0.29 | 0.26 | 0.34 | 0.34 | 0.29 | 0.31 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 2 | 0.27 | 0.49 | 0.46 | 0.50 | 0.46 | 0.43 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 3 | 0.23 | 0.36 | 0.30 | 0.49 | 0.29 | 0.33 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 4 | 0.33 | 0.40 | 0.40 | 0.37 | 0.50 | 0.40 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 5 | 0.34 | 0.23 | 0.39 | 0.33 | 0.33 | 0.32 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 10 | 0.23 | 0.30 | 0.27 | 0.46 | 0.30 | 0.31 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 20 | 0.21 | 0.33 | 0.36 | 0.31 | 0.31 | 0.31 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue2 | 50 | 0.27 | 0.24 | 0.29 | 0.31 | 0.29 | 0.28 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 2 | 0.25 | 0.62 | 0.31 | 0.48 | 0.31 | 0.39 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 3 | 0.40 | 0.34 | 0.37 | 0.31 | 0.34 | 0.35 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 4 | 0.26 | 0.31 | 0.38 | 0.31 | 0.42 | 0.34 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 5 | 0.34 | 0.31 | 0.23 | 0.34 | 0.23 | 0.29 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 10 | 0.32 | 0.49 | 0.42 | 0.42 | 0.37 | 0.40 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 20 | 0.35 | 0.38 | 0.20 | 0.45 | 0.42 | 0.36 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue3 | 50 | 0.25 | 0.29 | 0.34 | 0.26 | 0.35 | 0.30 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 2 | 0.34 | 0.24 | 0.24 | 0.44 | 0.38 | 0.33 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 3 | 0.48 | 0.32 | 0.38 | 0.30 | 0.36 | 0.37 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 4 | 0.34 | 0.38 | 0.46 | 0.26 | 0.38 | 0.36 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 5 | 0.48 | 0.48 | 0.24 | 0.32 | 0.26 | 0.36 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 10 | 0.24 | 0.46 | 0.44 | 0.44 | 0.40 | 0.40 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 20 | 0.26 | 0.38 | 0.48 | 0.36 | 0.36 | 0.37 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue4 | 50 | 0.36 | 0.32 | 0.36 | 0.32 | 0.34 | 0.34 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 2 | 0.33 | 0.34 | 0.38 | 0.25 | 0.35 | 0.33 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 3 | 0.40 | 0.33 | 0.40 | 0.35 | 0.34 | 0.36 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 4 | 0.41 | 0.30 | 0.24 | 0.43 | 0.29 | 0.33 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 5 | 0.28 | 0.24 | 0.31 | 0.45 | 0.29 | 0.31 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 10 | 0.46 | 0.35 | 0.28 | 0.38 | 0.38 | 0.37 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 20 | 0.34 | 0.34 | 0.34 | 0.38 | 0.36 | 0.35 |
| 5chains/0.2_0.1_0.2_0.3_0.2/tissue5 | 50 | 0.35 | 0.31 | 0.30 | 0.29 | 0.33 | 0.32 |

TABLE B.7: Genome length= 3×10^4 Cellular prevalence: 20%,10%,20%,30%,20%.
20×1000 iterations.

| | fmmm | exp1 | exp2 | exp3 | exp4 | exp5 | Average |
|---------------------|------|------|------|------|------|------|---------|
| 5chains/dependent | 0.23 | 0.20 | 0.34 | 0.27 | 0.22 | | 0.25 |
| 5chains/independent | 0.34 | 0.31 | 0.29 | 0.28 | 0.31 | | 0.30 |
| 2chains/dependent | 0.58 | 1.00 | 1.00 | 1.00 | 1.00 | | 0.92 |
| 2chains/independent | 0.82 | 0.79 | 0.82 | 0.92 | 0.79 | | 0.83 |
| 3chains/dependent | 0.40 | 0.36 | 0.55 | 0.50 | 0.50 | | 0.46 |
| 3chains/independent | 0.44 | 0.44 | 0.38 | 0.42 | 0.40 | | 0.42 |
| PyClone | | | | | | | |
| 5chains/dependent | 0.26 | 0.23 | 0.26 | 0.27 | 0.28 | | 0.26 |
| 5chains/independent | 0.28 | 0.29 | 0.35 | 0.29 | 0.28 | | 0.30 |
| 2chains/dependent | 0.75 | 0.71 | 0.75 | 0.88 | 0.79 | | 0.78 |
| 2chains/independent | 0.74 | 0.71 | 0.76 | 0.79 | 0.79 | | 0.76 |
| 3chains/dependent | 0.40 | 0.45 | 0.36 | 0.38 | 0.45 | | 0.41 |
| 3chains/independent | 0.42 | 0.36 | 0.38 | 0.40 | 0.42 | | 0.40 |

TABLE B.8: Genome length= 3×10^4 Cellular prevalence: 40%,30%,30%. 20×1000 iterations. Baseline comparison.

Bibliography

- [1] National Cancer Institute. What is cancer, 08 2013.
- [2] American Cancer Society. Cancer facts & figures 2013. Technical report, American Cancer Society, 2013.
- [3] Australian Institute of Health and Welfare. Cancer in Australia. Technical report, Australian Institute of Health and Welfare, 2012.
- [4] Paulien Hogeweg. The roots of bioinformatics in theoretical biology. *PLoS computational biology*, 7(3):e1002021, 2011.
- [5] Andriy Marusyk, Vanessa Almendro, and Kornelia Polyak. Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, 12(5):323–334, 2012.
- [6] GH Heppner, DL Dexter, T DeNucci, FR Miller, and P Calabresi. Heterogeneity in drug sensitivity among tumor cell subpopulations of a single mammary tumor. *Cancer research*, 38(11 Part 1):3758–3763, 1978.
- [7] Robert A. Weinberg. *The biology of cancer*, chapter 2, pages 42–50. Garland Science, Taylor & Francis Group, LLC, New York, 2007.
- [8] Nicholas Navin, Alexander Krasnitz, Linda Rodgers, Kerry Cook, Jennifer Meth, Jude Kendall, Michael Riggs, Yvonne Eberling, Jennifer Troge, Vladimir Grubor, et al. Inferring tumor progression from genomic heterogeneity. *Genome research*, 20(1):68–80, 2010.
- [9] Nicholas Navin, Jude Kendall, Jennifer Troge, Peter Andrews, Linda Rodgers, Jeanne McIndoo, Kerry Cook, Asya Stepansky, Dan Levy, Diane Esposito, et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90–94, 2011.
- [10] Marco Gerlinger, Andrew J Rowan, Stuart Horswell, James Larkin, David Endesfelder, Eva Gronroos, Pierre Martinez, Nicholas Matthews, Aengus Stewart, Patrick Tarpey, et al. Intratumor heterogeneity and branched evolution revealed by multi-region sequencing. *New England Journal of Medicine*, 366(10):883–892, 2012.

- [11] M Gerlinger and C Swanton. How darwinian models inform therapeutic failure initiated by clonal heterogeneity in cancer medicine. *British journal of cancer*, 103(8):1139–1143, 2010.
- [12] Christian Vogler and Dimitris Metaxas. Parallel hidden markov models for american sign language recognition. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 116–122. IEEE, 1999.
- [13] Layla Oesper, Ahmad Mahmood, and Benjamin J Raphael. THetA: Inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. *Genome biology*, 14(7):R80, 2013.
- [14] Lurdes Torres, Francim R Ribeiro, Nikos Pandis, Johan A Andersen, Sverre Heim, and Manuel R Teixeira. Intratumor genomic heterogeneity in breast cancer with clonal divergence between primary carcinomas and lymph node metastases. *Breast cancer research and treatment*, 102(2):143–155, 2007.
- [15] Serena Nik-Zainal, Peter Van Loo, David C Wedge, Ludmil B Alexandrov, Christopher D Greenman, King Wai Lau, Keiran Raine, David Jones, John Marshall, Manasa Ramakrishna, et al. The life history of 21 breast cancers. *Cell*, 149(5):994–1007, 2012.
- [16] Wei Jiao, Shankar Vembu, Amit G. Deshwar, Lincoln Stein, and Quaid Morris. Modeling the clonal evolution of cancer from next generation sequencing data. *CoRR*, abs/1210.3384, 2012.
- [17] Rachel Julie Clark. The Double Helix the structure of DNA, December 2012. URL <http://naturaltreasuresofchristmas.wordpress.com/2012/12/05/the-fifth-day-of-christmas-watson-and-crick-the-double-helix-dna/>.
- [18] National Cancer Institute. Understanding cancer genomics, 08/02/2013 2013. URL <http://cancer.gov/cancertopics/understandingcancer>.
- [19] What is a copy number variant, and why are they important risk factors for ASD?, May 2014. URL <http://neurowiki2013.wikidot.com/individual:copy-number-variations>.
- [20] Frederick Sanger, Steven Nicklen, and Alan R Coulson. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.
- [21] Kris Wetterstrand. DNA sequencing costs: Data from the NHGRI genome sequencing program (GSP), April 2014. URL <https://www.genome.gov/sequencingcosts/>.

- [22] Ayshwarya Subramanian, Stanley Shackney, and Russell Schwartz. Novel multi-sample scheme for inferring phylogenetic markers from whole genome tumor profiles. In *Bioinformatics Research and Applications*, pages 250–262. Springer, 2012.
- [23] Sohrab P Shah, K-John Cheung, Nathalie A Johnson, Guillaume Alain, Randy D Gascoyne, Douglas E Horsman, Raymond T Ng, and Kevin P Murphy. Model-based clustering of array cgh data. *Bioinformatics*, 25(12):i30–i38, 2009.
- [24] Franck Picard, Emilie Lebarbier, Mark Hoebeke, Guillem Rigail, Baba Thiam, and Stéphane Robin. Joint segmentation, calling, and normalization of multiple cgh profiles. *Biostatistics*, 12(3):413–428, 2011.
- [25] Hua Ren, Wendy Francis, Amber Boys, Anderly C Chueh, Nick Wong, Phung La, Lee H Wong, Jacinta Ryan, Howard R Slater, and KH Andy Choo. Bac-based pcr fragment microarray: High-resolution detection of chromosomal deletion and duplication breakpoints. *Human mutation*, 25(5):476–482, 2005.
- [26] Li Yang and Martha Kahle. Analyzing array-based comparative genomic hybridization data, April 2014. URL <http://www.mathworks.com.au/company/newsletters/articles/analyzing-array-based-comparative-genomic-hybridization-data.html>.
- [27] Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273, 1997.
- [28] Kai Wang, Mingyao Li, Dexter Hadley, Rui Liu, Joseph Glessner, Struan FA Grant, Hakon Hakonarson, and Maja Bucan. Penncnv: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome research*, 17(11):1665–1674, 2007.
- [29] Andrew J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2): 260–269, 1967.
- [30] W. Jiao, S. Vembu, A. G. Deshwar, L. Stein, and Q. Morris. Inferring clonal evolution of tumors from single nucleotide somatic mutations. *ArXiv e-prints*, October 2012.
- [31] Andrew Roth, Jaswinder Khattra, Damian Yap, Adrian Wan, Emma Laks, Justina Biele, Gavin Ha, Samuel Aparicio, Alexandre Bouchard-Côté, and Sohrab P Shah. Pyclone: statistical inference of clonal population structure in cancer. *Nature methods*, 2014.

- [32] Sohrab P Shah, Andrew Roth, Rodrigo Goya, Arusha Oloumi, Gavin Ha, Yongjun Zhao, Gulisa Turashvili, Jiarui Ding, Kane Tse, Gholamreza Haffari, et al. The clonal and mutational evolution spectrum of primary triple-negative breast cancers. *Nature*, 486(7403):395–399, 2012.
- [33] Stefano Colella, Christopher Yau, Jennifer M Taylor, Ghazala Mirza, Helen Butler, Penny Clouston, Anne S Bassett, Anneke Seller, Christopher C Holmes, and Jiannis Ragoussis. Quantisnp: an objective bayes hidden-markov model to detect and accurately map copy number variation using snp genotyping data. *Nucleic acids research*, 35(6):2013–2025, 2007.
- [34] Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.
- [35] Philip Chan. The monash campus cluster, November 2013. URL <https://confluence-vre.its.monash.edu.au/display/MCC/The+Monash+Campus+Cluster>.
- [36] Simon Michnowicz. Hardware, November 2013. URL <https://confluence-vre.its.monash.edu.au/display/MCC/Hardware>.
- [37] Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095. ACM, 2008.
- [38] David Belanger, Alexandre Passos, Sebastian Riedel, and Andrew McCallum. Map inference in chains using column generation. In *NIPS*, pages 1853–1861, 2012.