UNIVERSITY OF CAPE COAST

COMPUTATIONAL METHODS FOR DENOISING HIGH-THROUGHPUT DATA

BY

BURI GERSHOM

Thesis submitted to the Department of Mathematics and Statistics of the School of Physical Sciences, College of Agriculture and Natural Sciences, University of Cape Coast in partial fulfillment of the requirements for award of Master of Philosophy degree in Mathematics

JULY, 2015

DECLARATION

Candidate's Declaration

I hereby declare that this thesis is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:..... **Date:**.....

Name: Gershom Buri

Supervisors' Declaration

We hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by the University of Cape Coast.

Principal Supervisor's Signature:..... **Date:**.....

Name: Dr. Wilfred Ndifon

Co-Supervisor's Signature:..... **Date:**.....

Name: Dr. Bismark Kwao Nkansah

ABSTRACT

T-cell diversity has a great influence on the ability of the immune system to recognise and fight the wide variety of potential pathogens in our environment. The current state of art approach to profiling T-cell diversity involves high-throughput sequencing and analysis of T-cell receptors (TCR). Although this approach produces huge amounts of data, the data has noise which might obscure the underlying biological picture. To correct these errors, two computational methods have been developed; a method of moments and a method based on Bayesian inference. Using simulated data, it is shown that Bayesian Inference is superior to the method of moments in terms of accuracy but the latter is preferable when time is a limiting factor as it is faster and adequately accurate. Furthermore, using high-throughput sequencing data, it is shown that significant differences exist between the raw and the denoised relative abundances of TCR V segments. For TCR J segments, however, the difference between raw and denoised data is minimal. This observation agrees with the fact that primers, which are used to enrich T-cell receptors before they are sequenced, and which are the main source of errors, are specific for TCR V segments.

ACKNOWLEDGEMENTS

I want to thank God the almighty for having brought me this far and led me to this achievement. Great gratitude to my supervisor, Dr. Wilfred Ndifon for the sacrifice and effort invested in me and to my success and most importantly for being patient with me. Also to my internal co-supervisor, Dr. Bismark K. Nkansah for all his inputs and guidance in the success of this work.

My further appreciation and gratitude goes to the entire staff of the African Institute for Mathematical Sciences (AIMS, Ghana) for their immense support and encouragement especially to Rhoda Mahamah.

Lastly, I will like to show sincere gratitude to my parents, siblings, and friends for their care, support, prayers and encouragement through the hard times.

God richly bless them all.

DEDICATION

I dedicate this work to my parents, Mr. Bethuel and Mrs. Grace Muni.

TABLE OF CONTENTS

DECLARATION ii

ABSTRACT iii

ACKNOWLEDGEMENTS iv

DEDICATION v

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER ONE: INTRODUCTION 1

Background of the Study 2

Problem Statement 8

Research Objectives 9

Significance of the Study 9

Delimitations 10

Limitations 10

Definition of Terms 10

Outline of the Thesis 11

CHAPTER TWO: LITERATURE REVIEW 13

Introduction 13

Estimates of the Potential Repertoire 14

Model Fitting of Sequence Data 19

Mathematical Theory 20

CHAPTER THREE: ESTIMATION OF PARAMETERS BY METHOD OF MOMENTS AND BAYESIAN INFERENCE 22

Introduction 22

Mechanistic Model 23

Method of Moments Approach for Estimating N, p and f 32

Application of the Method Based on Moments on Data Samples with Different Sizes 38

BAYESIAN INFERENCE FOR ESTIMATING PARAMETERS N, f AND p 40

Introduction to Bayesian Inference 40

Calculating Posterior Distributions 45

Bayesian Model for Estimating Parameters N, f and p of PCR Amplification 70

Summary 82

CHAPTER FOUR: DENOISING OF HIGH-THROUGHPUT DATA 85

Introduction 85

Simulated Datasets 85

Denoising Real Datasets 109

Summary 111

CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS 113

Conclusions 113

Recommendations 114

REFERENCES 115

APPENDICES 125

LIST OF TABLES

Table		Page
1	Results of parameter estimation using the method of moments for different sample sizes.	38
2	Pearson correlation coefficients between the estimated values and the actual values across all the datasets, using the two methods	108
3	Slopes of lines of fit for the estimated means using the two methods	109

LIST OF FIGURES

Figure		Page
1	Structure of an antibody (Wikipedia, 2015a).	4
2	The two types of T-ell receptors (lookfordiagnosis, 2015)	6
3	Plot of X_t against t for Markov chains with a standard normal distribution. The chains in a, b c and d were obtained using step sizes of $3.7, 0.2, 13$ and 1 respectively. See Appendix B for the code.	57
4	Corresponding autocorrelation functions for Markov chains with standard normal stationary distribution shown in Figure 3. From (a) to (b), the figures were obtained using step sizes of $3.7, 0.2, 13$ and 1 respectively.	59
5	History plot of N against t (Number of iterations) (a) and of p against t (b).	64
6	lag 1 autocorrelation as a function of the stepsize, A. The lowest autocorrelation for N was at $A = 0.036$	65
7	Full lag autocorrelation function for N and p.	66
8	Thinned samples for N (a) and p (b).	67
9	Autocorrelation function of thinned samples for N and p.	68
10	Variation of mean and 95 % credible interval with increase in sample size for N (a) and p (b).	69
11	History plot for N	72
12	History plot for p	73
13	History plot for f	73
14	The autocorrelation function for N. .	74
15	The autocorrelation function for p.	75
16	The autocorrelation function for f.	75

17 Variation of the mean and 95% credible interval with increase in sample size for N(a) , p (b) and f (c). 77

18 Chains of N when f is constant. 78

19 Chains of p when f is constant. 79

20 Autocorrelation function of p when f is constant. 79

21 Autocorrelation function of p when f is constant. 80

22 Variation of the mean and 95% credible interval with increase in sample size for N and p when f is constant. 81

23 Estimates for p using method of moments 88

24 Estimates of $N = 1, 2, \cdots 25$, using the method of moments. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence. A value of p, optimised over all (a_i, b_i) pairs is used to obtain each of the estimates. 89

25 Estimates for $N = 1, \cdots, 25$ for the 25×10 dataset using the method of moments, each obtained using its own p_i calculated from Eq. (4.3). 90

26 Estimates of $N = 1, 2, \cdots 50$, using the method of moments. A value of p, optimised over all (a_i, b_i) pairs, is used to obtain each of the estimates. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence. 92

27 Estimates of N for each of the ten 25×5 datasets, whereby each row has its own p_i value 93

28 Estimates of N for each of the ten 25×5 datasets, with a single optimised p. 94

29 Estimates of N for each of the ten 25×10 datasets, with a single optimised p. 94

30 Estimates of N for each of the ten 50×5 datasets, with a single optimised p. 95

31 Estimates of N for each of the ten 50×10 datasets, with a single optimised p. 95

32 Estimates of p using Bayesian inference. Each of the four estimates uses the entire dataset as evidence for p, rather than only the entries in each row. 96

33 Estimates of $N = 1, 2, \cdots 25$, using Bayesian Inference. A value of p optimised over all (a_i, b_i) pairs is used to obtain each of the estimates. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence. 98

34 Estimates of $N = 1, 2, \cdots 50$, using Bayesian Inference. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence. A value of p optimised over all (a_i, b_i) pairs is used to obtain each of the estimates. 100

35 Estimates of N for each of the ten 25×5 datasets. 101

36 Estimates of N for each of the ten 25×10 datasets. 101

37 Estimates of N for each of the ten 50×5 datasets. 102

38 Estimates of N for each of the ten 50×10 datasets. 102

39 Bayesian inference for $N = 1$ a)History plot of $N = 1$. 104

40 Bayesian inference for $N = 25$. 105

41 Bayesian inference for $N = 50$ 106

42 Bayesian inference for p. 107

43 Relative abundancies of V gene segments before and after denoising. 110

44 Relative abundancies of J gene segments before and after denoising. 111

CHAPTER ONE

INTRODUCTION

The immune system is our primary defence against pathogenic organisms and cells that have become malignantly transformed. One of the ways by which the immune system is able to perform this task, is by means of specialised blood cells called T-cells (a special type of white blood cells). The interaction between T-cells and pathogens is highly specific, that is, particular T-cells are able to recognise particular antigens. Therefore, the more diverse the T-cell repertoire, the higher the chance that any foreign body will be identified by the immune system. It is of great interest to computationally profile T-cell diversity in both sick and healthy individuals in order to better understand immunity, and its role in protecting against disease. The current state of art approach to profiling T-cell diversity involves high-throughput sequencing (HTS) and analysis of T-cell receptors (TCRs), that is, parts of the T-cell that perform actual antigen recognition. This approach produces huge amounts of data which inherently contains noise (extraneous or irrelevant data that gives inaccurate information). A key source of the noise is the enrichment of DNA that precedes sequencing. This noise might obscure the underlying biological picture.

Computational and mathematical methods are needed to denoise the data. The goal of the thesis is to develop new methods for performing such denoising of HTS data. In addition to study of TCR-diversity, which is the

primary motivation for developing these methods, they can also be applied to the analysis of gene expression dynamics more generally (i.e. using RNA-seq data).

The accuracy of the methods developed will be assessed by using simulated data. The methods will then be applied to real data.

1.1 Background of the Study

1.1.1 Immunology

Immunology is a branch of biomedical science that covers all aspects of the immune system in all multicellular organisms (nature, 2015). The immune system consists of a large network of cells and molecules distributed through out the body that detect and fight pathogenic agents. The human body provides an ideal environment for the growth and multiplication of most pathogens, the reason they try to break into it. It is the role of the immune system to deter them from succeeding and to seek and destroy them if they happen to enter the body. It is able to recognise and hold onto the memory of the antigens that it has encountered (Perelson & Weisbuch, 1997). Any substance that can elicit an immune response is called an antigen. This could be a microbe or part of a microbe. Operations of the immune system are realised by the action of tens to hundreds of different types of regulatory and effector molecules which communicate via cell to cell contact, and via the secretion of molecules (Perelson & Weisbuch, 1997). All the molecules that are important in the immune response have not yet been identified, but they include various cell surface receptors and molecules such as interleukins that can transmit signals between cells.

The immune system consists of various types of cells. The most important type is a group of small white blood cells known as lymphocytes. As a result, organs of the immune system are called lymphoid organs. They

are positioned around the body (NIAID, 2015). All blood cells, including lymphocytes, are created in the bone marrow and are transported throughout the body via the blood stream. Through the blood capillaries, white blood cells can navigate the tissues in search for antigens after which they return to the blood through the lymph (i.e a fluid bathing the cells of the body). During circulation, lymphocytes spend considerable time resident in lymphoid organs such as the bone marrow, the spleen, the lymph nodes and the thymus. Other clumps of lymphoid tissue are scattered else where in the body, especially in territories that act as gateways to the body, for example air passageways.

B cells and T-cells are the main two classes of lymphocytes. B lymphocytes make and secrete anti bodies; one of the major protective molecules of our bodies. Each B cell is programmed to make only one specific antibody. When stimulated, B cells give rise to many large plasma cells which manufacture millions of antibodies and secrete them into the blood. However these vast amounts of antibodies are powerless and cannot penetrate cells. They work by grabbing and sticking onto microbes, which makes it easier for the immune system to get rid of the microbes. An antigen matches an antibody much as a key matches a lock. In some cases, the matches are exact while in others, loose fits are sufficient to trigger a response. Whenever an antibody and an antigen interlock, the antibody marks the antigen for destruction.

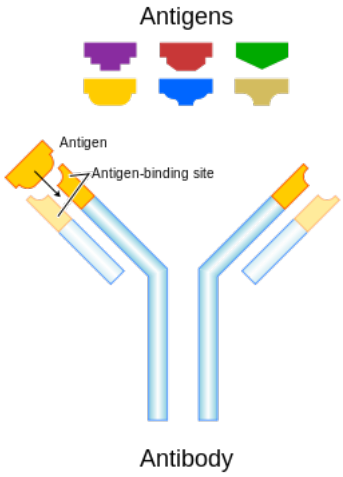


Figure 1: Structure of an antibody (Wikipedia, 2015a).

T-cells mainly function by interacting with other cells. They can be further subdivided into helper, regulatory and cytotoxic T-cells.

1.1.2 T-cells

T-cells do not recognize free floating antigens but rather, using their receptors, they can recognise fragments of antigens bound to cell surfaces. They contribute to immune defence in two ways; by regulating and directing immune responses, and by directly attacking infected or cancerous cells. Helper T-cells, which generally express a cell surface marker called CD4, coordinate immune responses by communicating with other cells. Some stimulate B cells in the vicinity to produce antibodies, others invite phagocytes to areas of infection, while others activate other T-cells. On the other hand, cytotoxic (killer) T-cells or CTLs (and a related type of cell called the natural killer cell), which generally express a cell surface marker called CD8, directly attack cells carrying certain foreign or abnormal molecules on their surface. In particular, they are useful for attacking viruses since viruses often hide from other parts of the immune system while they multiply inside an infected cell.

It is common for T-cells to only recognize an antigen if it is carried on the surface of a cell by one of the body's major histocompatibility com-

plex (MHC) molecules (Zinkernagel & Doherty, 1997). MHC molecules are proteins utilised by T-cells to discern between the self and non self i.e the body's own cells and the foreign cells.

The most important feature of T-cells are receptor molecules on their surfaces that can recognise antigens. For the case of B cells, the receptor is an immunoglobulin molecule (similar to an antibody molecule) while in the case of T-cells, the receptor is called the T-cell receptor or simply TCR. In the immune system recognition occurs at molecular level and depends on the complementarity in shape between the binding region of the receptor and the recognizable region of the antigen called the epitope. The binding between TCR and antigen is of relatively low affinity and is degenerate (i.e many TCR recognize the same antigen and many antigens are recognized by the same TCR).

1.1.3 T-cell Diversity

Each lymphocyte is estimated to have between 10^4 and 10^5 receptor molecules that can detect antigens, all of which are of the same shape. Given that each T-cell has receptors with one specificity, and that there are an enormous variety of organisms that can infect us, the immune system needs to generate vast numbers of T-cells, each possessing a different TCR. The TCR is composed of two different protein chains, that is, it is a heterodimer. In 95% of T-cells, this consists of an alpha (α) and beta (β) chain, whereas in 5% of T-cells this consists of gamma (γ) and delta (δ) chains (Wikipedia, 2015g). This ratio may change during diseased states. An intricate genetic machine lies behind the construction of the receptors and to good approximation ensures that the receptors expressed on different lymphocytes have different randomly chosen shapes. Generation of TCR diversity is based mainly on somatic recombination of the DNA encoded segments in individual T-cells which occurs in the thymus. TCRs

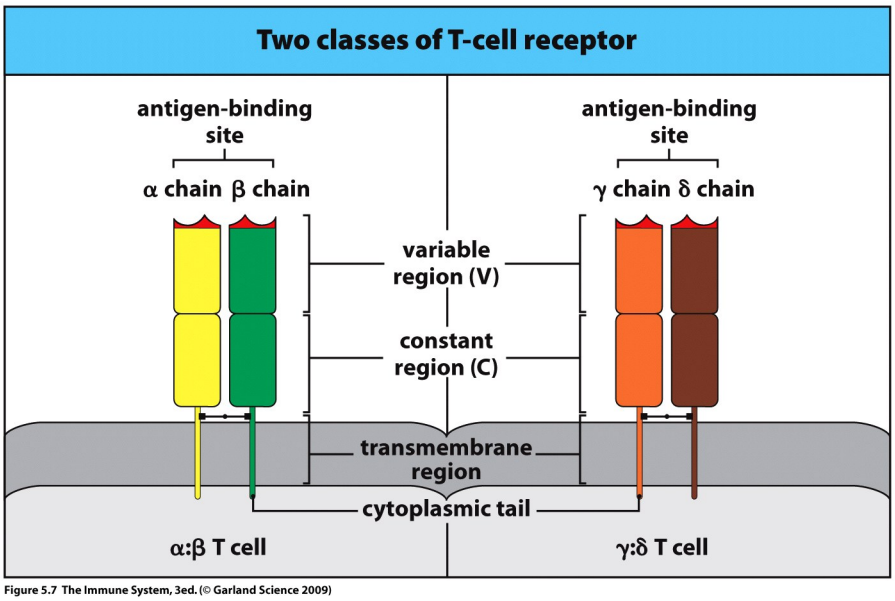


Figure 2: The two types of T-ell receptors (lookfordiagnosis, 2015)

have unique antigen specificity. This is determined by the structure of the antigen-binding site formed by the α and β chains (Janeway et al., 2001). The TCR alpha chain is generated by VJ recombination, whereas the beta chain is generated by VDJ recombination. In this sense, V is the variable gene segment, J is the joining gene segment and D is the diversity segment of the TCR. In the same way, the TCR gamma chain is generated by VJ recombination, whereas the TCR delta chain is generated by VDJ recombination (Wikipedia, 2015g). The intersection of the V and J (or VDJ)regions corresponds to the CDR3 (the complementarity determining region) region which is important for peptide recognition. The unique combination of the segments at this region, together with random and palindromic nucleotide additions, account for the great diversity observed.

1.1.4 High-throughput Sequencing

DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule (Wikipedia, 2015c). It includes any technology that is used to determine the order of the four bases i.e guanine (G), adenine (A), thymine (T), and cytosine (C), in a DNA strand. The

nucleotide sequence is the blueprint that contains the instructions for building an organism, and deciphering the information encoded in it is essential for almost all branches of biological research. The first DNA sequences were obtained by academic researchers in the early 1970s using laborious methods based on two-dimensional chromatography. Following the introduction of capillary electrophoresis (CE)-based Sanger sequencing (Sanger et al., 1978), researchers were able to elucidate any given genetic information from a biological system. The implementation of this cutting edge technology was however limited in terms of speed, throughput (a measure of how many units of data a system can process in a given amount of time.) and scalability (illumina, 2015). To over come these limitations a new technology called Next-Generation Sequencing (NGS) was developed.

In CE-based sequencing, bases of small fragments of DNA are successively identified from light signals emitted as each fragment is re-synthesized from a template DNA strand. NGS extends this process across millions of reactions rather than being restricted to one or a few DNA fragments. Such sequencing is often referred to as massive parallel sequencing and typically involves amplification of the DNA templates by the polymerase chain reaction (PCR). Amplification is necessary because significant amounts of a sample of DNA are necessary for molecular and genetic analyses. With current machines, a single run is capable of producing hundreds of gigabases of data, thus the term "high-throughput."

The output from next generation sequencing has increased at rate faster than even Moore's law of processing power; more than doubling every year (Institute, 2015). The last decade has witnessed rapid development of high-throughput sequencing methods, that is, fast, cheap ways to sequence and analyse large genomes (A genome is the complete set of an organism's DNA, including all of its genes. Each genome contains all of the information needed to build and maintain that organism. In humans, a copy of the entire

genome , which is more than 3 billion DNA base pairs, is contained in all cells that have a nucleus (Genomics, 2015)). A single run that could produce upto 1 gigabase of data in in 2007 had increased output to almost a terabase in 2011; nearly a $\times$1000 increase in 4 years (illumina, 2015). In addition, high throughput methods are capable of multiplexing in which multiple distinct samples are simultaneously sequenced in a single experiment. The ability to quickly generate such large volumes in a short time has enabled researchers advance from an idea, to data collection and then results in just a matter of days. Complete genomes can be sequenced in a hours or days for a couple of thousands of dollars, a great leap from the first human genome which in comparison, was sequenced for 13 years from 1990, with a staggering cost of 3 billion dollars (Collins et al., 2003).

However, while the cost and time have been greatly reduced, the new sequencing methods are significantly more prone to errors and other limitations. The errors are mainly as a result of amplification done prior to sequencing (Ndifon et al., 2012). Therefore, DNA sequencer output data has to be denoised in order to obtain the accurate biological picture (Rosen et al., 2012).

1.2 Problem Statement

During DNA sequencing, amplification of genetic material is necessary because significant amounts of a sample of DNA are necessary for molecular and genetic analyses. For accurate analysis, the relative abundances of particular clonotypes after amplification should be the same as before. Typically, different groups of DNA molecules are amplified at different rates such that their final abundances differ substantially from their initial abundances. This observation accrues from the use of a set of primers which introduce biases in the efficiencies of cDNA amplification. This makes difficult the accurate inference of the initial abundances which is generally of

the greatest biological interest. e.g "What is the relative frequency of each pathogen in the case of a co infection?".

1.3 Research Objectives

1.3.1 General Objective

The main objective of this study is to denoise sequence data (which has been amplified) by reconstructing the initial DNA copy numbers found in a biological sample before PCR amplification.

1.3.2 Specific Objectives

The following specific objectives are to be followed. These objectives are:

1. to develop computational methods for denoising sequence data.
2. to compare the methods developed.
3. to test the developed methods on simulated data.
4. to apply the developed methods on real data; including biased and unbiased data.

1.4 Significance of the Study

Of late, DNA sequencing is an essential tool for many basic and applied research applications. These range from basic applications such as parental testing and forensic identification, to more applied applications involving T-cell diversity and gene expression studies. More importantly, DNA sequencing is at the heart of personalised medicine, where patients are matched to their most appropriate drugs and the risk in terms of adverse effects is evaluated. The data generated by this technology however is

inherent with errors and denoising of such bias is necessary if information from the data is to be interpreted correctly. The methods developed in this thesis will address the problem of bias introduced during PCR amplification of the sequencing process, resulting in more accurate inferences from the data.

1.5 Delimitations

This work is limited to DNA sequencing using PCR technique for amplification as opposed to other alternative methods such as loop mediated isothermal amplification, nucleic acid sequence based amplification, strand displacement amplification and multiple displacement amplification.

1.6 Limitations

Some of the limitations that may affect the accuracy of the results are that:

1. the method of moments requires a large size of data to produce accurate results which are not available in reality.

2. The method based on moments requires choice of an optimal step size in the implemented Metropolis algorithm. This is difficult to choose manually and can affect the accuracy of the samples drawn.

1.7 Definition of Terms

Definition 1.7.1

Denoising is the extraction of a signal from a mixture of signal and noise.

Definition 1.7.2

Throughput is a measure of how many units of data a system can process

in a given amount of time.

Definition 1.7.3

DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule. It includes any technology that is used to determine the order of the four bases, that is, A,G,C and T in a strand of DNA.

1.8 Outline of the Thesis

The thesis is organised into six chapters. In addition to this chapter, Chapter Two reviews several work on T-cell diversity, applications of mathematics to immunology and denoising of HTS data. A critique of the previous methods of denosing is given and the mathematical theory of the thesis is introduced.

Chapter Three begins with introducing and explaining a mechanistic model developed by Ndifon et al. (2012), for correcting PCR induced bias. A method of moments approach for estimating the experimental parameters is then developed and applied on simulated data.

In Chapter Four, the Bayesian realm of inference is introduced and examples on estimation of parameters presented. A Bayesian model for estimating the experimental parameters is then developed and also applied on simulated data. The results are compared with those obtained with the method of moments.

In, Chapter Five synthetic datasets are generated in R and the two methods developed are applied. We then apply the methods to real data on V and J segments usage.

Chapter Six, concludes the findings, discusses at length results obtained in the analysis chapters and makes recommendations where necessary. Further work in the area of this thesis would also be explored briefly

with all other conclusions clearly shown.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

High-throughput sequencing, as a tool for exploring biological diversity has immense potential, but equally important are that challenges that arise from the analysis of the data. These technologies have made possible the characterization of the finest-scale diversity in heterogeneous populations of DNA, at low cost. However, each of these methods introduces random errors that are difficult to distinguish from genuine biological diversity. These errors are introduced during amplification and sequencing. In this thesis, we develop novel methods for denoising data generated by high throughput sequencing. One of the methods is based on moments, while the other is based on Bayesian inference. We intend to use the developed methods to denoise T-cell diversity data.

This chapter reviews past work on T cell diversity, the applications of mathematics in immunology, previous attempts to denoise high-throughput data, and lastly, introduces the mathematical theory of this thesis, based on the work of Ndifon et al. (2012).

2.2 Estimates of the Potential Repertoire

The potential repertoire is the number of possible distinct receptors that can be constructed given the genetic mechanisms involved. Bereck and Milstein estimated the potential repertoire of B cells to be of order 10^{11} in Berek & Milstein (1988) while Davis and Bjorkman estimated that for T-cells to be of order 10^{16} in Davis & Bjorkman (1988). A mouse contains of the order 10^8 B cells and 10^8 T-cells (Perelson & Weisbuch, 1997) and thus cannot contain all possible receptors. The number of different antibodies that can be expressed at any single time, called the expressed repertoire, is approximately 10^7. Such an immune system is capable of making of order 10^7 different receptors at any one time.

Since the most basic task of the immune system is to recognise foreign or abnormal particles and molecules, the diversity of the receptor types used by the immune system is fundamental. MacFarlene (Burnet, 1959) was the first to elucidate the algorithm that the immune system uses, called clonal selection. Clonal selection is the idea that only those cells that recognize the antigen proliferate, hence being selected at the expense of those that don't. Clonal selection applies to both T-cells and B cells. However, for this algorithm to work, the receptor population or repertoire should be diverse enough to recognize any foreign shape. Such a repertoire is said to be complete. The repertoire size, however, is continuously changing. The change is caused by the dynamic nature of the repertoire, ageing (Qi et al., 2014) and by the antigenic experience of the animal.

The immune system learns by raising the population size of lymphocytes that have successfully recognised antigens. The ability to learn the shapes of antigenic determinants from experience helps the immune system to maintain more lymphocytes bearing receptors of the appropriate shape so as to be prepared for future attacks by the same antigen.

Initial works on estimating the diversity of the human TCR repertoire involved sequencing just a few hundred sequences and then extending to the scale of the entire repertoire. The results were, a lower limit estimate of less than one million different TCR Beta genes (Arstila et al., 1999). With the advance of high-throughput sequencing, studies were able to estimate the diversity based on deeper sequencing depth. In Warren et al. (2011) an estimate of one million different TCR Beta sequences is obtained. By using a Poisson process model to infer unseen TCR beta sequences from deep sequencing data, Robins et al. (2009) estimated a diversity of 3-4 million TCR Beta sequences in the total T-cell populations of two healthy donors. In Qi et al. (2014), the lower bound for the total number of different TCR beta sequences in naive CD4 and CD8 T-cell repertoires of young adults, was estimated at 100 million.

2.2.1 Applications of Mathematics and Computing in Immunology

In the early years of its development, Immunology developed as a branch of medicine initially aimed at disease prevention through vaccination (Riedel, 2005). With such beginnings, one is not surprised that immunology was dominated by physicians and taught only in medical school. In the beginnings of the 20th century, these were slowly joined by chemists and later molecular biologists. The latest group of scientists to gain keen interest in immunology are those trained in physics and mathematics.

Mathematics provides a universal and quantitative framework for describing physical phenomena. Traced back to Bernoulli in 1760 Bernoulli (1760), mathematical modelling has been fundamental in predicting and rationalising disease spread and control Hamer (1906); Ross (1911); Kermack & McKendrick (1927); Anderson et al. (1992). Present literature is abundant with epidemiological models which have been very instrumental

to our understanding of outbreaks of epidemics and pandemics of a variety of pathogens. Notably, the principles articulated by Hamer (1906) and later extended by Ronald Ross (1911) and Kermack & McKendrick (1927) in 1927 form the genuine foundations of mathematical modelling. In those days, immunological knowledge of protection against infectious diseases was minimal. Diseases were differentiated from each other only through the reproduction number (Ro) values and estimates of the incubation period. Immunology is one of the branches of biology in which mathematical modelling has played a pivotal role since the 1960s (Marchalonis & Gledhill, 1968; Groves et al., 1969).

As insights into immune responses to infection emerged, models describing the interaction between pathogens and various parts of the immune system began to incorporate more specific aspects of the interaction in an attempt to improve on their predictability. Initially, these efforts were unsuccessful due to a shallow understanding of the immune system. Fortunately, beginning with the 1980s, simple but insightful models, describing the dynamic behaviour of pathogens and specific cells, were developed by several researchers (notably Alan Perelson, Robert May and Roy Anderson), particularly about HIV and influenza infection (Levin et al., 2004; May et al., 1988). Collectively, these contributions laid a foundation for mathematical immunology.

Mathematical models have been utilized in various domains of immunology such as antigen-receptor interactions (Stenberg & Nygren, 1988), hormone-receptor interactions (Rodbard, 1973), lymphocyte population dynamics (Thomas-Vaslin et al., 2008) and immune receptor signaling (Goldstein et al., 2004). Among others, Percus et al. (1993) used a mathematical model to predict the size of epitopes, while models for affinity maturation and for receptor cross linking are covered in Kauffman et al. (1988); Kepler & Perelson (1993) and Perelson & DeLisi (1980) respectively. One aspect of

immunology in which many investigators have had great interest, is the interaction between HIV and T-cells. These include Nelson & Perelson (1992), Hraba & Doležal (1990),Merrill Merrill (1989), Perelson et al. (1993) and many others.

Over the last two decades, the models used in immunology have changed significantly. Most of the classical models listed above were based on ordinary differential equations, difference equations and cellular automata. These models focused on "basic" dynamics derived from a modest number of reagent types, for example, one type of antigen with two T-cell populations. However, with the introduction of high throughput methods, availability of genomic data and faster computing power, immunological modelling has moved towards informatics (The science of computer information systems which studies the structure, algorithms, behaviour, and interactions of natural and artificial systems that store, process, access, and communicate information.). The nature of models has moved from being mainly ODEs of simple systems to voluminous use of Monte Carlo simulations.

Immuno-informatics is a field of science that encampuses high throughput genomic and bioinformatics (An interdisciplinary field of science,that combines computer science, statistics, mathematics, and engineering to study and process biological data.) approaches to immunology (Wikipedia, 2015b). Recent findings in genomic and proteomic (Proteomics is the large-scale study of proteins, particularly their structures and functions.) technologies have drastically transformed immunological research. Sequencing of human and other model organism genomes has produced large volumes of data relevant to immunology. Simultaneously, huge amounts of functional and clinical data are being reported in scientific papers and clinical records. Bioinformatics tools are used to manage and analyse such data in order to make high confidence predictions. In De Groot, Sbai, Saint Aubin, et al. (2002), work of several groups that are engaged in development of

immuno-informatics tools is reviewed and applications of these tools to vaccine discovery are illustrated.

The tools used in immunoinformatics can be classified into two groups; sequence based and structure-based. Sequence based approaches include all Motif based approaches (Meister et al., 1995), Quantitative matrices (Jesdale et al., 1997), Machine learning techniques which include neural networks Brusic et al. (1998), Hidden Markov model and Support vector machine. On the other hand structure-based approaches include molecular dynamics, threading algorithms (Altuvia et al., 1995), docking of peptides (Rosenfeld et al., 1995) and screening of peptide libraries (Hammer et al., 1994). Immuno-informatics has been useful in identification of antigens relevant to immune response, prediction of T-cell epitopes (Davenport et al., 1995), designing vaccines based on antigen and epitope identification (De Groot et al., 2001; De Groot, Sbai, Martin, & Berzofsky, 2002), reverse vaccinology(Sette & Rappuoli, 2010) and many others.

Still there exists challenging immunological questions to provide answers to and challenges in mathematical modelling as well, with the requirement of learning new mathematics and statistics. With the explosion in the amount data produced by high-throughput sequencing, uncovering the remaining genes and molecules that influence the behaviour of single lymphocytes is imminent. What will then remain will be to determine quantitatively, how the various cells of the immune system interact with each other to produce the coordinated response usually observed. One of the goals of modelling in immunology has been to deduce the macroscopic properties of the system from the properties and interactions among the elementary components.

2.3 Model Fitting of Sequence Data

Assuming that the read counts were sampled independently from a population with fixed proportions, the total counts for each gene (the term gene is used synonymously to an axon or transcript or clonotype) in a given sample (library) can be modelled using a Poisson distribution. As a result, the Poisson distribution has, in previous studies, been used to model count data (Marioni et al., 2008; Wang et al., 2010). The Possion distribution is parametrised by its mean which is equal to its variance. However, Robinson & Smyth (2007) and Nagalakshmi et al. (2008) report greater variation than the mean in real data. This is a condition of overdispersion, and it often fails to control type 1 errors (false positives) in statistical testing as noted in Anders & Huber (2010). Overdispersion is mainly as a result of amplification done prior to sequencing (Ndifon et al., 2012). Due to such extra-Poisson variation, better distributions such as the quasi-Poisson distribution (Venables & Ripley, 2002) and the Negative binomial distribution (Robinson & Smyth, 2007) have been proposed to model count data. Particularly, these two distributions have an extra parameter that estimates extra dispersion (under or lower) relative to a Poisson model. Previous modelling of count data by the Negative Binomial (NB) distribution has been phenomenologically justified by the fact that a NB distribution arises when each outcome is drawn from a Poisson distribution, whose rate parameter is is a gamma-distributed random variable. However, clear biological interpretations of the parameters of the Negative Binomial distribution were not always given. In contrast, the model by Ndifon et al. (2012) provides a less arbitrary rationale for applying the negative binomial distribution to RNA-seq data, and also allows us to interpret the distribution's parameters in terms of parameters of the PCR reaction used in any TCR-seq method.

In the next chapter, we introduce the mathematical theory onto which

this work is based.

2.4 Mathematical Theory

In Ndifon et al. (2012), a mechanistically motivated, bottom-up method is developed for correcting sequencing bias introduced by PCR amplification during library construction. These biases accrue from the use of different primers for each clonotype's Vβ gene, which introduce biases in the efficiencies of the cDNA amplification. Firstly, using stochastic dynamics and techniques from combinatorics, an equation for the probability distribution of a clonotype's copy number is derived, conditioned on the parameters of the PCR amplification model. This turns out to be a negative binomial distribution given by:

$$\begin{aligned} \mathrm{P}(x_i,t) &= \binom{x_i-1}{x_i-N_i} e^{-N_i r_i t}(1-e^{-r_i t})^{x_i-N_i} \\ &= \binom{x_i-1}{x_i-N_i} p^{N_i}(1-p)^{x_i-N_i}, \end{aligned} \tag{2.1}$$

a negative binomial distribution with probability of success, $p = e^{-r_i t}$ and predefined number of failures, $r = x_i - N_i$. The model parameters are N_i; the initial clonotype copy number, and $p = e^{-r_i t}$; the amplification efficiency.

Secondly, the effect of amplicon sub-sampling, which is implemented before the sequencing process, is incorporated into the negative binomial in Eq. (2.1). The result is a superposition of the negative binomial and the binomial distribution given by:

$$\mathrm{P}(x_i,t) = \sum_{j=0}^{min(N_i,x_i)} \binom{N_i+x_i-j}{N_i-j,j,x_i-j} \frac{((1-p)f)^{x_i-j} f^j (1-f)^{N_i-j}}{(p+(1-p)f)^{N_i+x_i-j}} \times \frac{N_i p_i^N}{N_i+x_i-j}, \tag{2.2}$$

which simplifies to;

$$\mathrm{P}(x_i,t) = \sum_{j=0}^{min(N,x_i)} \binom{N+x_i-j}{N-j,j,x_i-j} \frac{(1-p)^{x_i-j} f^{x_i} (1-f)^{N-j}}{(p+(1-p)f)^{N+x_i-j}} \frac{Np^N}{N+x_i-j} \quad (2.3)$$

In addition to parameters p and N_i in Eq. (2.1), we have parameter f, for the downward sampling.

In this thesis, we base on the probability distribution in 2.3 to derive computational methods for de-noising sequenced data.

The next chapter introduces and explains the mechanistic model for correcting sequencing bias introduced by PCR amplification during library construction developed by Ndifon et al. (2012). The method based on moments is then developed and applied to simulated data.

CHAPTER THREE

ESTIMATION OF PARAMETERS BY METHOD OF MOMENTS AND BAYESIAN INFERENCE

3.1 Introduction

This chapter has two main parts. In the first part, we will introduce and describe a mechanistic model for correcting PCR induced sequencing bias, developed by Ndifon et al. (2012). This is done in two steps. First, we attempt to derive an equation for the probability distribution of a clonotype's copy number, conditioned on parameters of the PCR reaction. The result is a negative binomial. Secondly, we incorporate the effect of amplicon sub-sampling into the derived distribution which results into a superposition of the negative binomial with a binomial distribution. We then develop a method of moments for estimating the parameters of the PCR reaction i.e p, f and N. We apply this method to simulated datasets of different sizes and compare the outputs of different datasets. We conclude by discussing the strengths and weaknesses of this method of moments.

In the second part of this chapter, the reader is introduced to the Bayesian realm of inference and examples are given. We describe a Bayesian model for estimating amplification parameters N, p and f, which entails

a Gibbs sampling algorithm and full conditionals. Lastly, we apply our Bayesian method to data samples with different sizes, discuss about the diagnostics used (i.e. autocorrelation), and the advantages of this new method over the one described in the first part of the chapter.

3.2 Mechanistic Model

3.2.1 Polymerase chain reaction amplification

Polymerase Chain Reaction (PCR) is a technology in molecular biology used to amplify a single copy or a few copies of a piece of DNA across several orders of magnitude, generating thousands to millions of copies of that DNA sequence. Because significant amounts of a sample of DNA are necessary for molecular and genetic analyses, it is easier to amplify isolated pieces of DNA than to extract the same region from a large amount of genetic material. PCR technology is commendable in a number of laboratory and clinical techniques, including DNA fingerprinting, detection of bacteria or viruses (particularly AIDS), and diagnosis of genetic disorders. The entire reaction involves using short DNA sequences (primers) to select the portion of the genome to be amplified. The temperature of the sample is repeatedly raised and lowered to help a DNA replication enzyme, to copy the target DNA sequence. The technique can produce a billion copies of the target sequence in just a few hours.

For accurate analysis, the relative abundances of particular clonotypes in the library after amplification should be the same as before amplification. However, this is not always the case. Typically, different groups of DNA molecules are amplified at different rates such that their final abundances differ substantially from their initial (or pre-amplification) abundances. These errors accrue from the use of a set of primers that are specific for each clonotype's Vβ gene, which introduce biases in the efficiencies of

DNA amplification. This makes difficult the accurate inference of the initial abundances, which is generally of the greatest biological interest. For example, answers to questions like, "What is the relative frequency of each pathogen in the case of a co-infection?" are sometimes of greater interest.

In this thesis, methods for correcting the biases introduced by PCR amplification are developed and examined. To begin with, a mechanistic method for correcting sequencing bias introduced by PCR amplification during library construction, developed in Ndifon et al. (2012).

Consider the fate of a particular clonotype say clonotype i, whose complementary DNA (cDNA) are found in the library that is amplified by PCR and then sequenced. cDNA is DNA synthesized by reverse transcriptase using RNA as a template. Let N_i represent the initial copy number of clonotype i (initial number of clones with phenotype i in the pre amplification library). Also, let $X_{i,t}$ be a random variable representing the copy number of clonotype i after t cycles of PCR. At each PCR cycle, each of the $X_{i,t}$ clonotypes is replicated with a probability corresponding to the PCR amplification efficiency. The difference in the efficiency of replication of distinct clonotypes, arises from the primers used for each clonotype's Vβ gene and other factors such as cDNA length. For convenience, we can replace $X_{i,t}$ with just X_i. The amplification process is modelled as a Markov jump process (Gardiner, 1985) to capture the small discrete increments in X_i during the small intervals of PCR amplification, where by the magnitude of each increment at a particular time interval depends only on the state of X_i in the preceding time interval. The model yielded an equation for the probability distribution of X_i conditioned on the initial copy number of clonotype i, N_i, its PCR amplification efficiency, r_i and the number of cycles, t. The model mimics very well the average dynamics of X_i, as well as fluctuations associated with both the discreteness and smallness of the copy number increments. Because of high fidelity of modern PCR machines, the

mutation rate is very small (especially if compared to sequencing error) and thus considered negligible.

Let

$$\mathrm{P}(X_i = x_i, t | X_i = x_{i'}, t') \tag{3.1}$$

denote the conditional probability that clonotype i has x_i copies at time t given that it had x'_i copies at time t'. From the discussion above, about the fate of a particular clonotype, this probability is increased by having $x_i - 1$ copies at the preceding state and is decreased by having x_i in the preceding state. This model is similar to the stochastic pure birth process in population dynamics, thus the master equation for the dynamics of probability distribution of X_i is obtained as;

$$\partial_t \mathrm{P}(X_i = x_i, t | X_i = x_{i'}, t') = r_i(x_i - 1)\mathrm{P}(X_i = x_i - 1, t | X_i = x'_i, t') - r_i x_i \mathrm{P}(X_i = x_i, t | X_i = x_{i'}, t'). \tag{3.2}$$

For convenience, we will write $\mathrm{P}(X_i = x_i, t | X_i = x_{i'}, t')$ as $p(x_i, t)$. Then Eq. (3.2) can be written as:

$$\partial_t p(x_i, t) = r_i(x_i - 1)p(x_i - 1, t) - r_i x_i p(x_i, t). \tag{3.3}$$

Here, r_i is the rate of increase of X_i, which is probabilistic and corresponds to the PCR amplification efficiency for clonotype i. In this case, it was assumed that r_i was constant due to relatively short duration of the PCR reactions considered. However r_i is bound to decrease if the PCR reaction is long enough for dynamics of cDNA copies to reach a plateau phase. Nevertheless, the analysis that was performed can be adapted to model PCR reactions in which r_i is assumed time dependent.

The mathematical concept of generating functions was employed to derive $p(x_{i,t})$ as follows; Consider a random variable X whose probability

mass function is $p(x_{i,t})$. Then the probability generating function of X is given by:

$$G(s,t) = \sum_{x_i=0}^{\infty} s^{x_i} p(x_i,t). \tag{3.4}$$

Substituting with $s = 1$ in the definition yields;

$$G(1,t) = \sum_{x_i=0}^{\infty} p(x_i,t) = 1. \tag{3.5}$$

Differentiating Eq. (3.4) with respect to s generates;

$$\partial_s G(s,t)|_{s=1} = \sum_{x_i=0}^{\infty} x_i s^{x_i-1} p(x_i,t)|_{s=1} = \langle x_i \rangle \tag{3.6}$$

$$\partial_s^2 G(s,t)|_{s=1} = \sum_{x_i=0}^{\infty} x_i(x_1-1) s^{x_i-2} p(x_i,t)|_{s=1} = \langle x_i(x_i-1) \rangle \tag{3.7}$$

Multiplying Eq. (3.2) by s^{x_i} and summing over all possible values of x_i gives;

$$\sum_{x_i=0}^{\infty} s^{x_i} \partial_t p(x_i,t) = r_i \sum_{x_i=0}^{\infty} s^{x_i}(x_i-1)p(x_i-1,t) - r_i \sum_{x_i=0}^{\infty} s^{x_i} x_i p(x_i,t), \tag{3.8}$$

$$\partial_t \sum_{x_i=0}^{\infty} s^{x_i} p(x_i,t) = r_i s^2 \sum_{x_i=0}^{\infty} s^{x_i-2}(x_i-1)p(x_i-1,t) - r_i s \sum_{x_i=0}^{\infty} s^{x_i-1} x_i p(x_i,t) \tag{3.9}$$

Using Eq. (3.4) and Eq. (3.6), Eq. (3.8) simplifies as follows;

$$\partial_t G(s,t) = r_i s^2 \partial_s G(s,t) - r_i s \partial_s G(s,t),$$

$$\partial_t G(s,t) = r_i s(s-1)\partial_s G(s,t).$$

s was eliminated from the coefficient of $\partial_s G(s,t)$ by making the substitution;

$$1 - \frac{1}{s} = e^y \text{ and } \Phi(y,t) = G(s,t), \tag{3.10}$$

such that;

$$\partial_t \Phi(y,t) = r_i s(s-1)\partial_s \Phi(y,t) = r_i s(s-1)\frac{\partial}{\partial y}\Phi(y,t)\frac{\partial y}{\partial s}. \tag{3.11}$$

From Eq. (3.10),

$$\frac{\partial y}{\partial s} = \frac{1}{s(s-1)}. \tag{3.12}$$

Substituting into Eq. (3.11), we obtain:

$$\partial_t \Phi(y,t) = r_i \partial_y \Phi(y,t) \tag{3.13}$$

The solution of Eq. (3.13) is of the form $\Psi[\exp(y+r_i t)]$. $G(s,t)$ can therefore be expressed as

$$G(s,t) = \Psi[s^{-1}(s-1)e^{r_i t}]. \tag{3.14}$$

If at $t = 0$, clonotype i has N_i copies, then $p(x,0) = 0$ if $x \neq N_i$ and $p(x,0) = 1$ otherwise. In light of this observation;

$$G(s,0) = \Psi[s^{-1}(s-1)] = \sum_{x_i=0}^{\infty} s^{x_i} p(x_i,0) = s^{N_i}. \tag{3.15}$$

Let

$$\theta = \frac{s}{s-1}, \Rightarrow s = \frac{\theta}{\theta - 1}.$$

Then

$$\Psi(\theta) = \left(\frac{\theta}{\theta-1}\right)^{N_i} \Rightarrow \Psi\left(\frac{s}{s-1}\right) = \left(\frac{\frac{s}{s-1}}{\frac{s}{s-1}-1}\right)^{N_i}$$

$$\Rightarrow \Psi\left(\frac{se^{-rt}}{s-1}\right) = \left(\frac{\frac{se^{-rt}}{s-1}}{\frac{se^{-rt}}{s-1}-1}\right)^{N_i}.$$

Therefore

$$G(s,t) = \left(\frac{se^{-r_it}}{1-s(1-e^{r_it})}\right)^{N_i},$$

and

$$G(s,t) = e^{-N_ir_it}s^{N_i}(1-(1-e^{-r_it})s)^{-N_i}. \tag{3.16}$$

The probability distribution of interest, $p(x,t)$, is given by the coefficients of s^{x_i} in $G(s,t)$. These can be obtained by expanding $G(s,t)$ in power series in s, that is;

$$s^{N_i}(1-(1-e^{-r_it})s)^{-N_i} = s^{N_i} + N_i(1-e^{-r_it})s^{N_i+1} + \frac{N_i(N_i+1)}{2!}(1-e^{-r_it})^2s^{N_i+2} + \frac{N_i(N_i+1)(N_i+2)}{3!}(1-e^{-r_it})^3s^{N_i+3} + \ldots$$

which after some algebra, yields;

$$\mathrm{P}(x_i,t) = \binom{x_i-1}{x_i-N_i}e^{-N_ir_it}(1-e^{-r_it})^{x_i-N_i} \tag{3.17}$$

$$= A(x_i-N_i, N_i, e^{-r_it}), \tag{3.18}$$

where A denotes a negative binomial distribution with probability of success, $p = e^{-r_i t}$ and predefined number of failures, $r = x_i - N$.

3.2.2 Subsampling of cDNAs amplified by the Polymerase Chain Reaction

Not all amplified cDNAs are always sequenced. Only a portion is randomly selected and eventually sequenced. Sub-sampling of amplicons can be done to save a portion for later use, to scale down a particular sample of clonotype to the capacity of the sequencer or in order to assure efficiency and accuracy of the machine. This has the advantage of ensuring that different cDNA clusters occupy different spatial locations in the sequencer so that the risk for between cluster interference in a florescence signal is minimal. However, because sub-sampling is random in nature, there is a chance of losing clonotypes with small initial copy numbers or those with low amplification efficiency. As the PCR reaction proceeds, the relative frequency of clonotypes with below average amplification rate decreases and as a result, such clonotypes stand a high chance of being missed during fixed rate sub-sampling. This can have a significant effect on the statistics of the clonotype repertoire and it is therefore rational to include the effect of sub sampling on the final sequencing output before analysing such data. Any amplification bias that could arise from inside the sequencer was ignored because universal primers are applied to all cDNAs, and also because each newly created cDNA cluster typically contributes a single sequence read to the final output.

A probability generating function was used to incorporate sub-sampling into the expression for the probability distribution of a clonotype's copy number derived in the previous section. The same could be achieved in a direct way but would complicate the subsequent task of calculating the moments of X_i. The master equation for the dynamics of the probability

distribution of X_i is given by:

$$\partial_\tau p(x_i,\tau) = -cx_i p(x_i,\tau) + c(x_i+1)p(x_i+1,\tau), \tag{3.19}$$

that is, having x_i copies at time τ is increased from losing copies from x_i+1 and decreased by losing copies from x_i. c is the probabilistic rate at which copies of clonotype i are lost during the process of sub sampling and $p(x_i,0)$ is given by Eq. (3.17). In principle, τ equals 1 time unit.

Multiplying through by s^{x_1} in Eq. (3.19) and summing over all x_i, we obtain;

$$\begin{aligned}\partial_\tau \sum_{x_i=0}^{\infty} s^{x_1} p(x_i,\tau) &= -c\sum_{x_i=0}^{\infty} x_i s^{x_1} p(x_i,\tau) + c\sum_{x_i=0}^{\infty}(x_i+1)s^{x_1}p(x_i+1,\tau) \\ &= -cs\sum_{x_i=0}^{\infty} x_i s^{x_1-1} p(x_i,\tau) + c\sum_{x_i=0}^{\infty}(x_i+1)s^{x_1}p(x_i+1,\tau).\end{aligned}$$

Following from Eq. (3.4) and Eq. (3.6), the dynamics of the probability generating function corresponding to Eq. (3.19) can be expressed as:

$$\partial_\tau \widetilde{G}(s,\tau) = c(1-s)\partial_s \widetilde{G}.$$

By making the substitutions $e^y = 1-s$ and $\Phi(y,\tau) = \widetilde{G}(s,\tau)$, we obtain;

$$\partial_\tau \Phi(y,\tau) = -c\partial_y(y,\tau), \tag{3.20}$$

the solution of which is function $\Psi[\exp(\ln(s-1)-c\tau)]$. For convenience we substitute $f = \exp(-c\tau)$, we obtain;

$$\widetilde{G}(s,\tau) = \Psi[(s-1)f].$$

Since $\widetilde{G}(s,0)$ is given by Eq. (3.16), then;

$$\widetilde{G,}(s,0) \quad = \quad \Psi[(s-1)] = e^{-N_i r_i t} s^{N_i} (1-(1-e^{-r_i t})s)^{-N_i}.$$

Let

$$\theta = s-1, \Rightarrow s = \theta + 1.$$

Then

$$\begin{aligned} \Psi[\theta] &= e^{-N_i r_i t}(\theta+1)^{N_i}(1-(1-e^{-r_i t})(\theta+1))^{-N_i}. \\ \Rightarrow \Psi[(s-1)f] &= e^{-N_i r_i t}(1+(s-1)f)^{N_i}(1-(1-e^{-r_i t})(1+(s-1)f))^{-N_i}. \end{aligned}$$

The generating function with subsampling is thus given by:

$$\widetilde{G}(s,\tau) \quad = \quad e^{-N_i r_i t}(1+(s-1)f)^{N_i}(1-(1-e^{-r_i t})(1+(s-1)f))^{-N_i}, \tag{3.21}$$

where f represents the fraction of sequenced amplicons. $p(x_i,t)$ was obtained from Eq. (3.21) using the formula for individual terms arising from the composition of two generating functions, that is;

$$[s^{x_i}]\, g(h) = \sum_{k=0}^{\infty} \left(\left[s^{x_i} \right] g \right) \left[s^{x_i} \right] h^k,$$

where the function g is given by Eq. (3.16) and function $h = 1+(s-1)f$. This resulted into a superposition of the negative binomial with the binomial distribution, that is;

$$\begin{aligned} \mathrm{P}(x_i,t) &= \sum_{j=0}^{\infty} \binom{j-1}{j-N_i} e^{-N_i r_i t}(1-e^{-r_i t})^{j-N_i} \binom{j}{x_i} f^{x_i}(1-f)^{j-x_i} \\ &= \sum_{j=0}^{\infty} \mathrm{A}(j-N_i, N_i, e^{-r_i t})\mathrm{C}(x_i, j, f), \end{aligned} \tag{3.22}$$

where A and C denote the negative binomial and binomial distributions respectively.

Alternatively, an expression for $p(x_i, t)$ can be obtained by first expanding $(1 + (s-1)f)^{N_i}$ in a Maclaurin series for s and then obtaining the coefficient of s^{x_i} in the resulting expression, that is;

$$\mathrm{P}(x_i, t) = \left[s^{x_i}\right] \sum_{j=0}^{\infty} \binom{j-1}{j-N_i} e^{-N_i r_i t} (1 - e^{-r_i t})^{j-N_i} \times \frac{e^{-N r_i t} s^j}{\left(1 - (1 - e^{-r_i t})\,(1 + (s-1)\,f)\right)^{N_i}}$$

$$= \sum_{j=0}^{min(N_i, x_i)} \binom{j-1}{j-N_i} e^{-N_i r_i t} (1 - e^{-r_i t})^{j-N_i} \binom{N_i + x_i - j - 1}{x_i - j} \times \frac{e^{-N_i r_i t}\left((1 - e^{-r_i t})\,f\right)^{x_j - j}}{\left(e^{-r_i t} + (1 - e^{-r_i t})\,f\right)^{N_i + x_i - j}}$$

$$= \sum_{j=0}^{min(N_i, x)} \binom{N_i + x - j}{N_i - j, j, x - j} \frac{\left((1-p)f\right)^{x-j} f^j \,(1-f)^{N_i - j}}{\left(p + (1-p)f\right)^{N_i + x - j}} \frac{N_i p_i^N}{N_i + x - j}. \tag{3.23}$$

Computationally, Eq. (3.23) is better than Eq. (3.22) because it requires evaluation of fewer terms; typically much less than x_i.

3.3 Method of Moments Approach for Estimating N, p and f

In this section we develop a novel method for estimating parameters N, p and f of the PCR reaction. From the generating function in Eq. (3.21), we obtain the first, second and third moments which in turn we use to obtain expressions for estimates of N, p and f.

In Eq. (3.21), substitute;

$$p_i = e^{-r_i t},$$

the PCR amplification efficiency for clonotype i. For convenience we drop the subscript i such that Eq. (3.21) becomes;

$$\widetilde{G}(s,\tau) = p^N(1+(s-1)f)^N(1-(1-p)(1+(s-1)f))^{-N}.$$

In the following subsections, we will use the notation $\langle X\rangle$ for expectation of X.

3.3.1 Expression for $\langle X\rangle$

$\langle X\rangle$ is obtained, from the generating function using the relation;

$$\langle X\rangle = \frac{\partial \widetilde{G}}{\partial s}\bigg|_{s=1}.$$

But;

$$\frac{\partial \widetilde{G}}{\partial s} = p^N(1+(s-1)f)^N Nf(1-p)\Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+1)} + \Big(1-(1-p)(1+(s-1)f)\Big)^{-N} Nfp^N(1+(s-1)f)^{N-1},$$

$$= Nfp^N(1+(s-1)f)^{N-1}\Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+1)} \Big((1+(s-1)f)(1-p) + \Big(1-(1-p)(1+(s-1)f)\Big)\Big)$$

$$= Nfp^N(1+(s-1)f)^{N-1}\Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+1)}. \quad (3.24)$$

Therefore;

$$\begin{aligned}\left.\frac{\partial \widetilde{G}}{\partial s}\right|_{s=1} &= Nfp^N\left(1-(1-p)\right)^{-(N+1)}, \\ &= \frac{Nf}{p}.\end{aligned}$$

And so;

$$\langle X\rangle = \frac{Nf}{p},$$

which implies that;

$$p = \frac{Nf}{\langle X\rangle}. \tag{3.25}$$

3.3.2 Expression for $\langle X^2\rangle$

$\langle X^2\rangle$ is obtained from the generating function using the relation;

$$\langle X^2\rangle = \left.\frac{\partial^2 \widetilde{G}}{\partial s^2}\right|_{s=1} + \left.\frac{\partial \widetilde{G}}{\partial s}\right|_{s=1}.$$

From Eq. (3.24);

$$\begin{aligned}\frac{\partial^2 \widetilde{G}}{\partial s^2} = Nfp^{-N}\Big[&(1+(s-1)f)^{N-1}(N+1)(1-p)f \times \\ &\Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+2)} + \\ \Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+2)}&(N-1)f(1+(s-1)f)^{N-2}\Big]\end{aligned}$$

$$\begin{aligned}= Nf^2p^N(1+(s-1)f)^{N-2}&\Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+2)} \\ \Big((1+(s-1)f)(1-p)(N+1) + (N-1)&\Big(1-(1-p)(1+(s-1)f)\Big)\Big).\end{aligned}$$

$$= Nf^2p^N(1+(s-1)f)^{N-2}\Big(1-(1-p)(1+(s-1)f)\Big)^{-(N+2)}$$
$$\Big((N-1)+2(1+(s-1)f)(1-p)\Big). \quad (3.26)$$

Thus;

$$\left.\frac{\partial^2\widetilde{G}}{\partial s^2}\right|_{s=1} = Nf^2p^Np^{-(N+2)}\Big((N-1)+2(1-p)\Big)$$
$$= \frac{Nf^2}{p^2}\Big(N+1-2p\Big). \quad (3.27)$$

Therefore;

$$\langle X^2\rangle = \frac{Nf^2}{p^2}\Big(N+1-2p\Big)+\langle X\rangle. \quad (3.28)$$

3.3.3 Expression for $\langle X^3\rangle$

From the generating function $\langle X^3\rangle$ can be obtained as follows;

$$\langle X^3\rangle = \left.\frac{\partial^3\widetilde{G}}{\partial s^3}\right|_{s=1}+3\left.\frac{\partial^2\widetilde{G}}{\partial s^2}\right|_{s=1}-2\left.\frac{\partial\widetilde{G}}{\partial s}\right|_{s=1}$$
$$= \left.\frac{\partial^3\widetilde{G}}{\partial s^3}\right|_{s=1}+3\left(\left.\frac{\partial^2\widetilde{G}}{\partial s^2}\right|_{s=1}+\left.\frac{\partial\widetilde{G}}{\partial s}\right|_{s=1}\right)-2\left.\frac{\partial\widetilde{G}}{\partial s}\right|_{s=1}$$
$$= \left.\frac{\partial^3\widetilde{G}}{\partial s^3}\right|_{s=1}+3\langle X^2\rangle-5\langle X\rangle. \quad (3.29)$$

From Eq. (3.26);

$$\frac{\partial^3\widetilde{G}}{\partial s^3} = Nf^2p^N\Bigg[\Big((N-2)f(1+(s-1)f)^{N-3}(1-(1-p)(1+$$
$$(s-1)f))^{-(N+2)}+(1+(s-1)f)^{N-2}(N+2)(1-p)f$$
$$(1-(1-p)(1+(s-1)f))^{-(N+3)}\Big)\Big((N-1)+2(1+(s-1)f)(1-p)\Big)$$
$$+\Big((1+(s-1)f)^{N-2}(1-(1-p)(1+(s-1)f))^{-(N+2)}\Big)2f(1-p)\Bigg].$$

It follows that;

$$\left.\frac{\partial^3 \widetilde{G}}{\partial s^3}\right|_{s=1} = Nf^2p^N\Bigg[\Big((N-2)fp^{-(N+2)} + (N+2)(1-p)fp^{-(N+3)}\Big)\Big(N-1+2(1-p)\Big) + p^{-(N+2)}2f(1-p)\Bigg]$$

$$= Nf^2p^N fp^{-(N+3)}\Bigg[\Big((N-2)p + (N+2)(1-p)\Big)\Big(N-1+2(1-p)\Big) + 2p(1-p)\Bigg]$$

$$= Nf^3p^{-3}\Bigg[\Big((N+2)-4p\Big)\Big(N-1+2(1-p)\Big) + 2p - 2p^2\Bigg]$$

$$= \frac{Nf^3}{p^3}\Big[(N+2)(N+1) - 6Np - 6p + 6p^2\Big].$$

Therefore;

$$\langle X^3\rangle = \frac{Nf^3}{p^3}\Big[(N+2)(N+1) - 6Np - 6p + 6p^2\Big] + 3\langle X^2\rangle - 5\langle X\rangle. \tag{3.30}$$

3.3.4 Expressions for N and f in terms of $\langle X^3\rangle$, $\langle X^2\rangle$ and $\langle X\rangle$

Substituting for p in Eq. (3.28) gives;

$$\begin{aligned}
\langle X^2\rangle &= \frac{Nf^2\langle X\rangle^2}{N^2f^2}\left(N+1-\frac{2Nf}{\langle X\rangle}\right)+\langle X\rangle \\
\langle X^2\rangle &= \frac{\langle X\rangle^2}{N}\left(\frac{(N+1)\langle X\rangle-2Nf}{\langle X\rangle}\right)+\langle X\rangle \\
X\langle X^2\rangle &= (N+1)\langle X\rangle^2-2Nf\langle X\rangle+N\langle X\rangle \\
f &= \frac{(N+1)\langle X\rangle^2+N\langle X\rangle-N\langle X^2\rangle}{2N\langle X\rangle} \\
f &= \frac{N(\langle X\rangle^2+\langle X\rangle-\langle X^2\rangle)+\langle X\rangle^2}{2N\langle X\rangle}. \qquad (3.31)
\end{aligned}$$

Substituting p into Eq. (3.30) gives;

$$\langle X^3\rangle = \frac{Nf^3\langle X\rangle^3}{N^3f^3}\left[(N+2)(N+1)-\frac{6N^2f}{\langle X\rangle}-\frac{6Nf}{\langle X\rangle}+\frac{6N^2f^2}{\langle X\rangle^2}\right] +3\langle X^2\rangle-5\langle X\rangle$$

$$= \frac{\langle X\rangle}{N^2}\left[(N^2+3N+2)\langle X\rangle^2-6N^2f\langle X\rangle-6N\langle X\rangle+6N^2\langle X\rangle^2\right] +3\langle X^2\rangle-5\langle X\rangle$$

$$N^2\langle X^3\rangle = \langle X\rangle\left[(N^2+3N+2)\langle X\rangle^2-6N^2f\langle X\rangle-6N\langle X\rangle+6N^2\langle X\rangle^2\right] +N^2(3\langle X^2\rangle-5\langle X\rangle).$$

We rearrange to obtain a quadratic function in N, that is;

$$N^2\left[6f^2\langle X\rangle-6f\langle X\rangle^2+\langle X\rangle^3+\langle X^2\rangle-5\langle X\rangle-\langle X^3\rangle\right]+ N\left[3\langle X\rangle^3-6f\langle X\rangle^2\right]+\langle X\rangle^3=0.$$

On substituting for f, we obtain;

$$N^2\left[-\frac{1}{2}\langle X\rangle^3 - \frac{7}{2}\langle X\rangle + \frac{3}{2}\frac{\langle X^2\rangle^2}{\langle X\rangle} - \langle X^3\rangle\right] + \frac{1}{2}\langle X\rangle^3 = 0. \tag{3.32}$$

Eq. (3.25), Eq. (3.31) and Eq. (3.32) are used to estimate parameters p, f and N respectively. From these equations, we notice that estimation of N entirely depends on means of the sequenced output, the estimation f depends on N and the estimation p depends on both N and p.

The developed method of moments was applied on data samples simulated using the probability distribution in Eq. (3.23) and the results obtained are presented in the next section.

3.4 Application of the Method Based on Moments on Data Samples with Different Sizes

Samples of different sizes were generated from the probability distribution derived in Eq. (3.23) using the *distr* package in R (see Appendix A 5.2). Parameter values of $N = 10, f = 0.3$ and $p = 0.001$ were used. The results are presented in Table 1 As the sample size increases, the estimate

Table 1: Results of parameter estimation using the method of moments for different sample sizes.

Sample size	N	f	p	p (with $f = 0.3$)
10^2	14.38639	-56.10474	-0.240795	0.001287565
10^3	9.832573	8.059372	0.02398896	0.0008929589
10^4	10.24743	0.005214	0.00001599	0.0009200699
10^5	9.629958	6.559691	0.01897116	0.0008676243
10^6	9.992452	0.488579	0.00146728	0.0009009514
10^7	9.989312	0.556357	0.00167117	0.0009011318

for N becomes closer to 10 and that for p with f fixed, closer to 0.001.

The estimates for f (in the third column) however, are turbulent but begin to stabilise for a sample size of 10^6 and above. Even with a large sample size, the estimate for f is not accurate. The estimates of p in the fourth column of Table 1, improve as the sample size increases but are not quite accurate due to the inaccuracies of the estimated f on which they depend. For small sample sizes (100), the precision of N is very poor and this has drastic effects on the precision of f and p that are calculated subsequently.

The results show that the precision of the estimates increases as the sample size increases. Even though estimates for f improve as sample size increases, at large samples sizes, the estimates are still not satisfactory and cannot be relied upon.

3.4.1 Strengths and Weaknesses of the Method of Moments

Some of the strengths are that; the method of moments is easy to derive and implement, and for a fixed value of f, the method is capable of producing accurate estimates of N and p. This is feasible because in practice, the person conducting the experiment should have an idea about what f may be, since it is the dilution applied before sequencing. For a small data sample, this method is fast.

The downside of the method of moments is that it is not able to generate accurate estimates for f. Also, any small errors in the estimation of N can cause big errors in the estimates for f and p. Additionally, the method of moments requires a sufficiently large number of data points to produce accurate results. Practically this is not feasible due to the costs involved in generating sequence reads.

3.5 BAYESIAN INFERENCE FOR ESTIMATING PARAMETERS N, f AND p

The shortcomings of the method of moments discussed in the first part of this chapter call for alternative methods for estimating the parameters of PCR amplification. In the second part of this chapter, the reader is introduced to the Bayesian realm of inference and examples are given. We describe a Bayesian model for estimating amplification parameters N, p and f, which entails a Gibbs sampling algorithm and full conditionals. Lastly, we apply our Bayesian method to data samples with different sizes, discuss about the diagnostics used (i.e. autocorrelation), and the advantages of this new method over the one described in Chapter Three.

3.6 Introduction to Bayesian Inference

Bayesian inference is one of the two dominant methods of statistical inference, that is, together with frequentist inference. The two approaches differ in that Bayesians use the term probability to describe all unknown quantities whereas frequentists limit the application of the same term to summaries of hypothetical replicate data sets. The result is that the Bayesian usage of the term "probability" appears to be more consonant with the informal use of the concept than the restrictive sense required by the frequentists. This virtue makes Bayesian inference more intelligible than corresponding frequentist statistics.

3.6.1 Bayes Theorem

Bayesian inference is based on Bayes' theorem; a simple relation between two conditional probabilities that are the reverse of each other. Bayes' thereom is named after Reverend Thomas Bayes (1702-1761) who proposed

it even though Stigler (1983) reports about an earlier incidence of the theorem before Thomas Bayes. It expresses the conditional probability of an event A after B is observed (denoted B|A), in terms of the probability of event A, probability of event B, and the conditional probability of B given A. That is;

$$Pr(A|B) = \frac{Pr(B|A)Pr(A)}{Pr(B)}. \tag{3.33}$$

Instead of a single event A, suppose we have a set $A_j, j = 1, 2, ..., k$ of mutually exclusive and exhaustive events (only one of them can occur at one particular time, and one or the other is bound to happen). Since events A_j are mutually exclusive, it follows from the laws of probability that;

$$Pr(B) = \sum_{j=1}^{k} Pr(B, A_j) = \sum_{j=1}^{k} Pr(B|A_j)Pr(A_j). \tag{3.34}$$

Thus for a specific event A_i, Bayes' theorem becomes:

$$Pr(A_i|B) = \frac{Pr(B|A_i)Pr(A_i)}{\sum_{j=1}^{k} Pr(B|A_j)Pr(A_j)}. \tag{3.35}$$

For clarity in modelling, we will use the bracket notation, that is, $[A, B]$ for the joint distribution of A and B, $[A|B]$ for the conditional distribution for A given B and $[B]$ for marginal distribution of B. Using this notation, Eq. (3.33) can be written as;

$$[A|B] = \frac{[B|A][A]}{[B]}. \tag{3.36}$$

3.6.2 Likelihood function

The likelihood function is a function of parameters of a statistical model (Wikipedia, 2015d). It is very useful in methods of estimating a pa-

rameter from a set of statistics, such as in the Bayesian approach presented in this chapter. The term Likelihood is usually used informally in reference to probability but a distinction is made in statistical usage. Probability is used when describing a function of the outcome given a fixed parameter value. For example when a fair coin is tossed 10 times, what is the probability of obtaining heads all the time? The likelihood on the other hand is used when describing a function of a parameter given an outcome. For example, if in the aforementioned experiment of 10 trials, 2 heads were obtained, what is the likelihood that the coin is fair?

The likelihood of a set of parameter values, $\theta = (\theta_1, \theta_2, ..., \theta_m)$, given outcomes $X = (x_1, x_2, ..., x_n)$, is equal to the probability of those observed outcomes given those parameter values, that is;

$$\mathcal{L}(\theta|X) = Pr(X|\theta) \tag{3.37}$$

For Discrete probability distributions, which we will predominantly use in this work, the likelihood is defined as a function of θ given by:

$$\mathcal{L}(\theta|X) = Pr_\theta(X = x), \tag{3.38}$$

where X is a discrete random variable. We note that the value on the right hand side of Eq. (3.37) is not a conditional probability since θ is not a random variable. As such, we choose to write it instead as $Pr_\theta(X = x)$.

Often, the natural logarithm of the likelihood function, called the log-likelihood, is more convenient to work with. Since the logarithm is a monotonically increasing function, the logarithm of a function acquires its maximum value at the same points as the function itself. As a result, the log-likelihood can be used in place of the likelihood in maximum likelihood estimation and in the case of other related estimations. Also, finding the maximum of a function often involves taking the derivative of a function

and solving for the parameter being maximized which is always easier for the log-transformed function than for the original function itself. Consider observations from Binomial distribution, $X \sim B(n, p)$. The likelihood function is given by;

$$\mathcal{L}(p|X) \propto p^x(1-p)^{n-x}. \tag{3.39}$$

The log-likelihood is given by;

$$l(p) = \log(\mathcal{L}(p|X)) \propto x\log(p) + (n-x)\log(1-p). \tag{3.40}$$

By setting

$$l'(p) = \frac{x}{p} - \frac{n-x}{1-p} = 0, \tag{3.41}$$

we obtain the maximum likelihood estimator as;

$$\hat{p} = x/n,$$

which is the same as the one obtained with the original likelihood function.

3.6.3 Basics of Bayesian Inference

For model-based Bayesian inference, B in Eq. (3.36) is replaced with observations X and A, with parameter set $\theta = (\theta_1, \theta_2, ..., \theta_n)$. Bayes' theorem obtained thus becomes:

$$[\theta|X] = \frac{[X|\theta][\theta]}{[X]}, \tag{3.42}$$

where

$$[X] = \int [X|\theta][\theta]d\theta \tag{3.43}$$

is the marginal distribution of the data, $[\theta|X]$ is the joint posterior distribution, $[X|\theta]$ is the likelihood and $[\theta]$, the joint prior distribution. The prior distribution can be thought of as a summary of all that is known about the parameter of interest, before observing the data. The posterior distribution summarises all that is known about the parameter, combining prior knowledge and information provided by the data. If one chooses a prior which expresses dead certainty about the value of the parameter, then the data will be ignored (subjective inference). However, if the prior expresses uncertainty about the parameter, then as the sample size increases, the data will prevail in guiding inference (objective inference). All Bayesian inference is based on the posterior distribution. For example, if a point estimate is desired, one usually uses the posterior mean and when an interval estimate is desired, one employs the percentiles of the posterior distribution.

The two distributions, that is, the prior and the posterior, together with the likelihood, are the primary features of Bayesian analysis. The marginal distribution in Eq. (3.43) is sometimes used for model checking but need not be computed in characterising the posterior distribution of θ. As such, we may write Eq. (3.42) as;

$$[\theta|X] \propto [X|\theta][\theta]. \tag{3.44}$$

Since $[\theta|X]$ is a probability distribution, it integrates to 1 with respect to θ and the constant of proportionality is $1/[X]$.

In defining the posterior distribution, the right hand side of Eq. (3.44) treats $[X|\theta]$ as a function of θ with X fixed, that is, a likelihood function. According to Eq. (3.44), the posterior distribution is proportional to the product of the likelihood and the prior. Thus, the basis of inference is the product of information provided by the data, and by the prior, that is, the posterior $[\theta|X]$ expresses uncertainty about θ after taking the prior and data into account.

3.7 Calculating Posterior Distributions

When defining the posterior distribution, the normalizing constant is irrelevant. However if we wish to report the mean, its variance, its percentiles and maybe to know specific probabilities, such as $Pr(\theta > 0|X)$, it maybe necessary to evaluate the normalising constant in Eq. (3.36). Unfortunately, evaluating this integral is almost always difficult and frequently impossible. It is not exaggerating by saying that this computation has been the primary drawback to implementation of Bayesian methods. A couple of methods for overcoming this difficulty, in order to evaluate the posterior distribution, are discussed in this section.

3.7.1 Conjugacy

One of the simplest ways of going around evaluation of Eq. (3.36) is to identify a family of distributions, that includes both the prior and posterior distributions. In these special cases, the prior combines with the likelihood to produce a posterior distribution of similar form to the prior: the prior is said to be conjugate to the likelihood. Distributions in this "conjugate family" are identified by a hyperparameter ψ. We can summarize what was known before data collection by a particular value ψ_0, and what is known afterwards, by a new value ψ_1. The process of updating knowledge by data zeros down to updating ψ, and the transition from ψ_0 to ψ_1 involves simple summaries of the data. The existence of a conjugate family depends on the form of the likelihood function. Conjugacy completely solves the computational problem, but there are relatively few cases where it applies.

Example 3.7.1

Consider the choice of a beta distribution as a prior for the binomial success

parameter p. In this case, the likelihood is of a binomial form, given by:

$$[X|p] = \binom{n}{x} p^x (1-p)^{10-x}. \tag{3.45}$$

The beta prior for p on the other hand is given by:

$$Be(p, a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1}, 0 < p < 1 \text{ and } a, b > 0. \tag{3.46}$$

From Eq. (3.44), the resulting posterior distribution is:

$$\begin{aligned} [p|X] &\propto Be(x, n, p)Be(p, a, b) \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1} \binom{n}{x} p^x (1-p)^{10-x} \\ &\propto p^{x+a-1}(1-p)^{n-x+b-1} \\ &\propto p^{A-1}(1-p)^{B-1}, \end{aligned} \tag{3.47}$$

where $A = x + a$ and $B = n - x + b$. All that is required now is a suitable proportionality constant to put Eq. (3.47) into the beta form described in Eq. (3.46). For this reason, the beta family of distributions is said to be conjugate for the binomial success parameter since both the prior and the posterior are in the same family of beta distributions. Also of great interest is the fact that the posterior from one study can be used as the prior for the next study. In this example, we can think of a and b as running totals of previous numbers of success and failure. Hence a $Be(10, 14)$ can be thought of as roughly equivalent to knowledge acquired from a previous experience of 10 success and 14 failures in 24 previous trials.

Next, we consider an example involving a normal likelihood to illustrate the additional complexities of posterior analysis associated with multivariate parameters. When inference of more than one parameter is required from the same data, a joint posterior distribution is used. Sometimes the multivariate characteristics of this joint distribution are of interest but most

often inference focuses on one parameter at a time. This is the case especially when nuisance parameters are involved. These are parameters which are of no direct interest but which are necessary in order to correctly describe the sampling model.

For a vector-valued parameter $\theta = (\theta_1, \theta_2, ..., \theta_n)'$, Eq. (3.44) can be written as;

$$[\theta_1, \theta_2, ..., \theta_n|X] \propto [X|\theta_1, \theta_2, ..., \theta_n][\theta_1, \theta_2, ..., \theta_n]. \tag{3.48}$$

Mostly but not always, the priors for each parameter are independent. To make inference about a particular parameter, we integrate the rest of the parameters from the joint posterior distribution as shown below for one case of parameter θ_1;

$$[\theta_1|X] = \int ... \int [\theta_1, \theta_2, ..., \theta_n|X] d\theta_2 ... d\theta_n. \tag{3.49}$$

This integral is usually intractable. As such, inference of a single parameter in a multi-parameter setting involves two potential obstacles: first, of obtaining the joint posterior, and second, of obtaining the marginal distribution from the posterior. For the cases where conjugacy is possible, the first task of obtaining the posterior distribution is made easier.

Example 3.7.2

Consider n observations $X = (X_1, X_2, ..., X_n)$ that are normally distributed with $X_i \sim N(\mu, \sigma^2)$, and that we wish to infer μ, with σ unknown. The parameter set is $\theta = (\mu, \tau)$ where $\tau = 1/\sigma^2$ (precision) is the nuisance parameter. Inference for μ will be based on the marginal distribution of μ;

$$[\mu|X] = \int [(\mu, \tau|X)] d\tau. \tag{3.50}$$

The conjugate family in this problem is defined in terms of a gamma dis-

tribution for τ and a normal distribution for μ given τ, that is, the prior;

$$[\theta] = [\mu|\tau][\tau], \tag{3.51}$$

where $[\mu|\tau] = N(\eta, 1/(\kappa\tau))$ and $[\tau] = Ga(\alpha, \beta)$. The set of hyperparameters in this case will be $\psi = (\alpha, \beta, \eta, \kappa)$. Let the prior be described by the set of hyperparameters $\psi_0 = (\alpha_0, \beta_0, \eta_0, \kappa_0)$. The joint posterior distribution in this case will be:

$$\begin{aligned}[\mu, \tau|X] &= \tau^{\frac{n}{2}} \exp\left[-\frac{\tau}{2}\sum_{i=1}^{n}(X_i - \mu)^2\right] \times (\kappa_0\tau)^{\frac{1}{2}} \times \\ &\quad exp\left[\frac{-\kappa_0\tau}{2}(\mu - \eta_0)^2\right] \tau^{\alpha_0 - 1} \exp(-\beta_0\tau) \\ &= (\kappa_1\tau)^{\frac{1}{2}} \exp\left[-\frac{k_1\tau}{2}(\mu - \eta_1)^2\right] \times \tau^{\alpha_1 - 1} \exp(-\beta_1\tau),\end{aligned}$$

where $\alpha_1 = \alpha_0 + n/2$, $\kappa_1 = \kappa_0 + n$,

$$\eta_1 = \frac{n}{n + \kappa_0}\bar{x} + \frac{\kappa_0}{n + \kappa_0}\eta_0 \text{ and} \tag{3.52}$$

$$\beta_1 = \beta_0 + \frac{(n-1)S^2}{2} + \frac{n\kappa_0(\bar{x} - \eta_0)^2}{2(n + \kappa_0)}. \tag{3.53}$$

Therefore the posterior is of the same family (normal-Gamma) as the prior and hence the normal-gamma family of distributions is conjugate for the normal likelihood. Consequently, the required marginal distribution can be obtained by integrating (in this case) or by sampling. First we sample τ from the $Ga(\alpha_1, \beta_1)$ and then use the sampled τ to draw μ from $N(\eta_1, 1/(\kappa_1\tau))$.

When simple solutions based on conjugacy are not an option, most Bayesian applications examine posterior distributions by random sampling. By sampling θ from $[\theta|X]$, most features of the sample such as sample mean, sample proportion of values $\theta > 0$, can be used to estimate corresponding summaries of the posterior distribution; in this case, the posterior mean value of θ and $Pr(\theta > 0|X)$ respectively. The estimates obtained usually

can be made as accurate as possible by drawing a large size of samples. Simulation methods for studying probability distributions are generally termed Monte Carlo methods.

3.7.2 Monte Carlo methods

Monte Carlo methods typically involve draws of independent samples from distributions being studied. The basic idea is that we can study features of a probability distribution G by examining corresponding features of a sample $X_1, X_2, ..., X_n$ from G. For example, suppose we wish to learn about the ratio of the largest lifespan to the average lifespan in samples of size 40. Assuming exponential lifespans, we would attempt to do so analytically based on the form of the exponential distribution but the calculations would be very difficult. An alternative would be to randomly generate samples of size 40 and obtain the ratio in each case. Doing this 10 million times and summarizing the results takes less than 8 seconds on a 3.2 GHz laptop. It also guarantees two decimal place accuracy in the mean (3.82) and the standard deviation (0.99) (Link & Barker, 2009). Due to independence of the samples, evaluation of the precision of summaries obtained using ordinary Monte Carlo methods is made simple.

A more straight forward Monte Carlo approach is through inversion of cumulative distribution functions, though for posterior distributions, this requires evaluation of the integral in *Eq.* (3.43). The other approach, which avoids this requirement, is rejection sampling.

Drawing independent samples from the posterior distribution is usually not straight forward even when using rejection sampling or related techniques. However, Bayesian statistics has been revolutionised by the development of techniques for drawing dependent samples from the posterior distribution, which can be used in a similar manner as the independent samples. These methods are collectively known as Markov Chain Monte

Carlo and are described in the next section.

3.7.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a simulation technique for examining probability distributions. Although the basic techniques of MCMC were first developed in the early 1950's, they were paid scant attention among statisticians until the late 1980's, with the publication of important applications to image processing by Geman & Geman (1984) and Besag (1986). Ever since then, MCMC has taken the world by the storm due to its usefulness and relatively easy implementation. It has revolutionized data analysis, breaking down the largest barrier to Bayesian analysis, that of computation.

Markov chain and stationary distributions

The $k-$th order Markov chain is a sequence $X_1, X_2, X_3, ...,$ with the property that, given all of the previous values, the probability distribution of the next value depends only on the last k values. That is;

$$[X_t|X_{t-1}, X_{t-2}, ..., X_1] = [X_t|X_{t-1}, X_{t-2}, ..., X_{t-k}]. \tag{3.54}$$

Many Markov chains describe natural processes evolving through time, hence the index t is often referred to as "time," and the values of X_t are referred to as "states" (of nature). From the definition in Eq. (3.54) above, the first order Markov chain is the one in which the probability of an outcome in any trial depends at most upon the outcome of the immediately preceding trial, that is;

$$[X_t|X_{t-1}, X_{t-2}, ..., X_1] = [X_t|X_{t-1}]. \tag{3.55}$$

The probability on the right had side of 3.55 above represents the transitional probability from state X_{t-1} to X_t.

Example 3.7.3

Consider an example extracted from Link & Barker (2009): A gambler enters a casino with a stake of X_1 dollars in his pocket and makes a series of pound bets on the roulette. Typically, the roulette wheel has 38 equally likely independent outcomes. Lets consider the case when 18 of the outcomes win the gambler a pound and 20 outcomes loose a pound. The gambler's stake after t gambles is X_t. Since the gambles on a roulette are independent, then X_t, $t = 1, 2, ...$ is a first−order Markov chain. X_t is either $X_{t-1} + 1$ (with probability $p = 18/38$) or $X_{t-1} - 1$ (with probability $1 - p = 20/38$) regardless of the preceding history, that is, $X_1, X_2, X_3, ..., X_{t-2}$.

Suppose the gambler's initial stake was 20 pounds and he is willing to gamble until he either doubles this amount or losses it all. In this particular case, his stake will always be in the set $S = \{0, 1, 2, ..., 40\}$, also called the state space. States 0 and 40 are called absorbing states because it is impossible to leave these states, that is, if $X_t = 0$, then $X_{t+k} = 0$ for $k = 1, 2, 3, ...$ and also when $X_t = 40$, then $X_{t+k} = 40$ for $k = 1, 2, 3, ...$

If after $t - 1$ bets, the gambler's stake is at $X_{t-1} = \$5$, then $X_t = \$4$ with probability $20/38$ or $X_t = \$6$ with probability $18/28$. Previous knowledge of say, 20 successive losses from \$25 to \$ 5 at $t - 1$ will provide no insights as to whether he will win or loose in the next bet. His stake at time t depends on $(X_1, X_2, X_3, ..., X_{t-1})$ only through X_{t-1} hence the sequence X_t is a first order Markov chain.

Definition 3.7.1 (**Stationary distribution**)

A stationary distribution is a probability distribution that satisfies

$$\pi(A) = Pr(X_t \in A), \tag{3.56}$$

for each subset A of the sample space. In this case, the probability that X_t is in a particular state or set of states does not depend on t.

Stationary distributions of Markov chains are the basis of MCMC. However, not all Markov Chains are stationary, the example of the gambler being a case in point. Considering $Pr(X_t = 19)$, at $t = 2$, the only possible states are $X_2 = X_1 - 1 = 19$ or $X_2 = X_1 + 1 = 21$, depending on whether he won his first gamble, thus $Pr(X_2 = 19) = 20/38$. With time, his chain will either reach 0 or 40, and remain there, so that $Pr(X_t = 19)$ approaches zero as t gets large. The existence of a stationary distribution would require that $Pr(X_t = 19)$ not change through time.

In cases where Cdf inversion and rejection sampling are not feasible, we can sample the posterior distribution by constructing a Markov chain X_t with a stationary distribution. The sampled values would not be independent, but atleast would be samples from the distribution we wish to investigate. Before we use MCMC, we need to be mindful of the ergodicity theorem.

Theorem 3.7.1 (**Ergodicity**)

A positive recurrent and aperiodic Markov chain has a stationary distribution $\pi(A)$ that satisfies

$$\pi(A) = \lim_{n\to\infty} Pr(X_n \in A|X_1), \tag{3.57}$$

for subsets A of the sample space.

Firstly, we define the terms aperiodic and recurrent.

Definition 3.7.2 (**Aperiodic state**)

A state i is aperiodic if returns to state i can occur at irregular times. In other words, there exists n such that for all $n' \geqslant n$,

$$Pr(X_{n'} = i|X_0 = i) > 0. \tag{3.58}$$

A Markov chain is said to be aperiodic if each of its states is aperiodic.

Definition 3.7.3 (**Recurrent state**)

A state i is recurrent if it is guaranteed (with probability =1) to have a finite hitting time (the first return time to state i). Let the random variable T_i be the hitting time, then:

$$T_i = \inf \{n \geqslant 1 : X_n = i | X_o = i\}. \tag{3.59}$$

The number

$$f_{ii}^{(n)} = Pr(T_i = n) \tag{3.60}$$

is the probability that we return to state i for the first time after n steps. State i is therefore recurrent if

$$Pr(T_i < \infty) = \sum_{n=1}^{\infty} f_{ii}^{(n)} = 1. \tag{3.61}$$

Otherwise, state i is said to be transient. The mean recurrence time at state i is the expected return time M_i, that is:

$$M_i = E(T_i) = \sum_{n=1}^{\infty} n f_{ii}^{(n)}. \tag{3.62}$$

State i is positive recurrent if M_i is finite. A Markov chain is positive recurrent if each of its states is positive recurrent.

The consequence of the ergodicity theorem is that it guarantees the existence of a stationary distribution and also states that the starting value X_1 does not affect the asymptotic behaviour of the chain. That is, regardless of the starting value of the chain, it eventually settles into a pattern of visiting A with specified probability, $\pi(A)$. This is a useful observation for implementation of MCMC; in practice, we must specify starting values. To

compensate for arbitrariness of the starting value, in practice, we usually discard some of the early values, which are not representative of the stationary distribution. These are referred to as "burn-in values" in Link & Barker (2009) .

As we stated above, the Markov chain of the example about the gambler is not stationary. First of all, this chain is not aperiodic because we can only return to a state in a number of steps that are multiples of 2. The chain is therefore periodic with period 2. Secondly the same chain is not recurrent because of the existence of absorbing states 0 and 40. From any state $0 < i < 40$, there is always a chance of reaching the absorbing state before returning to state i. So $Pr(T_i < \infty) < 1$ rather than $Pr(T_i < \infty) = 1$ as in Eq. (3.61). The ergodicity theorem therefore does not apply.

Next we present the two most popular MCMC methods i.e Metropolis Hastings algorithm and Gibbs sampling.

Metropolis Hastings Algorithm

The metropolis Hasting (MH) algorithm is one of the most popular MCMC methods for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult. Suppose that we wish to draw samples from target distribution $f(x)$. Let $j(x|y)$ be candidate generating distributions, describing probabilities for candidate values x, given current value y. We fix a value X_0 and then for $t = 1, 2, ...$, generate X_t according to the following rules:

1. Generate a candidate value, X_{cand} by sampling from $j(x|X_{t-1})$.

2. Calculate

$$r = \frac{f(X_{cand})j(X_{t-1}|X_{cand})}{f(X_{t-1})j(X_{cand}|X_{t-1})}. \tag{3.63}$$

3. Generate $U \sim U(0, 1)$.

4. If $U < r$, set $X_t = X_{cand}$, otherwise set $X_t = X_{t-1}$.

One important observation is that the target distribution is only involved in calculating r in Eq. (3.63), and this occurs in both the denominator and numerator. This means that the normalising constant which was a major stumbling block has been overcome.

From above, we notice that the MH algorithm has a wide allowance in selection of the candidate generating function. The main limitations on $j(x|y)$ is the requirement that Markov chains be positive recurrent in order to have stationary distributions: every state must be reachable from every other state. In addition to this minimal requirement, practically the chain is also required to move freely enough, to have reasonably low autocorrelation.

When the candidate generating function is symmetric such that $j(x|y) = j(y|x)$, r in step 2 becomes

$$r = \frac{f(X_{cand})}{f(X_{t-1})}, \tag{3.64}$$

which simplifies calculations and saves computation time. This special case is known as the symmetrical Metropolis Hastings. Consider an example in which the MH algorithm is used to generate samples of a uniform distribution.

Example 3.7.4

In this example, we will be generating samples of the standard normal distribution using Markov Chain Monte Carlo. In order to obtain a Markov chain, we define a tuning parameter, $A > 0$ which can be of any value although some values are better than others.

We let $X_0 = 0$ and then generate X_t according to the Metropolis Hastings steps afore mentioned, that is;

1. Generate two independent $U(0, 1)$ random variables, say u_1 and u_2.

2. Calculate a candidate value, $X_c = X_{t-1} + 2A(u_1 - 1/2)$.

3. Obtain

$$r = \frac{\exp\left(-\frac{1}{2}X_{cand}^2\right)}{\exp\left(-\frac{1}{2}X_{t-1}^2\right)}. \tag{3.65}$$

4. If $u_2 < r$, set $X_t = X_{cand}$, otherwise set $X_t = X_{t-1}$.

During a single run of the algorithm, the chain either remains at its current value or moves incrementally to a randomly generated candidate value found in the neighbourhood of the current value. The change in the second step has a $U(-A, A)$ distribution, so the candidate value is sampled uniformly over an interval centred at the current value, that is, $X_{cand}|X_{t-1} \sim U(X_{t-1} - A, X_{t-1} + A)$. We note that step 4 involves a Bernoulli trial, with success probability $= min(r, 1)$. This success parameter is referred to as the acceptance or movement probability.

Effect of the tuning parameter and starting value

Using different values of A (step sizes), the Metropolis algorithm explained in the example above was used to produce different samples of the standard normal distribution and the results for each step size are presented below. The history plot for Markov chain X_t is obtained by plotting X_t against t in Figure 3.

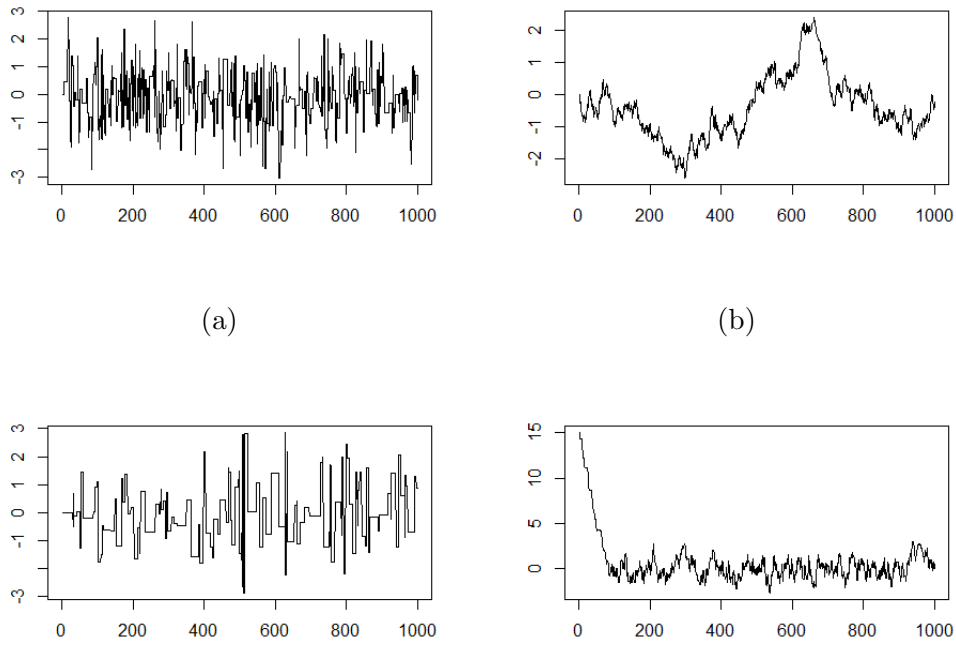


Figure 3: Plot of X_t against t for Markov chains with a standard normal distribution. The chains in a, b c and d were obtained using step sizes of $3.7, 0.2, 13$ and 1 respectively. See Appendix B for the code.

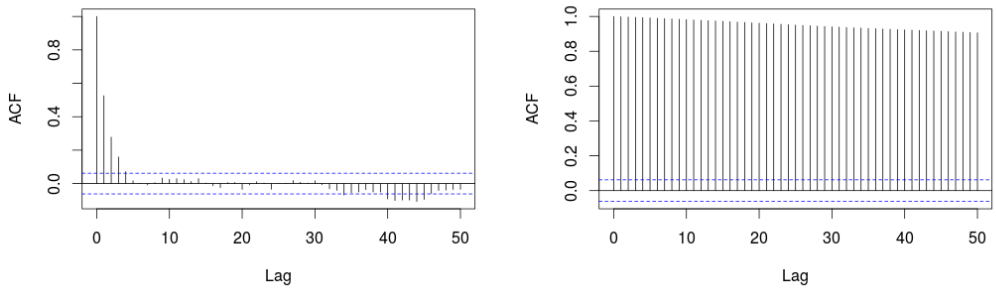
Each of the four chains has a standard normal stationary distribution, but it is clear that their plots are different. The chains with $A = 0.2$ and 1.0 (Figure 3b and 3d) take many small steps and move slowly over the range of the standard normal distribution. The chain with $A = 1$ and $X_0 = 15$ (3d) moves slowly but expectedly into the region of acceptable values of the standard normal distribution. Its first 100 values are not representative of the standard normal distribution and as a result, can be discarded as "burn in" values. Chains with $A = 3.7$ (Figure 3a) and the other with $A = 13$ (Figure 3c), take reasonably large steps over the range of values but the latter occasionally stalls.

Diagnostics of Markov Chain Monte Carlo

MCMC methods, like any other numerical method, are liable to failure. Unfortunately, it is highly probable that some of these failures go unnoticed which definitely affects the precision of the obtained estimate. The most basic way of noticing failures in MCMC implementations is by observing the history plot and ensuring that it is "grassy" like the one in Figure 3a. This however does not guarantee the absence of failures in the output. On the other hand, if the plot is not grassy as in Figure 3b, then something is wrong. It may be that the Markov chain was not properly tuned or that poor starting values of the parameter were used. A solution is to check as to whether the model has been correctly specified and there are no redundant parameters. It is also always a good idea to produce long chains if possible since features of the chains approximate more closely the features of the posterior distribution as the length of the chain increases.

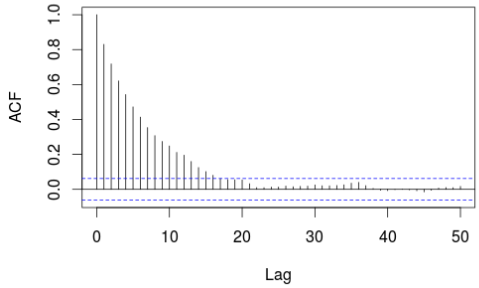
A more reliable means of determining the precision of the MCMC method in generating samples of the posterior is by observing the autocorrelation function. For more precise estimates, we want the generated samples to be as independent from each other as possible. The autocorrelation function is a measure of the strength of association among values of X_t. For $h = 1, 2, ...$, the correlation $\rho(X_t, X_{t+h})$ between X_t and X_{t+h} is called the autocorrelation at *lag* h, and $R(h) = \rho(X_t, X_{t+h})$ is called the autocorrelation function (ACF). If the ACF tails off quickly enough that observations at *lag* k can be confidently regarded as independent, then we can thin the chain of N samples $\{X_t\}$ by picking only the N/k th samples.

In light of these diagnostic methods, the autocorrelation functions for each of the four chains in Figure 3 were obtained and are shown in Figure 4 below.

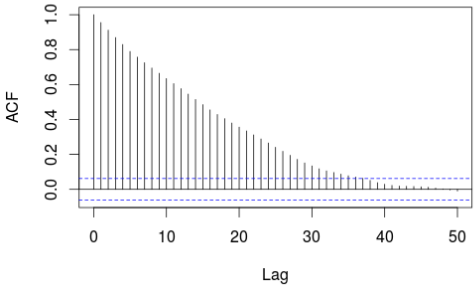


(a) step size = 3.7

(b) step size = 0.2



(c) step size = 13



(d) step size = 1

Figure 4: Corresponding autocorrelation functions for Markov chains with standard normal stationary distribution shown in Figure 3. From (a) to (b), the figures were obtained using step sizes of $3.7, 0.2, 13$ and 1 respectively.

For each of the step sizes, the autocorrelation plot agrees with the corresponding Markov chain history plot shown in Figure 3. Figure 4a has the least autocorrelation with dependence between samples persisting upto every other fourth value, that is any two sample values, with atleast four samples in between them can be considered as independent from each other. This agrees with the most grassy history plot in Figure 3a. The samples in Figure 3b were the most unstable with their history plot meandering far away from the expected value of zero. The corresponding autocorrelation plot in Figure 4b has the highest autocorrelation of the four plots in Figure 4, with a very high dependence between any two samples, even upto the $50th$ sample.

Generally in Figure 4, we observe that autocorrelation is high whenever the step size is much bigger or much smaller than 3.7. This is because

when the step size is small as in 4b, there are small increments in the candidate values and so r in 3.63 is close to 1 since nearby values have nearly identical probability. The acceptance rate is thus high and the result is a highly correlated chain that moves very slowly. Similarly, when the step size is large, candidate values will be far different from each other and hence $r << 1$. The result is lower acceptance probability which means that candidate values are maintained on the same value for many iterations and hence high autocorrelation.

Even though 4c does not have the least autocorrelation, independent samples can still be obtained form the Markov chain generated. In this case, $k \approx 20$. Better samples would be obtained by generating a chain of say 20,000 samples so as to thin it to 1000 by picking every other $20th$ value. We can then use the mean of this new sample to approximate the mean of the standard normal distribution. The variance of this sample mean is given by $\sigma^2/(N/k) = k\sigma^2/N$ with $\sigma^2 = Var(X_i)$, which provides a conservative measure of precision for the mean of the entire chain.

The Metropolis Hastings algorithm is only applicable in a single parameter setting. For multiple parameters, we introduce the Gibbs sampling technique in the next section.

3.7.4 Gibbs Sampling

For multivariate posterior distributions, a more suitable sampling technique is the Gibbs sampler. Let $\theta = (\theta_1, \theta_2, ..., \theta_n)$ be the parameter set, and X, the observation. The goal is to draw a sample from $[\theta|X]$.

Let θ_{-j} represent the set of parameters of length $n-1$, made up of all elements of θ, but excluding θ_j. The full conditional for θ_j is given by:

$$[\theta_j|\theta_{-j}, X], \tag{3.66}$$

a distribution of the jth component of θ, having fixed the values of all the other components, and having been informed by the data. It is proportional to $[X|\theta][\theta]$ just like the posterior $[\theta|X]$, the only difference being that the normalising constant now is $[\theta_{-j}, X]$ rather than $[X]$. As it turns out, full conditionals $[\theta_j|\theta_{-j}, X]$ are usually easily identified from $[X|\theta][\theta]$ by inspection when marginal distributions or joint posterior distributions are not. Next, we outline the steps of the Gibbs sampling algorithm.

Suppose that we wish to extract samples from a joint posterior distribution $[\theta|X]$. We fix a value $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_n^{(0)})$ and for $t = 1, 2, 3, ..,$ we generate $\theta^{(t)}$ according to the following rules:

- Step 1: Sample $\theta_1^{(t)}$ from the full conditional $[\theta_1|\theta_{-1}^{(t-1)}, X]$.
- Step 2: Sample $\theta_2^{(t)}$ from the full conditional $[\theta_2|\theta_{-2}^{(t-1)}, X]$.

$$\vdots$$

- Step n: Sample $\theta_n^{(t)}$ from the full conditional $[\theta_n|\theta_{-n}^{(t-1)}, X]$.
- Step n+1: Set $\theta^{(t)} = (\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_n^{(t)})$.

We note that one can also sequentially update $\theta^{(t)}$ after each step in the preceding algorithm, and use the partially updated $\theta^{(t)}$ in sampling subsequent full conditionals. For example, $\theta_4^{(t)}$ can be sampled from the full conditional distribution;

$$[\theta_4|\theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, \theta_5^{(t-1)}, ..., \theta_n^{(t-1)}, X] \tag{3.67}$$

instead of;

$$[\theta_4|\theta_1^{(t-1)}, \theta_2^{(t-1)}, \theta_3^{(t-1)}, \theta_5^{(t-1)}, ..., \theta_n^{(t-1)}, X]. \tag{3.68}$$

To illustrate Bayesian inference using MCMC, the parameters of a negative

binomial were estimated using pseudo random samples that were generated in R. This is presented in the next section.

Example 3.7.5

In an attempt to develop a Gibbs sampling algorithm, parameters N and p for the negative binomial distribution $A(x-N, N, p)$ given by:

$$p(x,t) = \binom{x-1}{x-N} p^N (1-p)^{x-N}. \tag{3.69}$$

were estimated. One thousand values were randomly generated in R (See the code in Appendix C) from the negative distribution, $A(x-10, 10, 0.01)$ and used for estimation. Since $p \in [0,1]$, an appropriate prior for it is the Beta distribution $Be(\alpha, \beta)$ and since N has count values, an appropriate prior for it is the Poisson distribution, $Po(\lambda)$. The posterior distribution was derived from the likelihood and priors as follows;

$$[p, N|X] = \binom{x-1}{x-N} p^N (1-p)^{x-N} \times \beta p^{\alpha-1}(1-p)^{\beta-1} \times \frac{\lambda^N \exp(-\lambda)}{N!}$$

$$\propto \binom{x-1}{x-N} p^{N+\alpha-1}(1-p)^{x-N+\beta-1} \times \frac{\lambda^N}{N!}. \tag{3.70}$$

For n data values $x_1, x_2, ...x_n$,

$$[p, N|X] \propto \left(\prod_{i=1}^{n} \binom{x_i-1}{x_i-N} p^N (1-p)^{x_i-N} \right) \times p^{\alpha-1}(1-p)^{\beta-1} \times \frac{\lambda^N}{N!}. \tag{3.71}$$

This joint posterior distribution or of any of the marginal posterior distributions are intractable because $[p, N|X]$ is a mess. Gibbs sampling, on the other hand, is fairly straightforward. However, before we can implement it, we need to identify full conditional distributions for p and N.

The full conditional for p is proportional to all of the terms in Eq. (3.71)

involving p and is therefore given by:

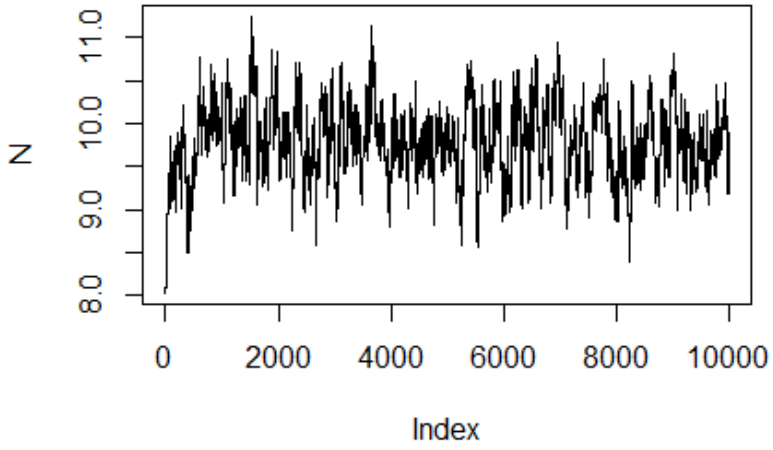
$$[p|N, X] \propto p^{Nn+\alpha-1}(1-p)^{\beta-1-nN+\sum x_i}.$$

Similarly, the full conditional for N is given by:

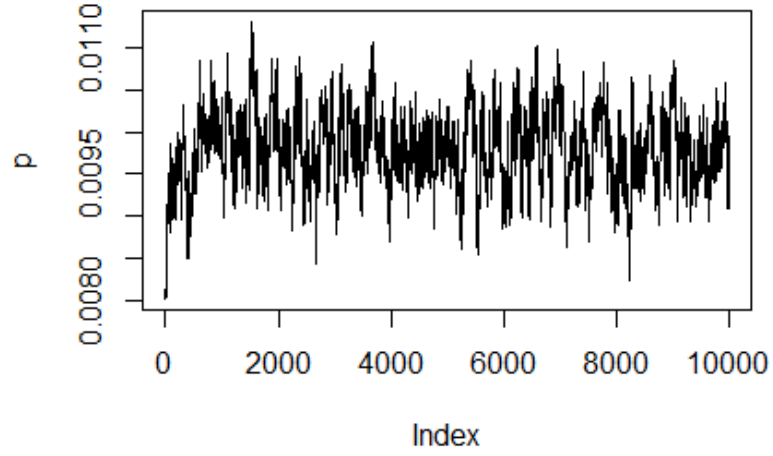
$$[N|p, X] \propto \frac{1}{(N-1)!^n \prod(x_i - N)!} \times p^{Nn}(1-p)^{-nN} \times \frac{\lambda^N}{N!}.$$

The prior and full conditional distribution for p are both in the Beta family of distributions while those for N are not. In this case, conjugacy worked for p but failed for N but our knowledge of conjugate forms has led to a choice of prior for which the full conditional distributions are easily identified. Also, due to the nature of the full conditionals of the two parameters, we will need a Metropolis Hastings algorithm to sample N unlike p.

Using the *distr* package in R, 1000 samples from the negative distribution with $N = 10$ and $p = 0.01$ were generated. A code (see Appendix C) was developed in R to implement Gibbs sampling in estimating back N and p. Using an optimal tuning parameter, $A = 0.036$ for the uniform proposal for N and initial values, 0.8 and 0.05 for N and p, the algorithm developed was run for 10000 iterations and produced the plots in Figure 5 for N and p.



(a)



(b)

Figure 5: History plot of N against t (Number of iterations) (a) and of p against t (b).

The optimal step size of 0.036 was obtained by calculating the lag 1 autocorrelation over a range of candidate values. The lag 1 autocorrelation as a function of A, that was obtained for N, is shown in Figure 6 below.

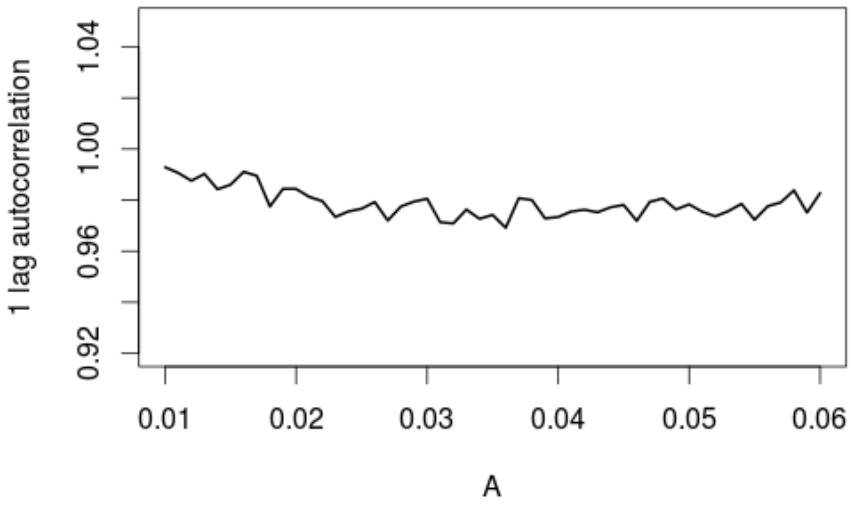
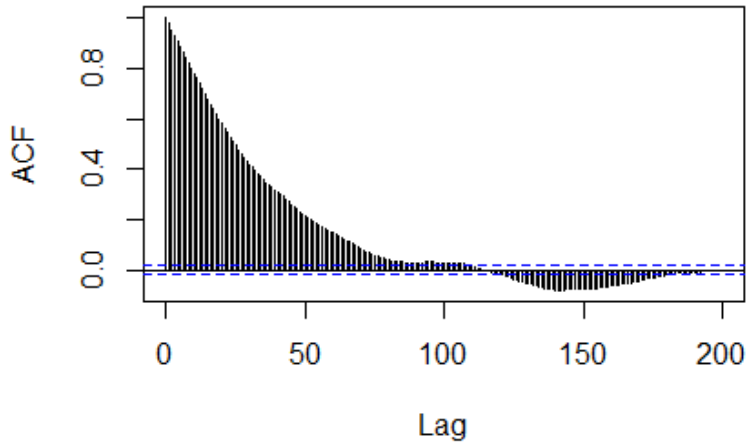
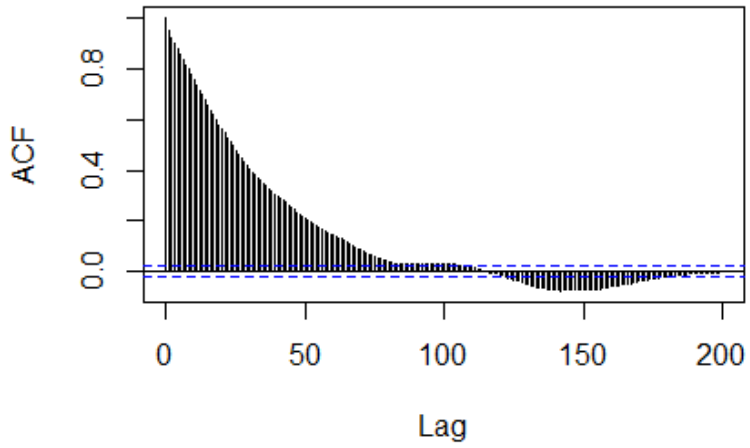


Figure 6: lag 1 autocorrelation as a function of the stepsize, A. The lowest autocorrelation for N was at $A = 0.036$

Practically, in order to rely on sample values for estimation, we have to be sure there is no dependence between them. As such, the samples obtained for each of the two parameters were investigated for independence by obtaining the autocorrelation between sample values. The autocorrelation functions obtained for N and p are shown in Figure 7.



(a) Autocorrelation plot for N

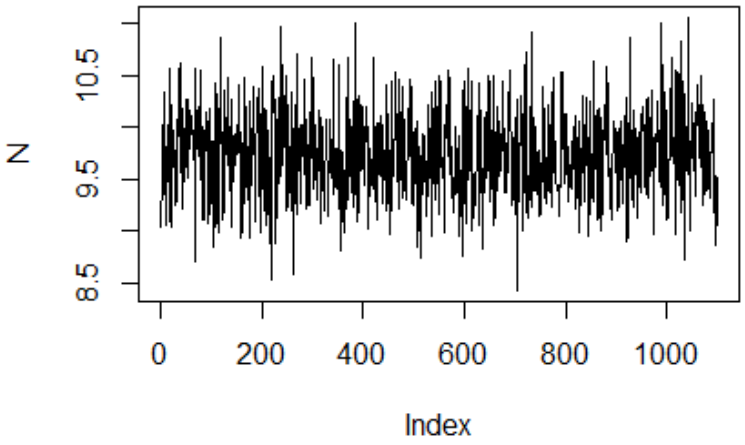


(b) Autocorrelation plot for p

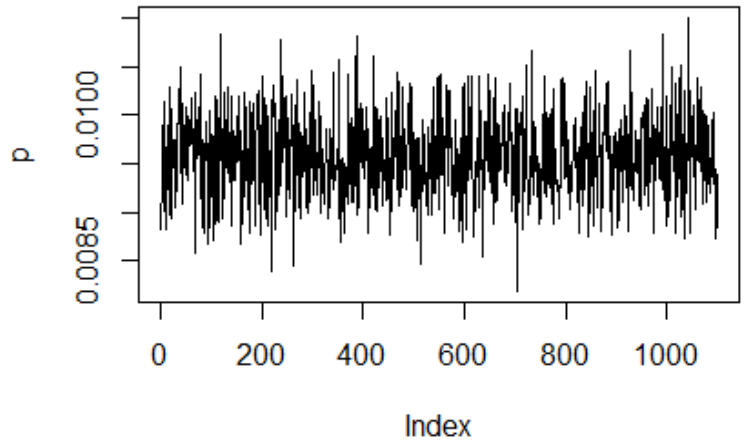
Figure 7: Full lag autocorrelation function for N and p.

From 7, it means that dependency between the samples is significant until when there is a difference of about 100 sample values between any two values. To obtain more accurate results, we will need to thin the samples drawn by picking every $100th$ value.

After thinning, the chains were more grassy as shown in Figure 8.



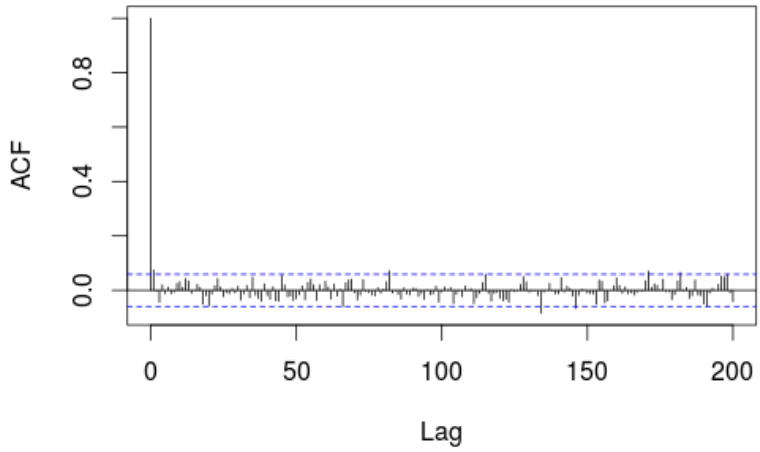
(a)



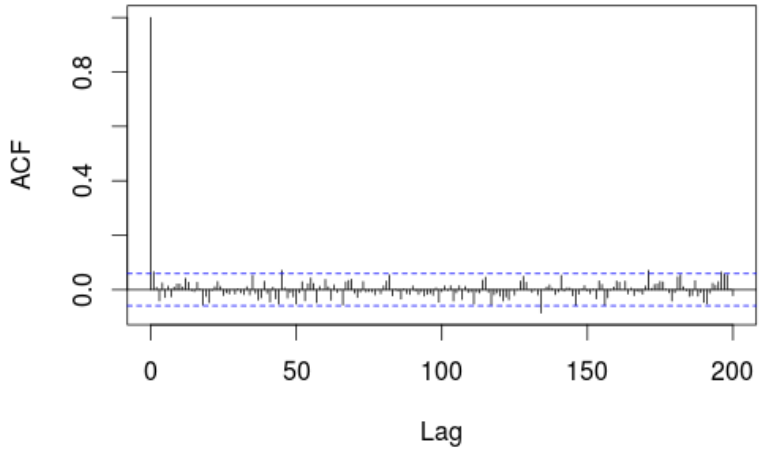
(b)

Figure 8: Thinned samples for N (a) and p (b).

The corresponding autocorrelation plots drop off zero immediately as shown in Figure 9 below. This shows that the samples left after thinning are independent from each other.



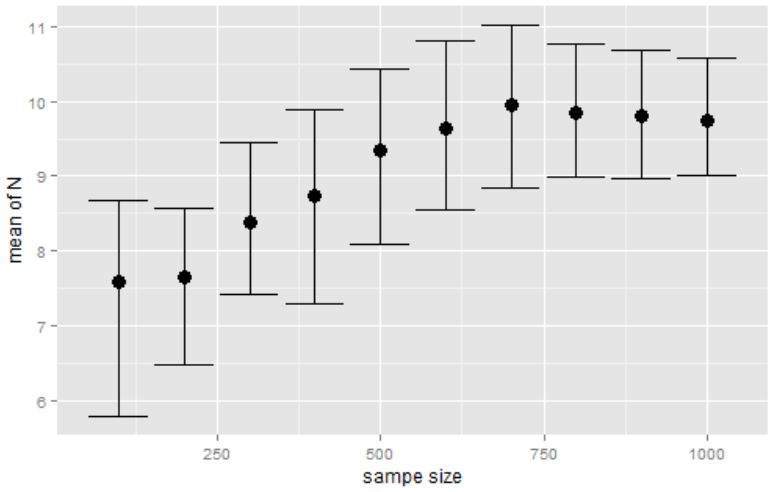
(a) Autocorrelation plot for N after thinning



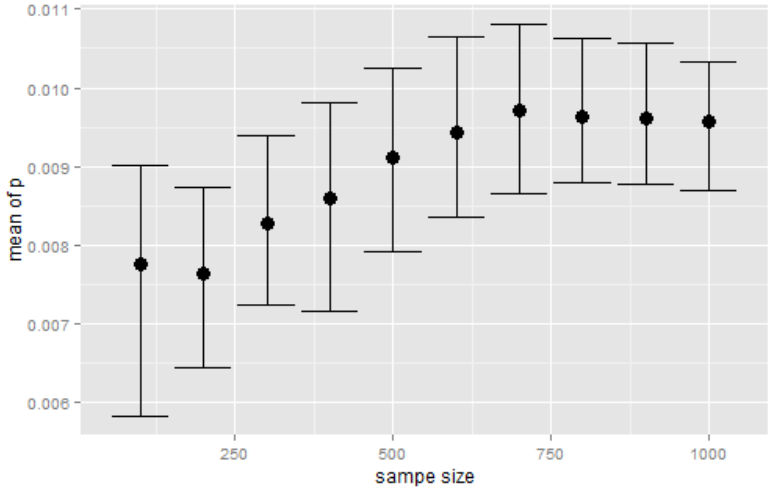
(b) Autocorrelation plot for p after thinning

Figure 9: Autocorrelation function of thinned samples for N and p.

The samples from the N and p distributions obtained with $A = 0.036$, were both thinned by picking each 100th value and the new samples obtained were used to calculate the 95% credible interval and the mean. For different sample sizes, the plots in Figure 10 were obtained for N and p.



(a)



(b)

Figure 10: Variation of mean and 95 % credible interval with increase in sample size for N (a) and p (b).

The mean and 95% credible interval of both parameters follow a similar trend. As the sample size increases, the length of 95% credible interval decreases which signifies a decrease in uncertainty. After a small decline, the sample mean increases as sample size increases up to a sample size of 700 samples, and it tapers slowly as we approach a sample size of 1000.

Having introduced the basics of Bayesian inference, we now describe a Bayesian model for estimating the parameters N, p, and f of PCR amplification from the probability distribution explained in the first part of this Chapter.

3.8 Bayesian Model for Estimating Parameters N, f and p of PCR Amplification

The probability distribution of the copy number of a particular clonotype after t cycles that was introduced in Eq. (3.23) as;

$$= \sum_{j=0}^{min(N,x)} \binom{N+x-j}{N-j,j,x-j} \frac{(1-p)^{x-j} f^x (1-f)^{N-j}}{(p+(1-p)f)^{N+x-j}} \times \frac{Np^N}{N+x-j} \tag{3.72}$$

Pulling terms independent of j out of the summation sign, we obtain;

$$p(x,t) = \frac{(1-p)^x f^x (1-f)^N N p^N}{(p+(1-p)f)^{N+x}} \sum_{j=0}^{min(N,x)} \binom{N+x-j}{N-j,j,x-j} \times \left(\frac{p+(1-p)f}{(1-p)(1-f)}\right)^j \frac{1}{N+x-j}.$$

Let;

$$A_j = \binom{N+x-j}{N-j,j,x-j} \frac{1}{N+x-j} \text{ and } B = \frac{p+(1-p)f}{(1-p)(1-f)}.$$

Then $p(x,t)$ becomes;

$$p(x,t) = \frac{(1-p)^x f^x (1-f)^N N p^N}{(p+(1-p)f)^{N+x}} \sum_{j=0}^{min(N,x)} A_j B^j.$$

For n values of x i.e $x_1, x_2, ...x_n$, the likelihood of the parameters is given by:

$$\mathcal{L}(N,f,p|X) = \prod_{i=1}^{n} \frac{(1-p)^{x_i} f^{x_i} (1-f)^N N p^N}{(p+(1-p)f)^{N+x_i}} \sum_{j=0}^{min(N,x_i)} A_{ij} B^j.$$

Due to the nature of the parameters, Beta priors are chosen for p and f and a Poisson prior is chosen for N. The joint posterior distribution for f, N

and p is given by:

$$[N, f, p|X] \propto p^{\alpha-1}(1-p)^{\beta-1} f^{\theta-1}(1-f)^{\gamma-1} \frac{\lambda^N}{N!} \times \prod_{i=1}^{n} \frac{(1-p)^{x_i} f^{x_i}(1-f)^N N p^N}{(p+(1-p)f)^{N+x_i}} \sum_{j=0}^{min(N,x_i)} A_{ij} B^j$$

Having obtained the posterior distribution function, we derive full conditionals for N, p and f.

The full conditional for N is a function proportional to all of the terms in the posterior involving N's and is therefore given by:

$$[N|f, p, X] \propto \frac{\lambda^N}{N!} \prod_{i=1}^{n} \frac{(1-f)^N N p^N}{(p+(1-p)f)^N} \sum_{j=0}^{min(N,x_i)} A_{ij} B^j.$$

Similarly the full conditional for p is a function proportional to all of the terms in the posterior involving p's and is therefore given by:

$$[p|N, f, X] \propto p^{\alpha-1}(1-p)^{\beta-1} \prod_{i=1}^{n} \frac{(1-p)^{x_i} p^N}{(p+(1-p)f)^{N+x_i}} \sum_{j=0}^{min(N,x_i)} A_{ij} B^j.$$

Lastly, the full conditional for f is proportional to all of the terms in the posterior involving f and is therefore given by:

$$[f|N, p, X] \propto f^{\theta-1}(1-f)^{\gamma-1} \prod_{i=1}^{n} \frac{f^{x_i}(1-f)^N}{(p+(1-p)f)^{N+x_i}} \sum_{j=0}^{min(N,x_i)} A_{ij} B^j.$$

Metropolis Hastings algorithms were implemented in R to sample from each of the above full conditionals. A log-normal proposal distribution was chosen for N while beta proposal functions were chosen for f and p. The developed Metropolis Hastings algorithms were combined into a Gibbs sampling algorithm by following the steps as shown in the second part of Section 3.7.4. That is, we use the partially updated parameters in sampling subsequent full conditionals.

We begun by specifying the probability distribution described by Eq. (3.72)

in R using the *distr* package from which we sampled random values for the final copy number x_i. For objective inference, non informative priors were chosen, that is, we set parameters $\alpha = \beta = \theta = \gamma = 1$ and replace the prior function for N with 1.

Using 1000 data points (sampled from the distribution in Eq. (3.72) with parameter values $N = 10$, $f = 0.3$ and $p = 0.001$), the developed Gibbs algorithm was run for 5000 iterations using tuning parameters AN= 0.09, Af= 0.02 and Ap= 5×10^5 for $N, f$and p respectively. Using $N = 1, f = 0.25$ and $p = 0.0017$, the code ran for $6.4 minutes$ and the following results were obtained for $N, f$and p.

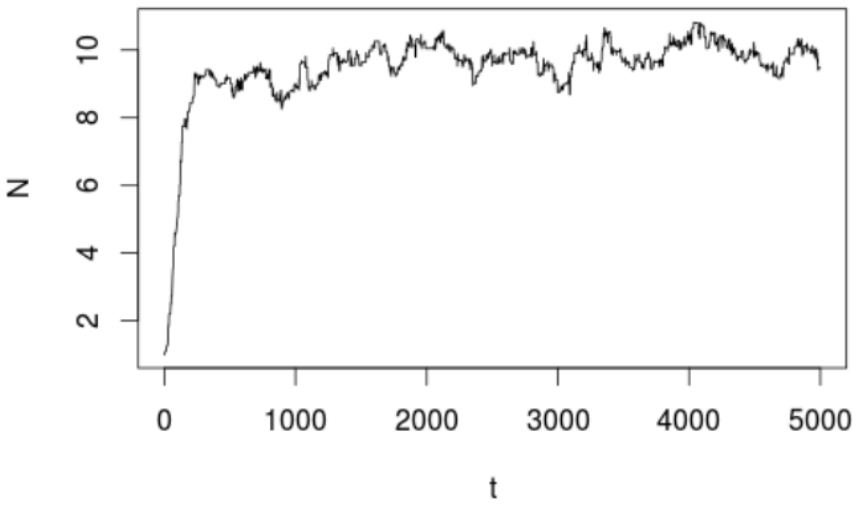


Figure 11: History plot for N

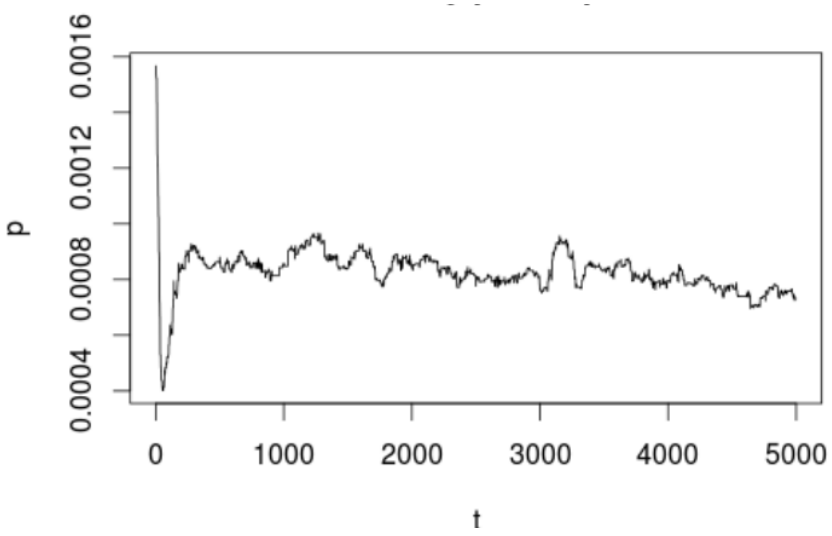


Figure 12: History plot for p

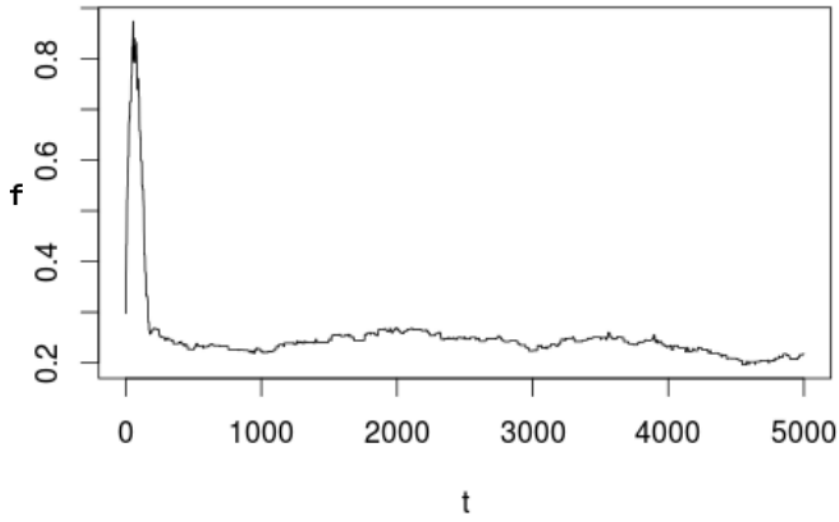


Figure 13: History plot for f

After some time, the sample chains for each of the parameters end up in their expected region, that is, near the actual values that were used to generate the data, although some parameters are closer than others. Samples of N climb up from the initial value of 1 into the expected region of 10, by the $20th$ sample. They are the closest to their expected value of the three parameters and a mean of 9.300927 was obtained. The sample chain for p drops from the initial value of 0.0017 way below the expected value of 0.001 to 0.0004 before rising back to the region around 0.0008 where it is maintained. This is far from the expected value of 0.001 but still a fair estimate. Lastly, the sample chain of f shoots up from the initial value of 0.25 to 0.9 before falling back to the region of interest by the $10th$ sample. Due to the nature of movement of their chains, the sample means of p and f are not very accurate compared to the actual values. Means of 0.0008257983 and 0.2602748 were obtained for p and f respectively.

The corresponding autocorrelation plots are shown below.

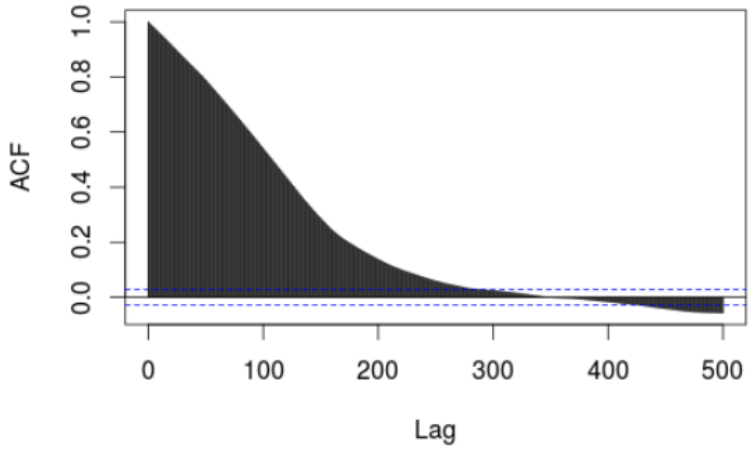


Figure 14: The autocorrelation function for N. .

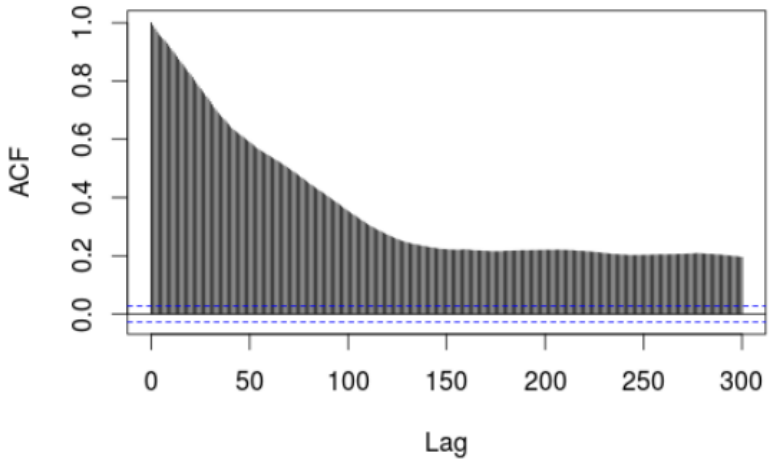


Figure 15: The autocorrelation function for p.

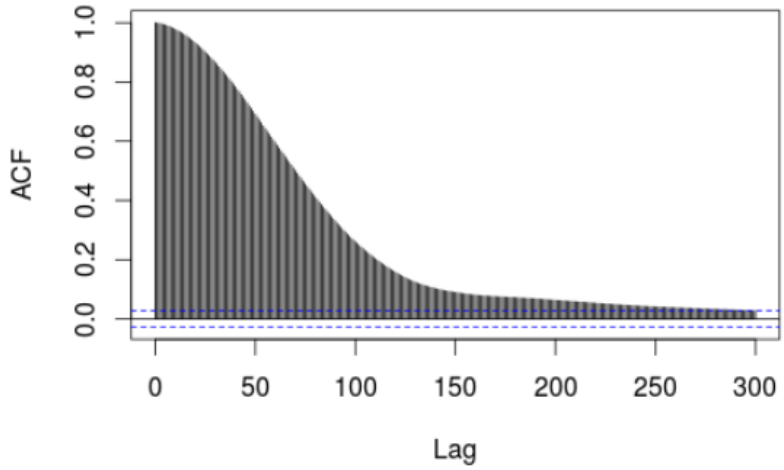


Figure 16: The autocorrelation function for f.

All the three parameters have a high autocorrelation, with dependence between the $1st$ sample and any other sample persisting upto the $100th$ sample. The autocorrelation for samples of N is significantly higher upto the $200th$ sample but tapers down to zero by the $300th$ sample. Those for p and f drop down to low values by the $150th$ value but the autocorrelation for p is significantly above zero by the $300th$ value. For this run of the Gibbs sampler, samples of f have the least autocorrelation, tapering down to zero by the $250th$ value. We note that this is one instance of the Gibbs implementation and these results may not be reproducible for different runs.

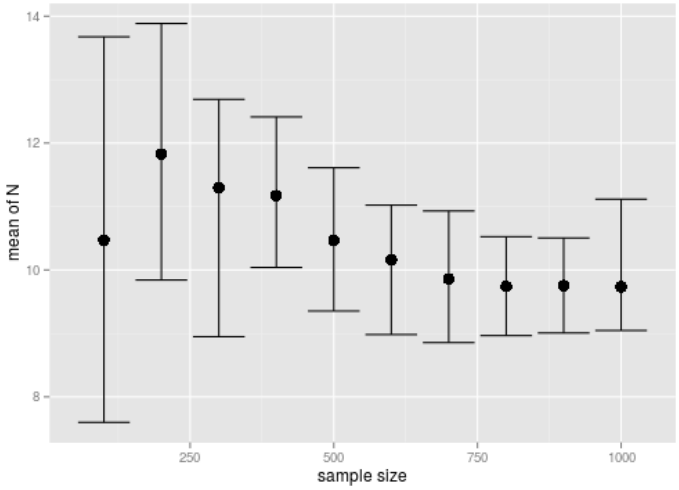
To obtain more accurate results, we would have to thin by picking roughly each $200th$ value of f and each $300th$ value for N but this would leave few data values for estimation, considering our total of 5000 sample values.

The most plausible reason for such behaviour (high autocorrelation)is estimating p and f concurrently. From the model, in Eq. (3.25), p and f are proportional to each other. Another possible cause for such high correlation could be the choice of the step size. As the number of parameters that need tuning increases, it becomes more difficult to obtain the optimal combination of step sizes.

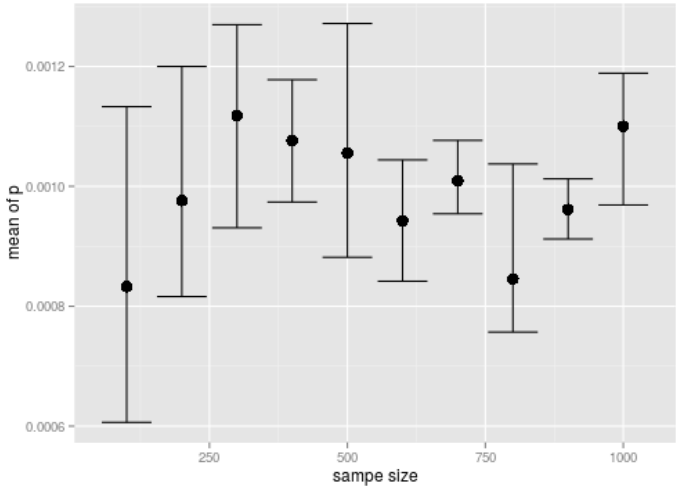
Gibbs sampling for different sample sizes

Our developed Gibbs algorithm was used to obtain 95% credible intervals for different sizes of final copy number x_i, ranging from 100 to 1000. For each sample size, the Gibbs sampler was run for 5000 iterations with fixed initial values of $N = 10, p = 0.001, f = 0.3$ and fixed step sizes of $AN = 0.09, Ap = 5 \times 10^5, Af = 0.05$. We note that no thinning was done because of the running time of the code. The plots in Figure 17 below show the trend of the mean and credible interval, as we move from 100 to 1000 sample values, x_i.

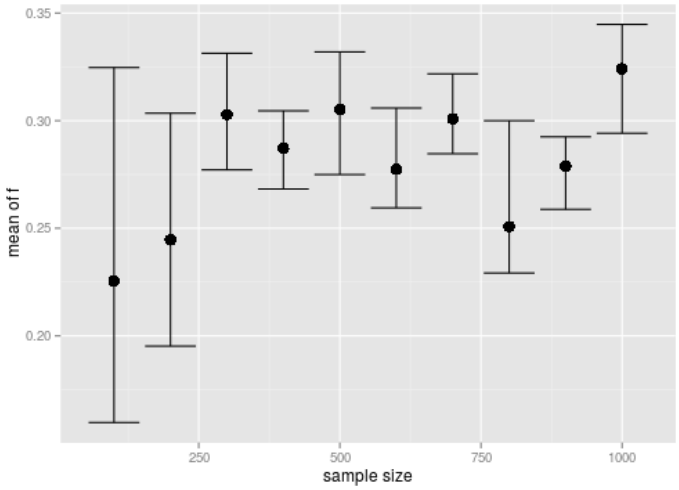
For parameter N in Figure 17a, as the sample size increases, the mean of sampled values increases far beyond 10 to almost 12 at the sample size of 200 and then decreases gradually to slightly less than 10. It is notable is that the length of the credible interval is shorter for larger sample sizes than for small sample sizes.



(a)



(b)



(c)

Figure 17: Variation of the mean and 95% credible interval with increase in sample size for N(a) , p (b) and f (c).

For p and f, the means oscillate around their expected values as the sample size increases, with the length of the 95% credible intervals increasing and decreasing haphazardly. However the lengths of the final three sample sizes are shorter than those obtained for small sample sizes. The discrepancies observed, as explained above, are as a result of estimating p and f concurrently.

Inference of parameters N and p when f is constant.

Due to the high correlation obtained when estimating all the three parameters simultaneously, as done for the method of moments in the first part, we also estimated N and p for fixed f in the case when the value of f is available. The estimates for both N and p improved and so did the autocorrelation as shown in the Figures 18,19,20,21 below.

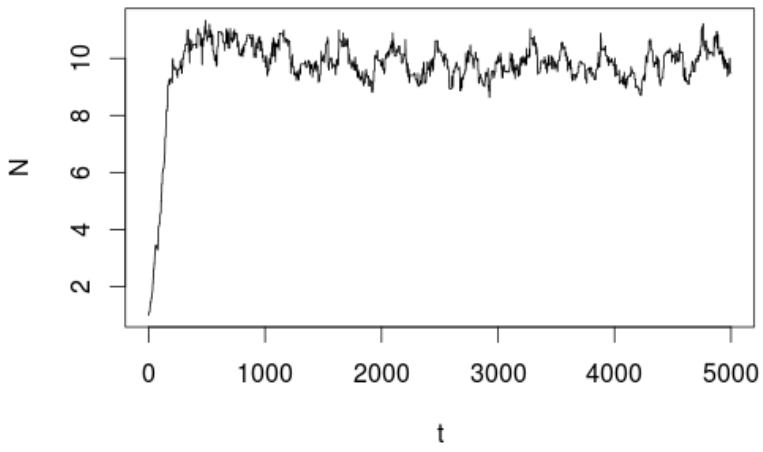


Figure 18: Chains of N when f is constant.

Using similar initial values and step sizes as with the case of changing f above, the Gibbs sampler was run for 5000 runs and means equal 9.728982 and 0.0009638469, were obtained for N and p respectively. Sampled values for N (Figure 18) rise from 1 to 10 by the 50th value and stay around 10. Those for p (Figure 19) drop from 0.0017 to the region around 0.0001

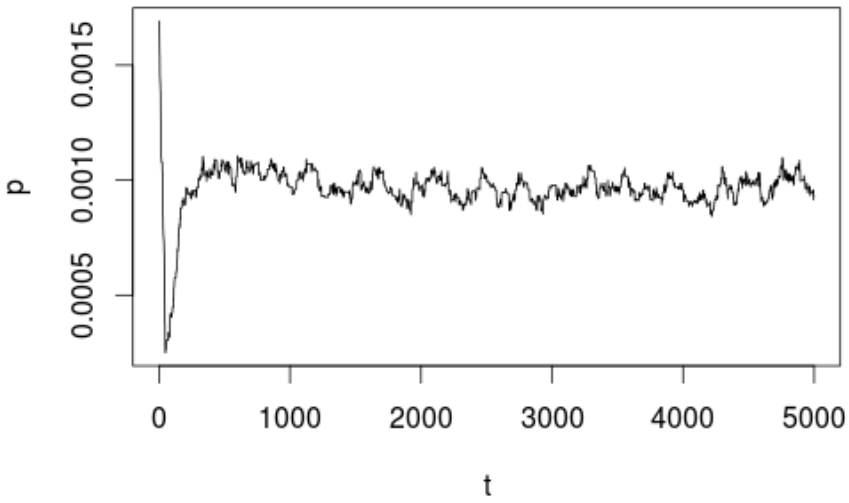


Figure 19: Chains of p when f is constant.

before rising back to 0.001, where they are maintained. The autocorrelation functions obtained with this special case are shown below.

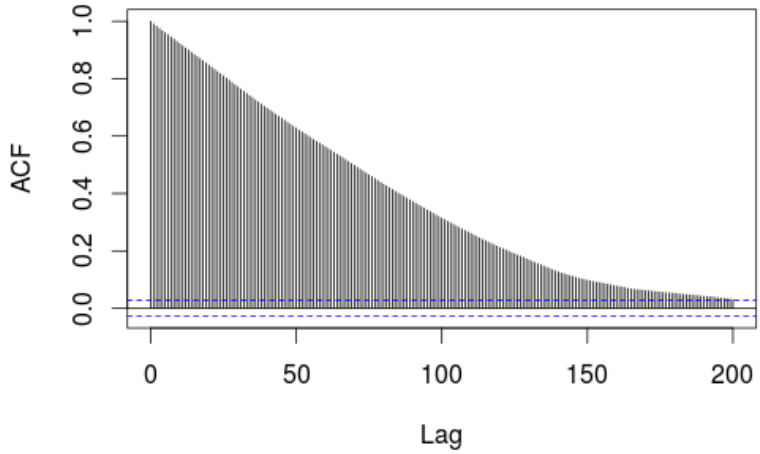


Figure 20: Autocorrelation function of p when f is constant.

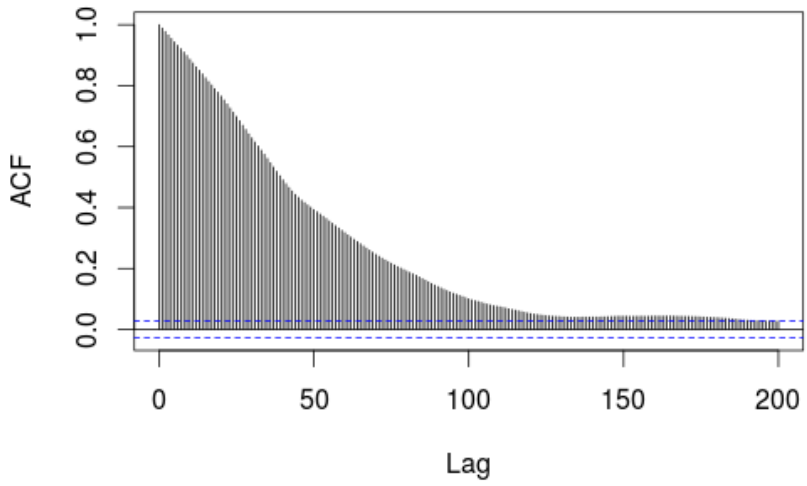
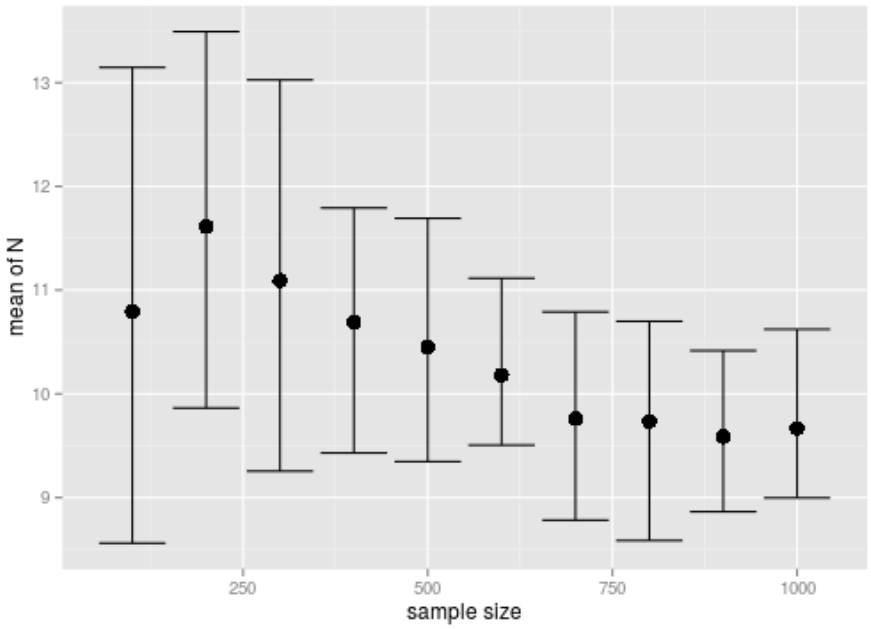


Figure 21: Autocorrelation function of p when f is constant.

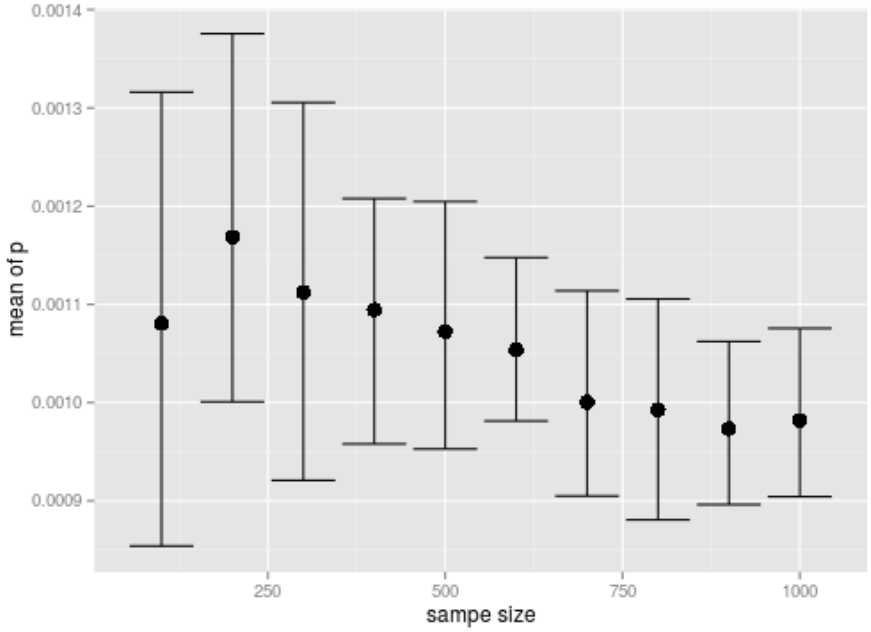
The autocorrelation for N doesn't deviate much from the case with changing f but that of p reduces significantly, dropping 0 by the 125 value. More precise samples can be obtained by generation very long chains and then picking every $125th$ and $200th$ value for p and N respectively.

Gibbs sampling for different sample sizes when f is constant

With the above results, we went ahead to examine the effect of sample size on the mean and credible interval of the estimates for N and p when f is constant. The Gibbs sampler was run 5000 times for each sample size with fixed initial values of $N = 10, p = 0.001$ and fixed step sizes of $AN = 0.06, Ap = 3 \times 10^6$. No thinning was done. The plots in Figure 22 below were obtained.



(a)



(b)

Figure 22: Variation of the mean and 95% credible interval with increase in sample size for N and p when f is constant.

Both means of N and p follow the same trend as sample size increases. They increase initially and then systematically decrease to near their expected values. Most importantly, the lengths of the credible intervals decrease systematically with 95% credible intervals of larger sample sizes being shorter

than those for small sample sizes.

3.8.1 Strengths and Weaknesses of Bayesian Inference over the Method of Moments

Bayesian inference of the parameters of PCR amplification has an edge over the method of moments discussed in the first part of this chapter because it yields consistent results. Small errors in some parameters like N do not affect significantly the accuracy of other parameter estimates. Also initial values used are not so much of an issue if the right step sizes are used. It also allows incorporation and updating of prior knowledge in order to inform the current model. With Bayesian inference, it is possible to obtain fairly accurate estimates with a small evidence size for N.

One of the main weaknesses of Bayesian inference is that for accurate results, the step sizes require substantial tweaking which gets worse as the number of parameters increases. This however can be resolved by updating the step sizes on the fly whilst the chains are being generated. Also, the accuracy of estimates by Bayesian inference is compromised by linear dependencies between parameters.

Having developed these two estimation methods, we go on to validate them by applying them to both real and synthetic data sets in Chapter Four. For quicker convergence of the Markov chain, parameter estimates produced by the method of moments in the first part can in principle be used to initialize the Gibbs sampler.

3.9 Summary

This chapter was developed in two parts. In the first part, we introduced and explained the a mechanistic model for correcting PCR induced bias that was developed in Ndifon et al. (2012). Upon employing mathe-

matical concepts of generating functions and techniques in combinatorics, this resulted in a probability distribution for the clonotype copy number x_i after t cycles of the PCR amplification. We then derived the first, second and third moments from the generating functions which we used to obtain expressions for each of the three parameters. Sample values were drawn from the probability distribution using the *distr* package in R and the derived method of moments was applied onto them to try and estimate back the parameters. We observed the accuracy of the results to depend much on the size of the samples and also that better results were obtainable if parameter f was assumed known. We concluded by discussing the strengths and weaknesses of the method of moments for estimating the parameters of our model for PCR amplification.

In the second part of this chapter, an introduction to Bayesian inference was given and examples were explained. The concept of likelihood function was introduced and the basics of Bayesian inference, that is, prior, likelihood function and posterior, were explained. A detailed introduction to Markov chains was given where the idea of stationary distributions and the Ergodicity theorem were covered. We then explored some of the ways of calculating the posterior probability which included conjugacy and Monte Carlo methods. Next came the two main methods for MCMC, that is, Metropolis Hastings and Gibbs sampling algorithm. Examples involving instances of these were given and the diagnostic methods to be used were also described. Starting with an example on the negative binomial distribution, a feel of the operations of MCMC was given, and the complexity involved with multivariate posterior distribution were portrayed in an example involving the normal distribution.

The most important part of the second part involved using the techniques of Bayesian inference to estimate the parameters N, p and f of PCR amplification. The posterior distribution of the parameters was derived us-

ing appropriate priors and full conditionals were extracted for each of the three parameters. A Gibbs sampling algorithm was implemented in R and the results obtained were presented. A high autocorrelation was obtained for all the three parameters which indicated poor mixing. However, lower auto correlation was obtained when the value of f was assumed known and fixed. For Bayesian inference too, precision of estimates increased with increase in sample size. Lastly the strengths and weaknesses of Bayesian inference versus method of moments were explored.

CHAPTER FOUR

DENOISING OF HIGH-THROUGHPUT DATA

4.1 Introduction

In this chapter, the two methods of parameter estimation developed in Chapter Three are applied; first to synthetic datasets, and later to real datasets. Using the probability distribution in Eq. (3.23), prototypes of real datasets are simulated and explained. The methods are then applied to datasets which were used to obtain the results published in Qi et al. (2014). The results of de-noising both datasets using our methods are presented. For both methods, I considered the case when f was known and was constant since this scenario produced better results in the previous chapter. In the case when it was unknown, an arbitrary f was chosen, whose effect cancelled out when I considered relative abundances of N rather than N itself.

4.2 Simulated Datasets

The costs involved in sequencing, limit the number of possible replicates of the same investigation. As such, the size of the evidence for a particular sequence with initial copy N_i is always limited as this is equal to the number of replicate experiments. In our data format, values in each column

are from individual replicates while values along each row are from individual sequences. In real experiments, more than 10 replicates are impractical as they are expensive to generate and thus the need for the maximum number of columns (number of evidence for N) to be 10. Each replicate has its own down sampling rate and therefore the values in each column are generated by the same value of f. If f was to be estimated, the values in each column would be its evidence. Secondly, more than 25 sequences are typically amplified with the same efficiency, which means that they have the same value for p. This implies that the evidence for p is has a wider allowance, while that for N is limited.

In an attempt to examine the effect of increasing evidence for p on the accuracy of estimates of N, four datasets were generated (tables of the form 25×5, 25×10, 50×5 and 50×10) using combinations of parameters p, N and f. For the $m \times 5$ datasets, $f = (0.1, 0.2, 0.3, 0.4, 0.5)^{\mathrm{T}}$ was used while for the $m \times 10$ datasets, $f = (0.1, 0.2, 0.3, 0.4, 0.5, 0.1, 0.2, 0.3, 0.4, 0.5)^{\mathrm{T}}$ was used. Similarly, for the $25 \times n$ dataset, values of $N = 1, \cdots, 25$ were used while for the $50 \times n$ datasets values of $N = 1, \cdots, 50$ were used. Every value in each of the tables depends on $p = 0.0017$.

The values in each column depend on one element of the vector f and values in each row depend on one of the values for N. In this set up, each of the $m \times 5$ datasets has 5 data points as evidence for N, and $m \times 5$ data points as evidence for p. Also each of the $m \times 10$ datasets has 10 data points as evidence for N, and $m \times 10$ data points as evidence for p.

Ten data sets were generated for each of the four formats described above. First the method of moments, and then Bayesian inference were applied to estimate N and p. The mean estimates and standard deviations for the 10 datasets of each kind were calculated and are presented below. Also as a measure of the closeness of estimates to the actual values, linear regression was performed on estimates from each of the 10 datasets and the

slope is presented in the proceeding sections.

4.2.1 Method of moments

From the generating function discussed in Chapter Three, we can define for each sequence i with counts in j samples, the following expected values:

$$\left\langle \frac{X_j}{f_j} \right\rangle = \frac{N_i}{p} = a_i, \tag{4.1}$$

and from Eq. (3.27)

$$\left\langle \frac{X_j(X_j-1)}{f_j^2} \right\rangle = \left(\frac{N_i}{p}\right)^2 + \frac{N_i}{p^2} - \frac{2N_i}{p} = b_i. \tag{4.2}$$

a_i is the expected value of X_i after normalising by the corresponding dilution constant, f. Similarly, b_i is the expected value of $X_i(X_i-1)$ after normalising by f^2.

From Eq. (4.1) and Eq. (4.2), we get:

$$b_i = a_i^2 + \frac{a_i}{p} - 2a_i. \tag{4.3}$$

a_i and b_i can be calculated from the data and used to estimate p. However, to make use of data from m different sequences, we can instead minimize the sum of the squared differences between the left and right hand sides of Eq. (4.3). That is, minimize

$$SSD = \sum_{i=0}^{m}(b_i - a_i^2 - \frac{a_i}{p} + 2a_i)^2 \text{ for } i = 1, 2, ..., m. \tag{4.4}$$

Figure 23 shows the estimates of p obtained using data points in the entire dataset as evidence for p. All the four estimates were obtained using the *optim* function in R as described above. We notice that as evidence

for p increases, the estimates get closer to the actual value of 0.0017 and the standard deviations decrease. However, datasets 25×10 and 50×5, which have the same size of evidence for p, seem to have different standard deviations.

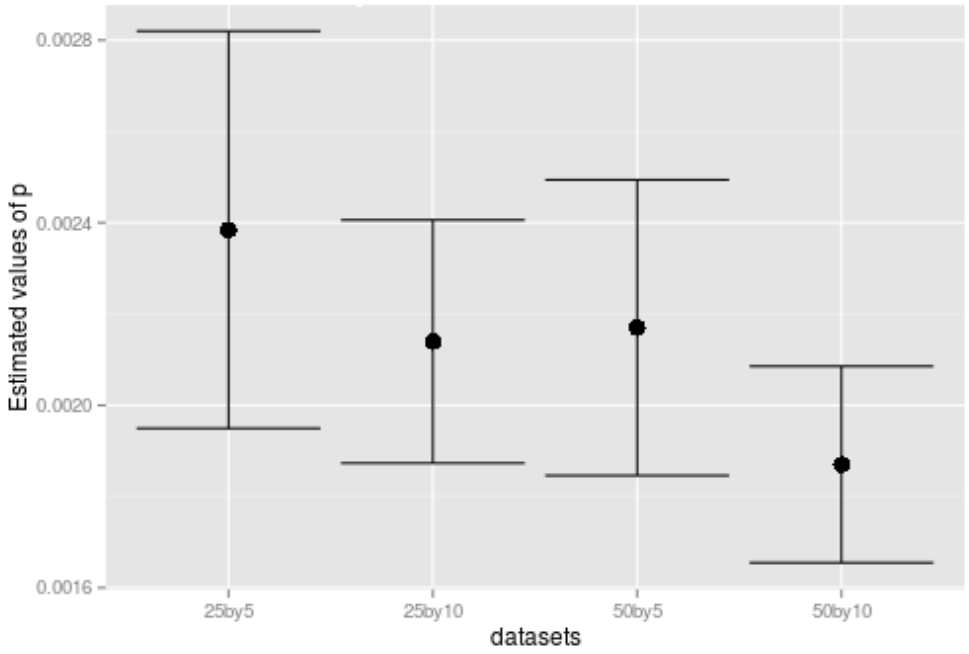
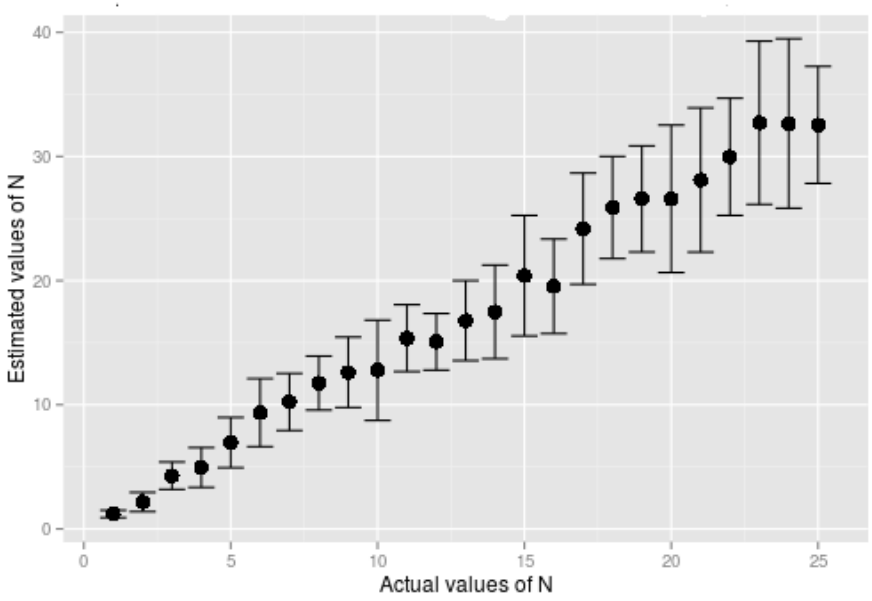


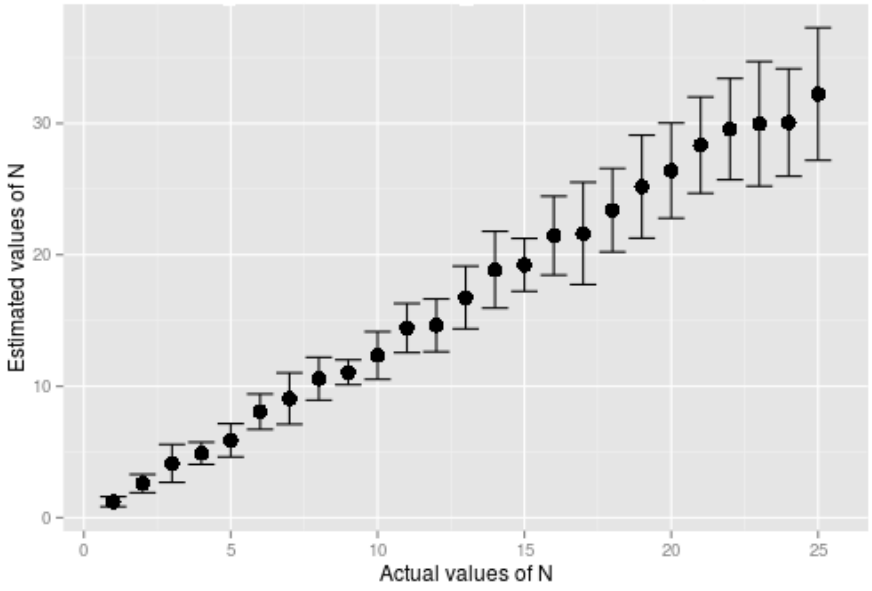
Figure 23: Estimates for p using method of moments. Each of the four estimates uses the entire dataset as evidence for p.

The observed difference is because the 25×10 dataset has more evidence for N than the 50×5 dataset, even though both datasets have the same evidence size for p.

The plots in Figure 24 below are obtained by calculating a_i and b_i from the respective dataset and then using this to obtain an optimized p, with the help of the *optim* function in R. The optimized value for p was then used to estimate N_i using the relation in Eq. (4.1). This was done for each of the 10 datasets for both the 25×5 and the 25×10 data formats.



(a) Estimates of N using the method of moments for the 25 by 5 datasets



(b) Estimates of N using the method of moments for the 25 by 10 datasets

Figure 24: Estimates of $N = 1, 2, \cdots 25$, using the method of moments. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence. A value of p, optimised over all (a_i, b_i) pairs is used to obtain each of the estimates.

The mean and standard deviations for each of the N_i's and p from the 10 datasets, were obtained and are presented in Figure 24 and in the Figure 23 above for p. One important observation is that when evidence for N increases from 5 to 10 (which implies that evidence for p increases from 125 to 250), the trend of the estimates of N increase systematically and the standard deviation decreases. The Pearson correlation coefficient between the actual and estimated values of N increases from 0.9953524 in Figure 24a to 0.9984065 in Figure 24b and the gradient of the fit of the estimates for N_is decreases from 1.354 to 1.302. However the estimates for N_i still remain far from the actual value. For example $N = 25$ is estimated as approximately 33.

In Figure 25, after obtaining a_i and b_i from the respective datasets, they are used to calculate p using the relation in Eq. (4.3). The result is a p_i value for each row in the dataset. This p_i is then used to estimate N_i, still using Eq. (4.1).

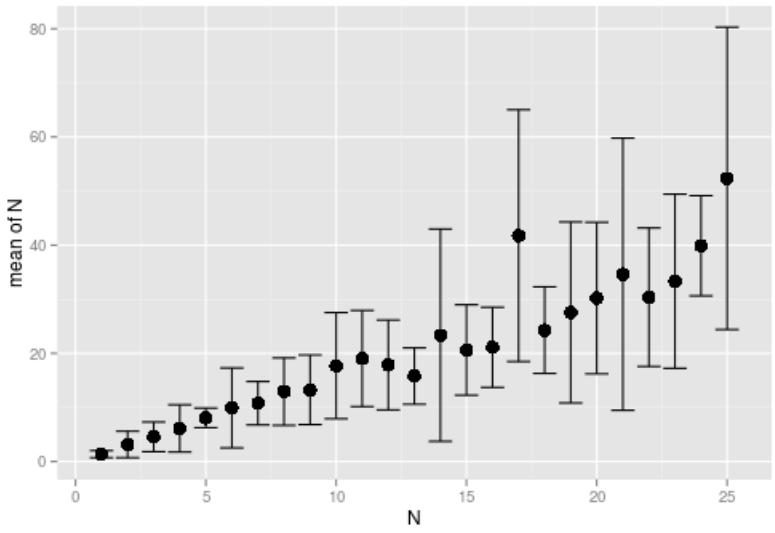


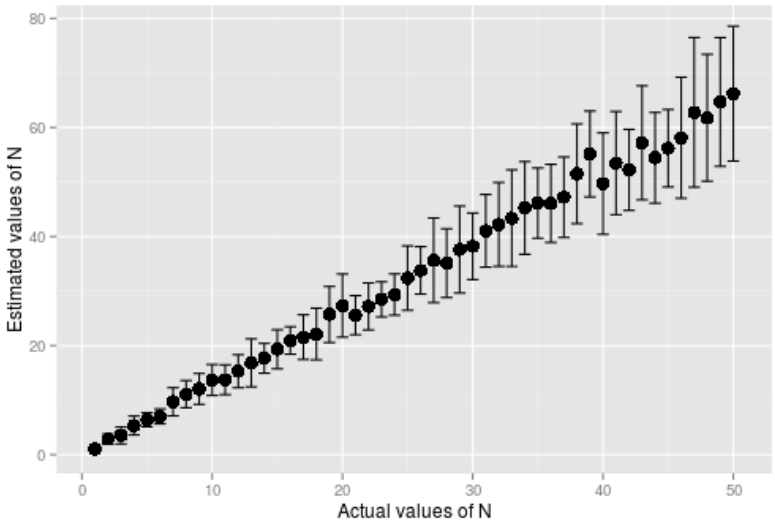
Figure 25: Estimates for $N = 1, \cdots, 25$ for the 25×10 dataset using the method of moments, each obtained using its own p_i calculated from Eq. (4.3).

.

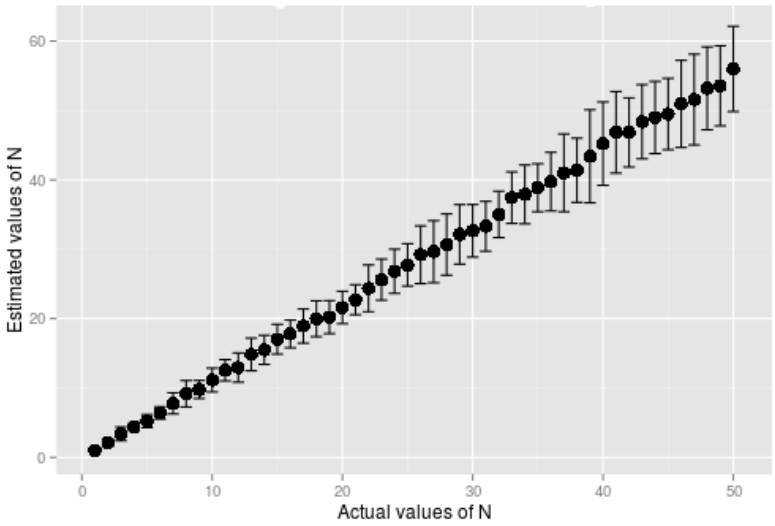
In this setting, each p_i has the same evidence as N, that is 5 data points

for the 25×5 datasets and 10 data points for the 25×10 datasets. The decrease in evidence for p is manifested by very poor estimates for N_i and large standard deviations as shown in Figure 25 above. The Pearson rank correlation dropped from 0.9953524 to 0.9377997 and for the same, 25×5 dataset the slope increased from 1.354 to 1.616 far from the expected slope of 1.

Figure 26 shows the effect of increasing the number rows (distinct N_i's) from 25 to 50. This doubles the evidence for p while maintaining the evidence of N_i for the two data formats shown in Figure 24. Since the 50×5 datasets have the same evidence for p as the 25×10, the plot in Figure 26a, upto $N = 25$, has almost similar features to that in Figure 24b. A correlation coefficient equal to 0.9976778 was obtained, slightly lower than that obtained for the 25×10 dataset. As explained above for p, the reason for this observation could be that for the same size of evidence for p, the 25×10 dataset has more evidence for N than for the 10×5 dataset. A slope of 1.297 was obtained for the fit of N_is estimated from each dataset.



(a)



(b)

Figure 26: Estimates of $N = 1, 2, \cdots 50$, using the method of moments. A value of p, optimised over all (a_i, b_i) pairs, is used to obtain each of the estimates. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence.

The 50×10 dataset in Figure 26b , has twice the evidence of p as compared to the 50×5 and this explains the better estimates for N_i and smaller standard deviations. For this dataset, a correlation equal to 0.9995978 and a slope of 1.109 were obtained. Particularly a difference in precision of estimation between the two datasets is observed around $N = 40$ in Figure 26.

For each data format, the trends of the estimates for N from each of the 10 datasets are presented below.

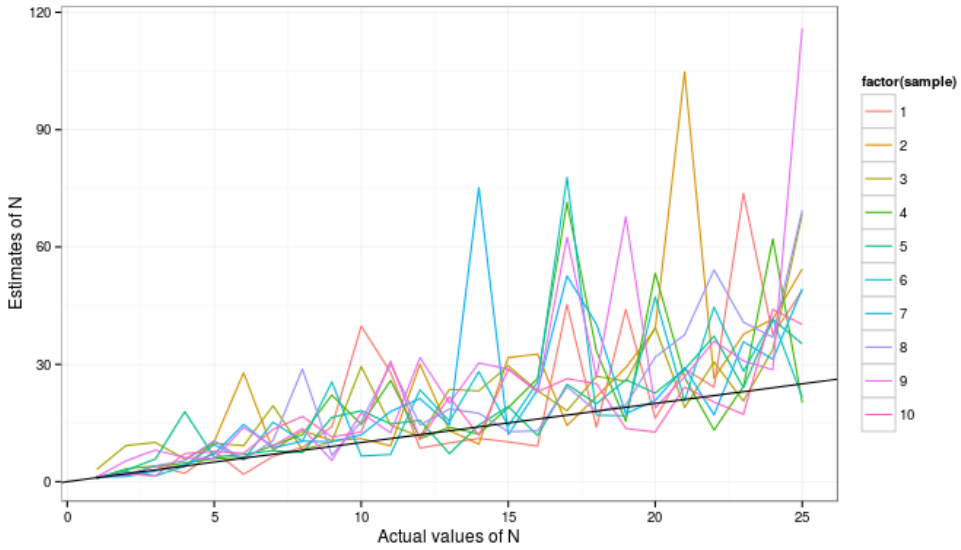


Figure 27: Estimates of N for each of the ten 25×5 datasets, whereby each row has its own p_i value

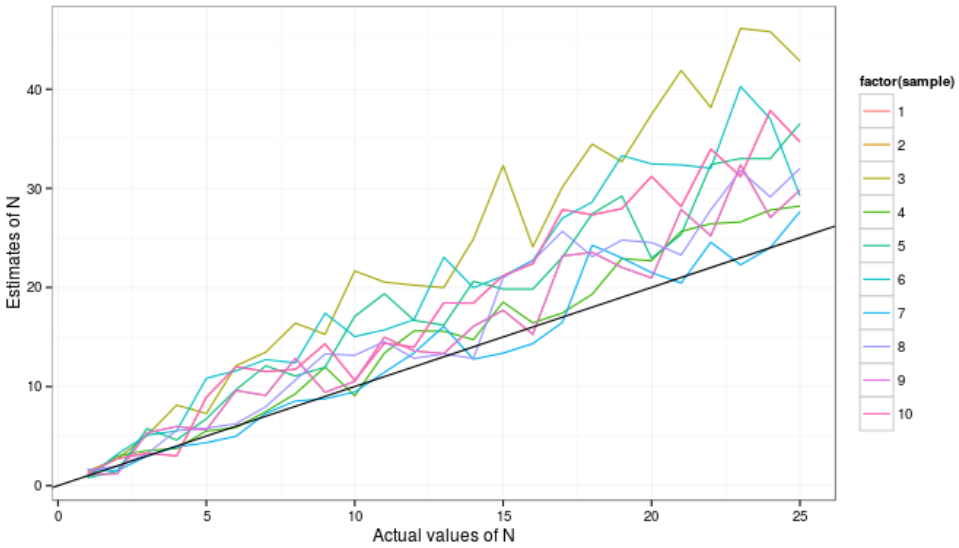


Figure 28: Estimates of N for each of the ten 25×5 datasets, with a single optimised p.

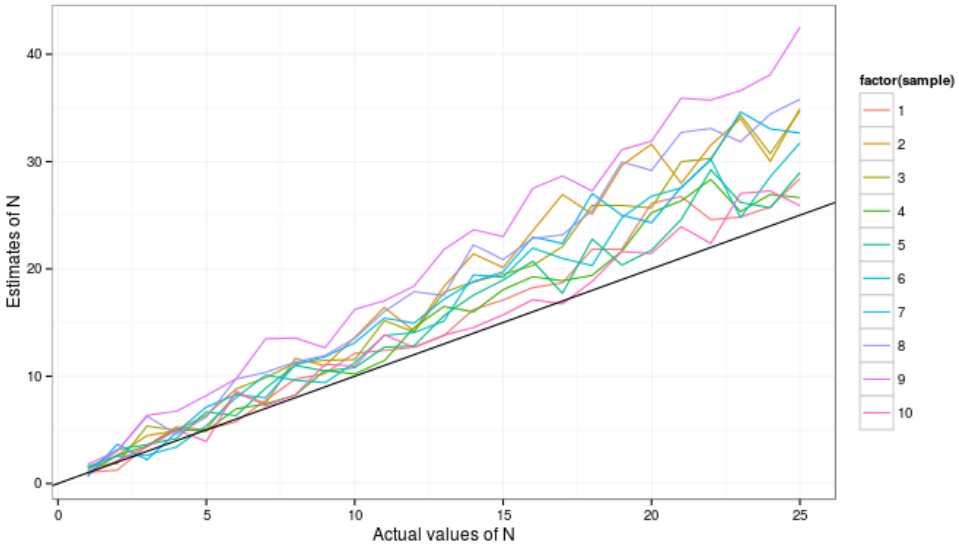


Figure 29: Estimates of N for each of the ten 25×10 datasets, with a single optimised p.

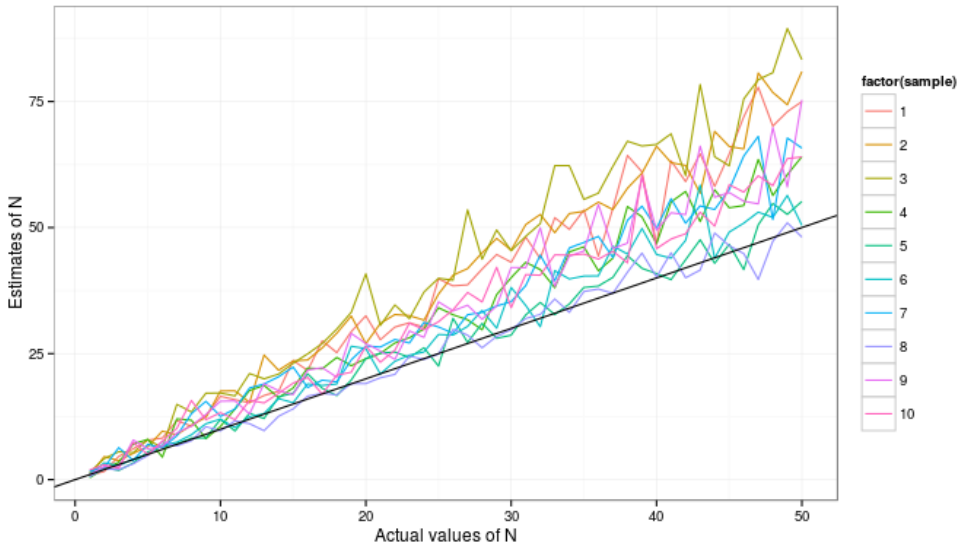


Figure 30: Estimates of N for each of the ten 50×5 datasets, with a single optimised p.

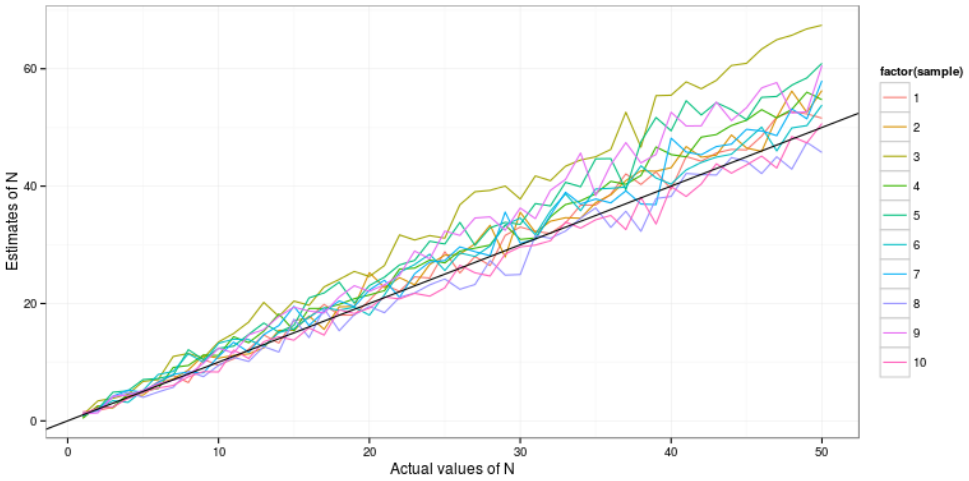


Figure 31: Estimates of N for each of the ten 50×10 datasets, with a single optimised p.

As the evidence for p decreases the area of diversion increases and we have more samples intersecting the line of expected values (black).

4.2.2 Bayesian Inference

The estimates for the parameters N and p obtained using the method of moments were then used as initial values for the Bayesian method de-

scribed in the second part of Chapter Three. Still, the evidence for each N_i is along the corresponding row in each dataset, while that for p is the entire dataset. The code used in Chapter Four was also adapted from a single sequence to suit multiple sequences. Still, a log-normal proposal distribution was employed in the Metropolis Hastings algorithm for N and a beta proposal was used in the Metropolis Hastings for p. We chose to use the case when f is known as this produced better results in Chapter Three. The Gibbs sampler was ran for 3000 runs for each of the 10 datasets of each of the 4 formats. The means and standard deviation for the 10 datasets were again calculated and the results are presented below.

In Figure 32, each of the estimates for p were obtained by using each of the data points in the dataset as evidence for p. The corresponding estimate obtained using the method of moments was used as the initial value and a step size of 5×10^5 was used for the beta proposal distribution function.

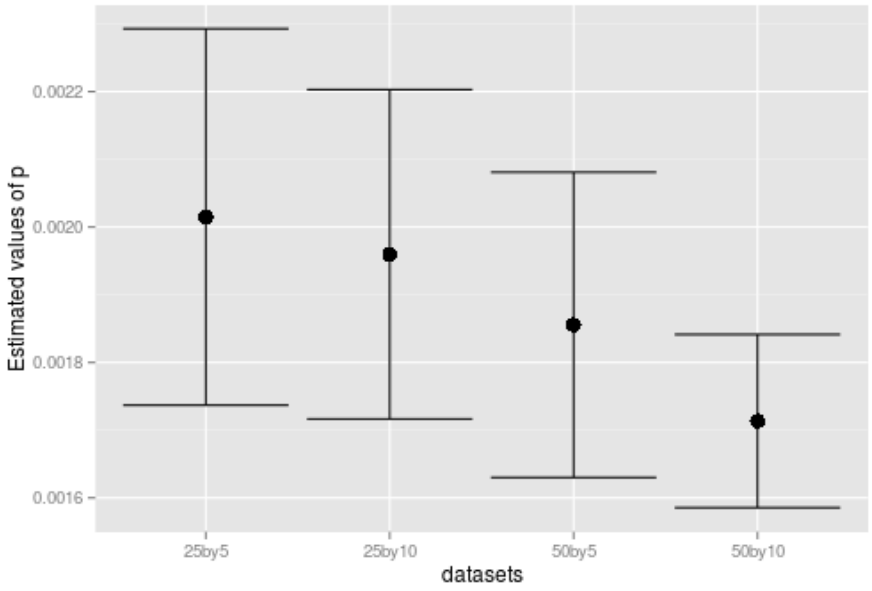
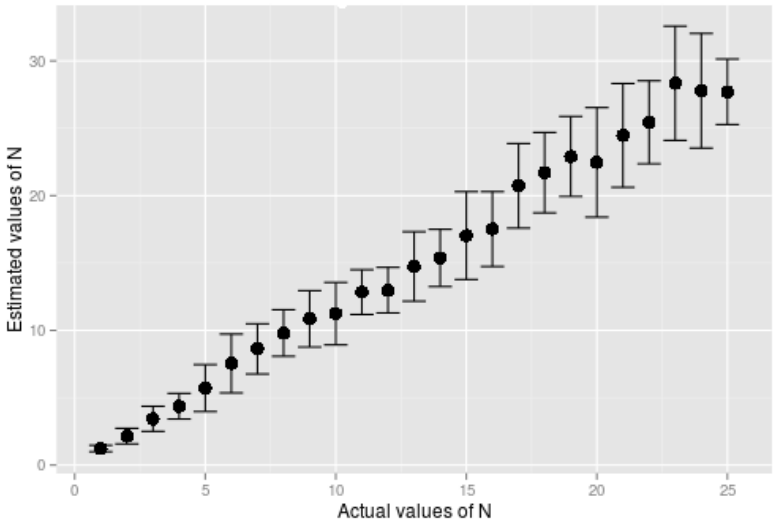


Figure 32: Estimates of p using Bayesian inference. Each of the four estimates uses the entire dataset as evidence for p, rather than only the entries in each row.

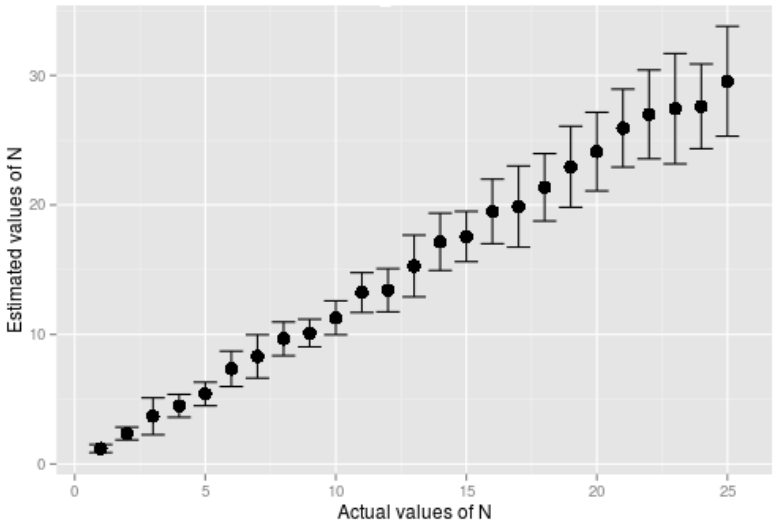
As the number of distinct N_i with the same value of p increases (which translates into more evidence for p), the estimate for p gets closer to the

real value of 0.0017 and the standard deviation decreases.

Figure 33 shows estimates for N obtained by applying Bayesian Inference to the 25×5 and 25×10 datasets. A step size of 5×10^5 was used to sample p while step sizes of 0.3 and 0.1 and were used to sample N from the log-normal proposal for the $m \times 5$ and $m \times 10$ datasets respectively.



(a)

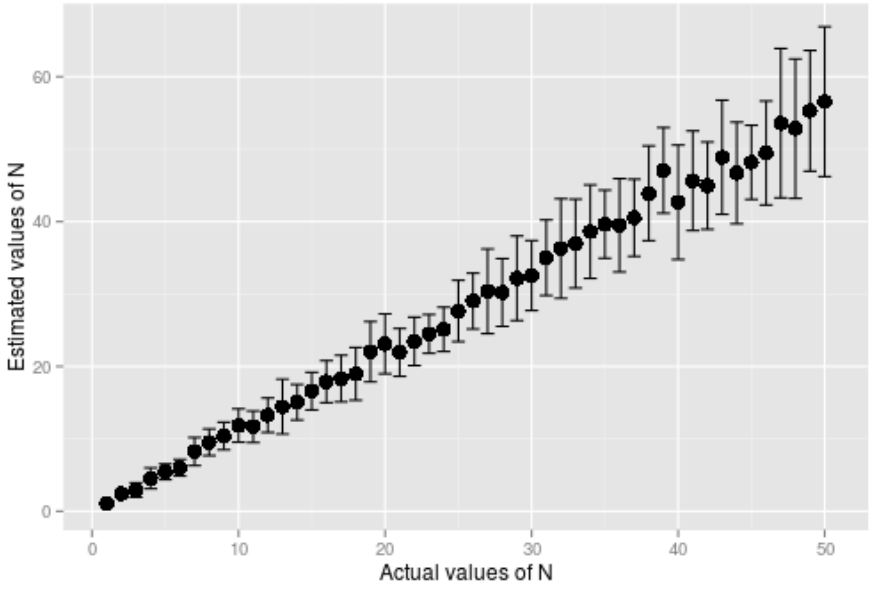


(b)

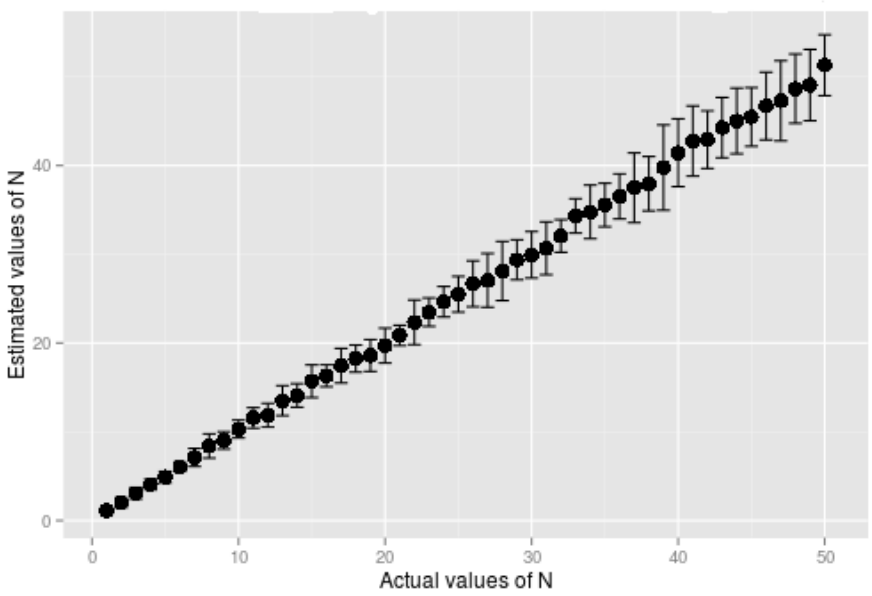
Figure 33: Estimates of $N = 1, 2, \cdots 25$, using Bayesian Inference. A value of p optimised over all (a_i, b_i) pairs is used to obtain each of the estimates. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence.

As the evidence for N increases from 5 to 10, the standard deviations decrease as observed in Figure 33 above. The correlation coefficient increased from 0.9966768 to 0.9986322 and the slope of the fit of N_is decreased from 1.16 to 1.142. We notice that the estimate for N, though not very accurate, is better than that obtained using the method of moments, for the same data size. This is even more apparent from the comparison of the two Pearson correlation coefficients and the slopes (Table 2 and Table 3).

Figure 34 shows estimates for N obtained by applying Bayesian Inference to the 50×5 and 50×10 datasets. When we increase the evidence for p by increasing the number of distinct N_i's with the same p from 25 to 50, we still have the same size of evidence for p as in the case of 25×10 dataset. A correlation equal to 0.9978789 is obtained, slightly lower than that obtained with the 25×10 dataset. The slope though was lower (1.109). Further increment of evidence for p and N, that is from 50×5 to 50×10, produces the best estimates for N and p with the least standard deviations and smallest slope (1.016), as seen in Figure 34b and in Figure 32 respectively. A correlation of 0.9996292 was obtained between actual values of N and its estimates.



(a)



(b)

Figure 34: Estimates of $N = 1, 2, \cdots 50$, using Bayesian Inference. a) Each N_i has 5 datapoints as evidence. b)Each N_i has 10 datapoints as evidence. A value of p optimised over all (a_i, b_i) pairs is used to obtain each of the estimates.

Similarly, the trends of the estimates for N from each of the 10 datasets were obtained and are presented below.

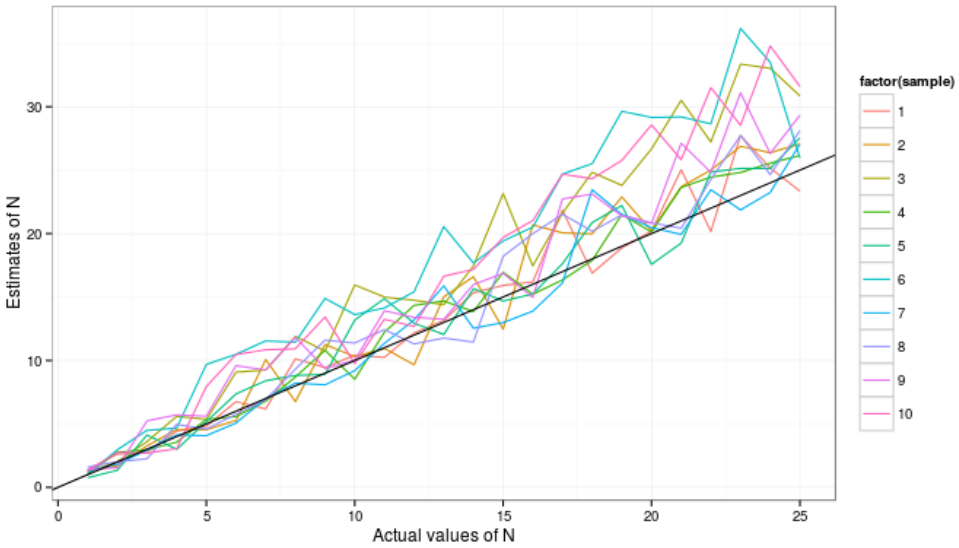


Figure 35: Estimates of N for each of the ten 25×5 datasets.

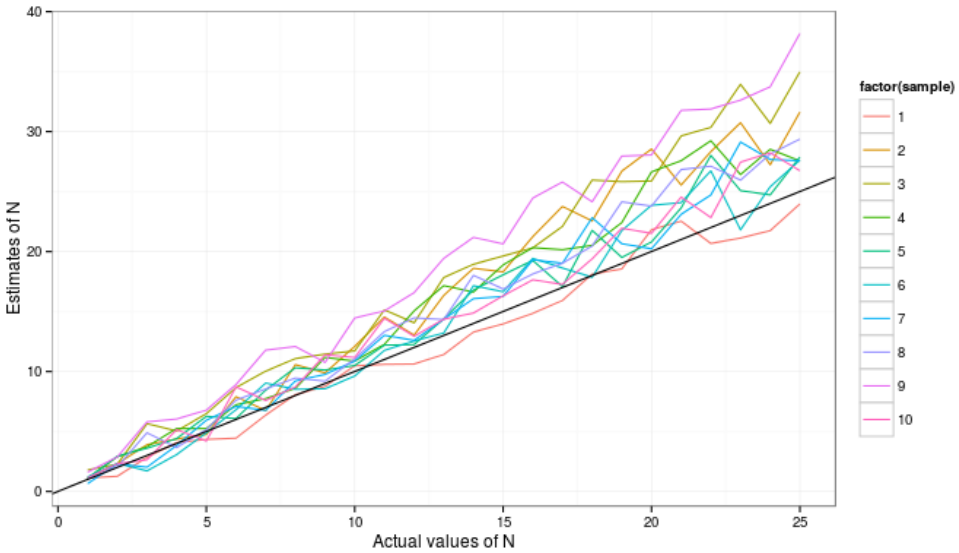


Figure 36: Estimates of N for each of the ten 25×10 datasets.

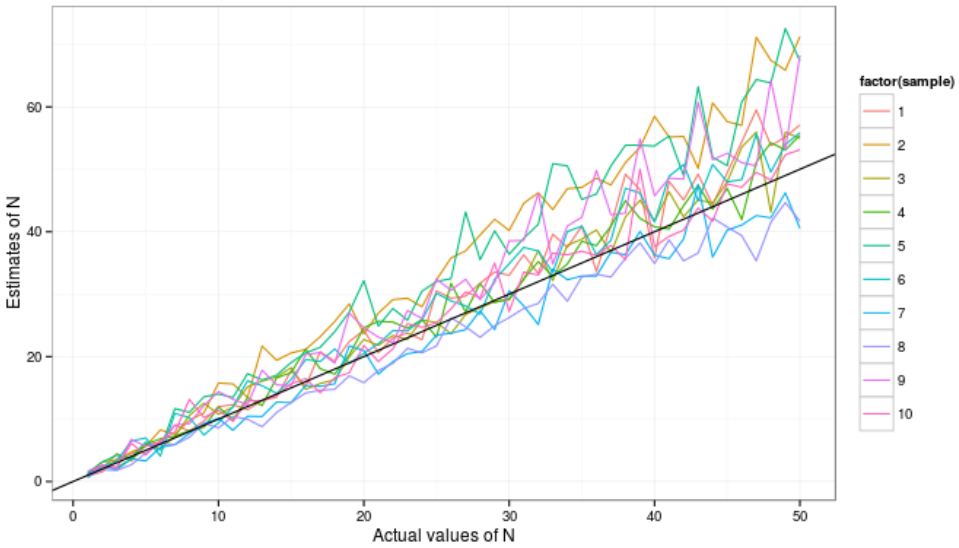


Figure 37: Estimates of N for each of the ten 50×5 datasets.

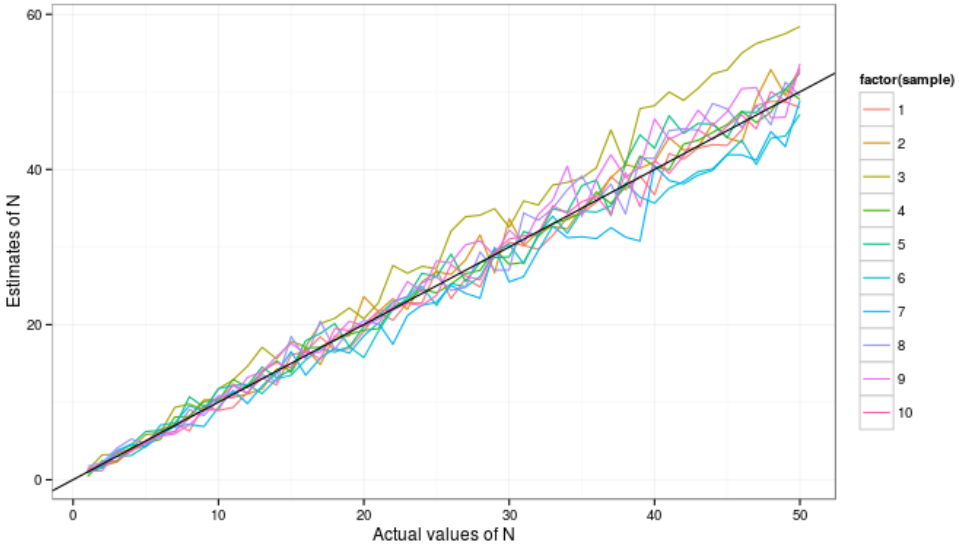
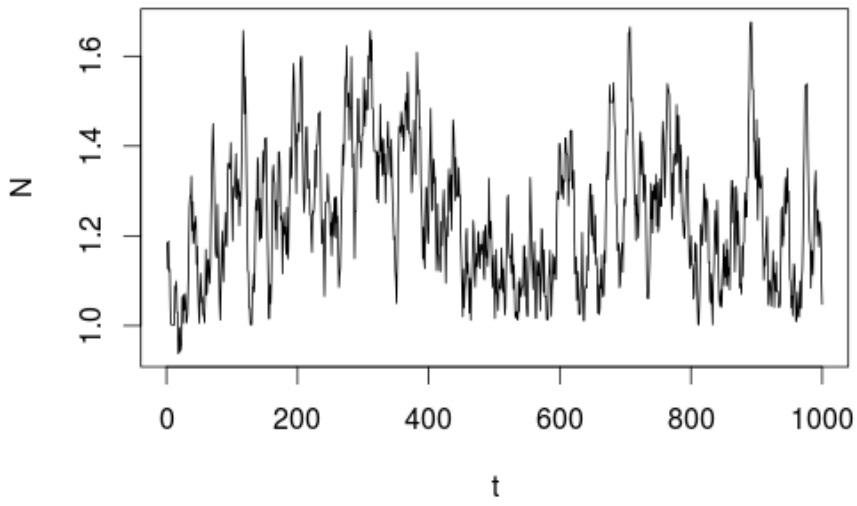


Figure 38: Estimates of N for each of the ten 50×10 datasets.

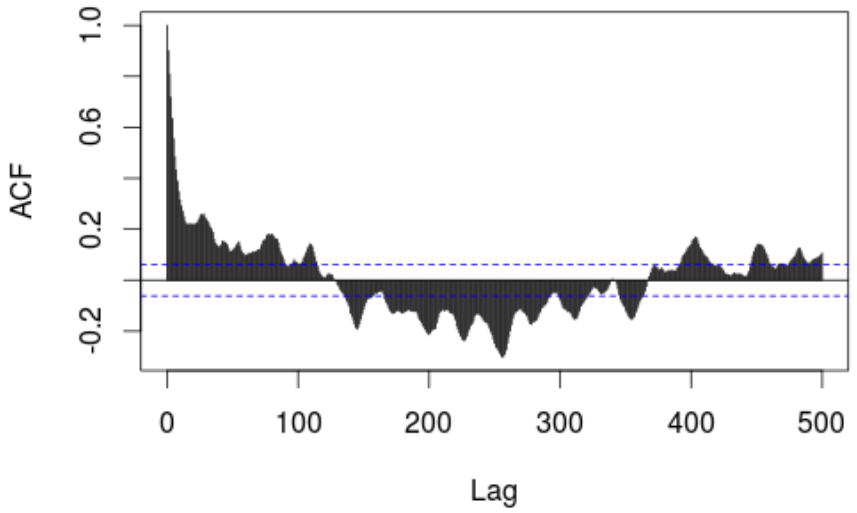
As the evidence for p increases the area of diversion decreases and we have more samples intersecting the line of expected values (black). This effect is greater than observed for corresponding datasets using the method of moments.

For the 50×10 dataset, which produced the best estimates, history plots and autocorrelation plots for a selected few parameters were extracted

to give a fair idea of how the Gibbs sampling was performing. Below we present history and autocorrelations plots for $N = 1, N = 25, N = 50$ and p. Each of the plots were obtained from a single iteration of the Gibbs algorithm and so the results are bound to slightly change for a different run with the same parameters.

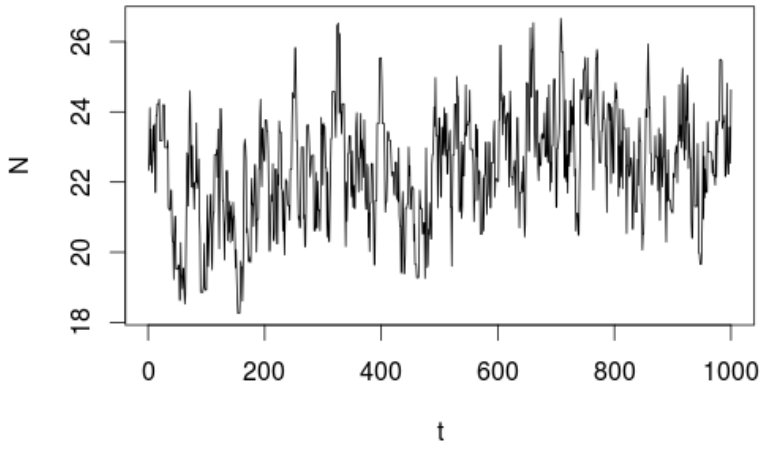


(a) History plot of samples generated from N = 1 as part of a 50 × 10 dataset.

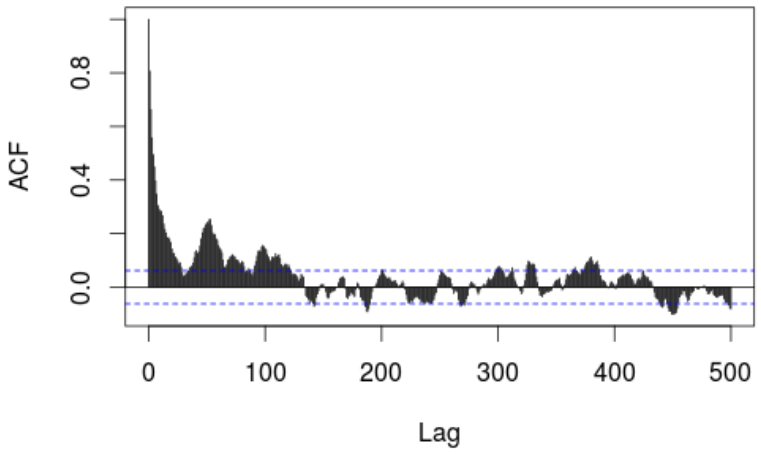


(b) Autocorrelation plot of samples generated from N = 1 as part of a 50 × 10 dataset.

Figure 39: Bayesian inference for $N = 1$.

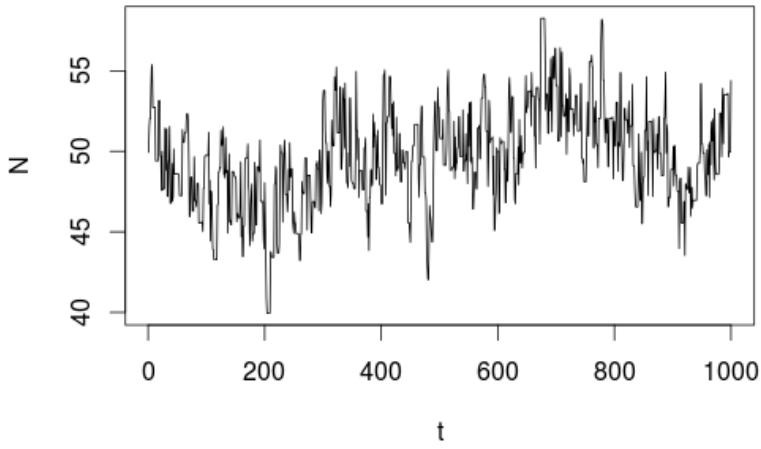


(a) History plot of samples generated from N = 25 as part of a 50 × 10 dataset.

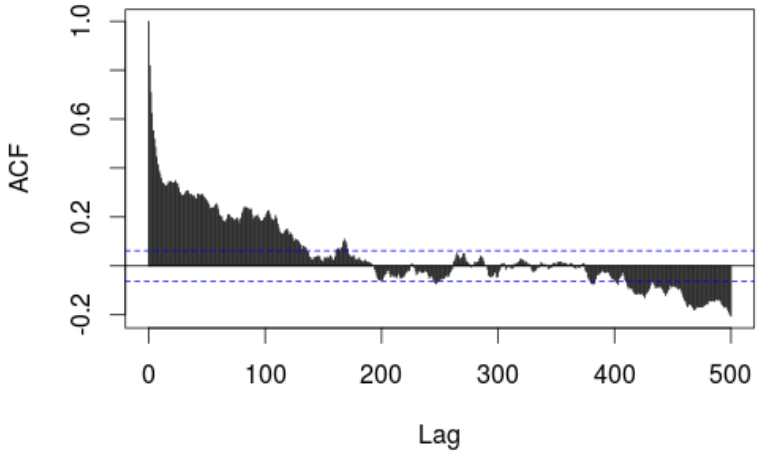


(b) Autocorrelation plot of samples generated from N = 25 as part of a 50 × 10 dataset.

Figure 40: Bayesian inference for $N = 25$.

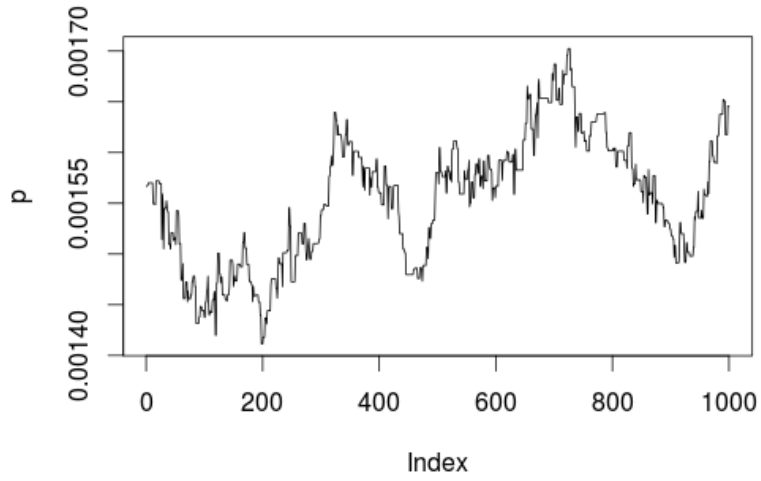


(a) History plot of samples generated from N = 50 as part of a 50 × 10 dataset.

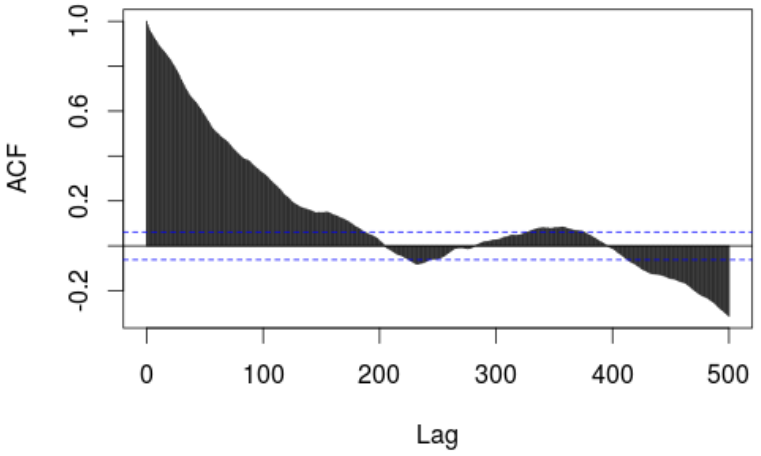


(b) Autocorrelation plot of samples generated from N = 50 as part of a 50 × 10 dataset.

Figure 41: Bayesian inference for $N = 50$



(a) History plot for p for the 50 $\times$10 dataset.



(b) Autocorrelation plot for p for the 50 $\times$10 dataset.

Figure 42: Bayesian inference for p.

As the size of N increases, the autocorrelation too increases. This mainly stems from the use of a uniform step size for the log-normal proposal across long range of values of N instead of each N having its own optimal step size. We also observe that p has the highest autocorrelation which also is a result of failure to obtain an optimal step size.

4.2.3 Summary of results from simulated data

From the results presented above, we observe that the greater the evidence for p, the better the estimate for p and hence the better the estimate for N. Also, more precise estimates for N are obtainable with a larger evidence size for N. Unfortunately, the possible evidence size for N is limited. We notice that Bayesian inference for p is very close to the actual value and that Bayesian inference is generally improvement of the method of moments. This can be observed in the Table 2 where the Pearson correlation coefficients of the Bayesian inference dominate their counterparts obtained using the method of moments. The dominance is even more apparent when we consider the difference in slopes between the two methods shown in Table 3. Nevertheless, estimates from both methods are acceptable for a larger evidence of parameter p, with the method of moments being preferable when computational time is a limiting factor.

Table 2: Pearson correlation coefficients between the estimated values and the actual values across all the datasets, using the two methods

Dataset	Method of moments	Bayesian Inference
25× 5	0.9953524	0.9966768
25× 10	0.9984065	0.9986322
50×5	0.9976778	0.9978789
50× 10	0.9995978	0.9996292

Table 3: Slopes of lines of fit for the estimated means using the two methods

Dataset	Method of moments	Bayesian Inference
25× 5	1.354	1.16
25× 10	1.302	1.142
50×5	1.297	1.109
50× 10	1.109	1.016

4.3 Denoising Real Datasets

Following satisfactory results from application of the denoising methods to synthetic data, we proceeded to apply them to real data. This data, generated using next generation sequencing, was used in Qi et al. (2014) to estimate a lower bound of the total number of different TCR beta (TCRB) sequences found in human T-cell repertoires and to evaluate the effect of age on TCRB diversity. As explained in Chapter One, this data contains errors and needs to be denoised if we are to obtain the correct biological information.

(Qi et al., 2014) obtained platelet donor apheresis lymphocytes from four young (aged 20-35 y) and five elderly (aged 70-85 y) adults from the Stanford Blood Centre. All individuals were healthy, regular platelet donors. The cells were purified and separated into Naive CD4 and CD8 and memory CD4 and CD8 T cells. For each individual and for each cell type, 5 replicates were obtained. This is why preference is made for the $m \times 5$ data format described earlier. During initial evaluation of TCRB repertoires in the young and elderly subjects, they compared the composition of TCRB gene rearrangements at the level of TCRBV and TCRBJ gene segment use and features of the CDR3-encoding junctional nucleotides.

The value of f used whilst preparing these datasets was unavailable and thus we opt to present the relative abundances of the clonotypes rather than their copy numbers, N_i. In this way, the scaling caused by using an

arbitrary f was done away with. Presented below are the relative frequency of TCRBV and TCRBJ segments that make up the naive CD4 lymphocytes for Donor 2 in Qi et al. (2014), before and after denoising using the method of moments.

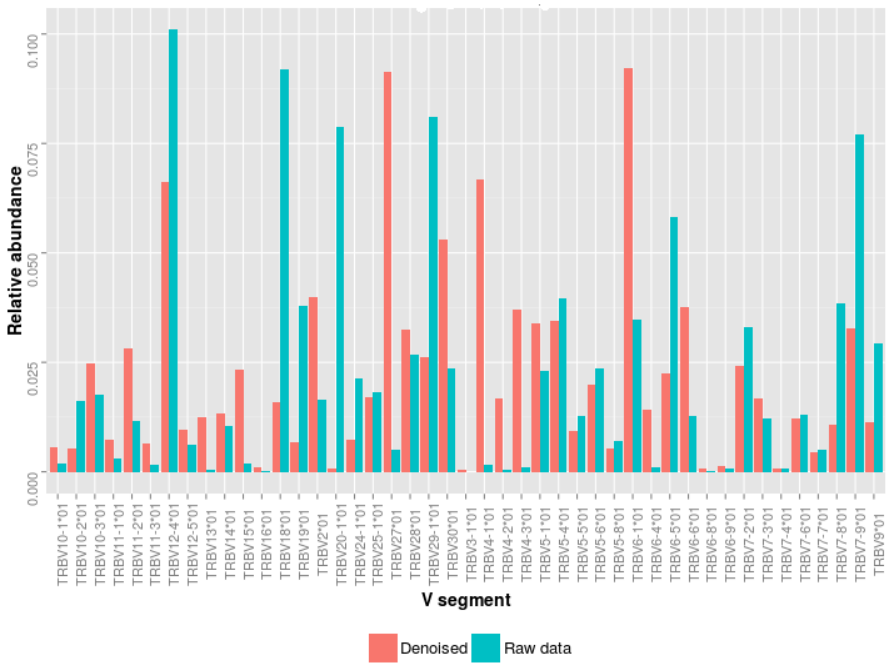


Figure 43: Relative abundancies of V gene segments before and after denoising.

Forty five distinct TCRBV segments were sequenced and their relative abundances are shown in 43. Clearly, there is a difference between the denoised and the noise containing relative abundances. Importantly, most of the V segments with almost zero relative abundance had their values increase significantly after denoising.

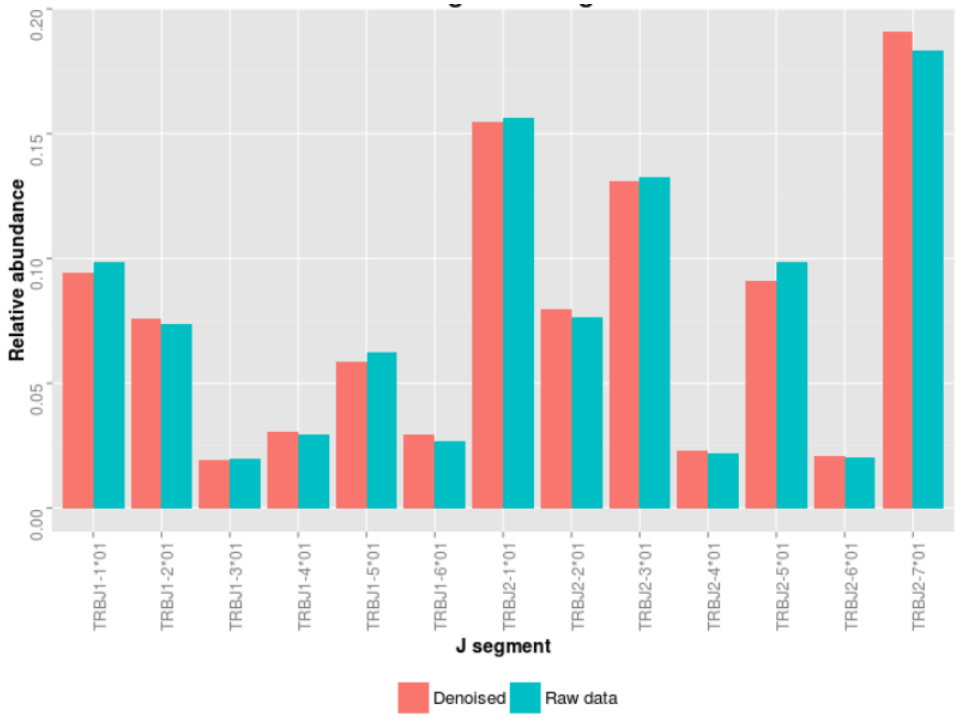


Figure 44: Relative abundancies of J gene segments before and after de-noising.

Thirteen distinct TCRBJ segments were sequenced. From 44 the relative abundances of these TCRBJ segments before and after de-noising are different but the difference is not as drastic as in the case of TCRBV segments.

Figures 44 and 43 seem to agree with the fact that the primers used in PCR amplification, which are are the main source of noise, are specific for the V segments.

4.4 Summary

In this chapter, the two methods of parameter estimation developed in Chapter Three were applied. In all the analyses, it was assumed that f was known since this had produced better results in Chapter Three. Firstly, synthetic datasets were generated and both methods were applied on them. Since the evidence for N is limited, the effect of increasing evidence for p,

on the precision of estimates for N, was investigated. It turns out that more evidence for p increases the accuracy of estimates for N and since evidence for p has a high upper bound, this can be utilised to generate better estimates for N. Also, from the Pearson correlation coefficient, we noticed that increase in evidence for N leads to better estimates for N.

Measures of slope and Pearson correlation coefficients were used to compare the two methods of inference and Bayesian inference emerged as superior to the method of moments. Surprisingly, adequately satisfactory results were obtained for the method of moments especially when evidence for p is high. This illuminated the idea of the method of moments being a preferable method, when working with large datasets that would actually take long to analyse using Bayesian inference.

Secondly, the method of moments was applied to real data that was used to publish the results in (Qi et al., 2014). Because there was no knowledge of the f used, the relative abundances of initial copy numbers N rather than N itself were estimated. Data obtained from sequencing 5 aliquots of CD4 Naive T-cells of Donor 2 were used to calculate the relative abundances of TCRBV and TCRBJ gene segments in the five aliquots and the resultant plots were presented. There existed significant differences between the denoised and raw data relative abundances for V segments but not for J segments. This observation is plausible since differences in amplification efficiencies are caused by different primers used during PCR amplification yet these primers are specific for the V segment.

All the analysis in this chapter was implemented in R.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

T-cell diversity has a great influence on the ability of the immune system to recognise and fight the wide variety of potential pathogens in our environment. The current state of art approach to profiling T-cell diversity involves high-throughput sequencing and analysis of T-cell receptors. Although this approach produces huge amounts of data, the data has inherent errors which might obscure the underlying biological picture. To correct these errors, two methods were developed; a method of moments and a method based on Bayesian inference.

Data was simulated, and both methods were applied on it. The results showed that the larger the evidence for p, the better the estimate obtained and thus the better the estimate for N. Using the Pearson correlation coefficient and the slope, the analysis of the simulated data showed that Bayesian inference generated more precise and accurate results than the method of moments. The method of moments however is preferable when time is a limiting factor due to large datasets as it is faster and adequately accurate.

Once applied to real data, the method of moments produces relative abundances that are significantly different from those plotted with raw data for the V segments. However the comparison of relative frequencies between raw and denoised data is not drastic for the J segments. This seems to be in line with the fact variation in PCR amplification of various sequences is caused by the primers used during the amplification process which are specific for the V segments.

The result of this work implies that a prior knowledge of f will yield better estimates for p and most importantly N and so a way should be followed during library preparation to record the value of f used. I recommend that for large datasets, the method of moments be used. However, for small datasets, the method based on Bayesian Inference is more appropriate.

5.2 Recommendations

One of the major drawbacks of Bayesian Inference is that choice of the optimal combination of step sizes for respective proposal distributions, is difficult. In the future, the step sizes can be updated on the fly until an optimum value is attained. Alternatively, the Gibbs sampling can be implemented in already existing high end packages in R or any other software.

In addition, an additional experiment needs to be conducted with known initial copy numbers so as to further examine the accuracy of the methods.

REFERENCES

Altuvia, Y., Schueler, O., & Margalit, H. (1995). Ranking potential binding peptides to mhc molecules by a computational threading approach. *Journal of molecular biology*, *249*(2), 244–250.

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome biol*, *11*(10), R106.

Anderson, R. M., May, R. M., & Anderson, B. (1992). *Infectious diseases of humans: dynamics and control* (Vol. 28). Wiley Online Library.

Arstila, T. P., Casrouge, A., Baron, V., Even, J., Kanellopoulos, J., & Kourilsky, P. (1999). A direct estimate of the human $\alpha\beta$ t cell receptor diversity. *Science*, *286*(5441), 958–961.

Berek, C., & Milstein, C. (1988). The dynamic nature of the antibody repertoire. *Immunological reviews*, *105*(1), 5–26.

Bernoulli, D. (1760). Essai d'une nouvelle analyse de la mortalité causée par la petite vérole et des avantages de l'inoculation pour la prévenir. *Histoire de l'Acad. Roy. Sci.(Paris) avec Mém. des Math. et Phys. and Mém*, 1–45.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 259–302.

Brusic, V., Rudy, G., Honeyman, G., Hammer, J., & Harrison, L. (1998). Prediction of mhc class ii-binding peptides using an evolutionary algorithm and artificial neural network. *Bioinformatics*, *14*(2), 121–130.

Burnet, S. F. M. (1959). *The clonal selection theory of acquired immunity*. University Press Cambridge.

Collins, F. S., Morgan, M., & Patrinos, A. (2003). The human genome project: lessons from large-scale biology. *Science*, *300*(5617), 286–290.

Davenport, M. P., Shon, I. A. H., & Hill, A. V. (1995). An empirical method for the prediction of t-cell epitopes. *Immunogenetics*, *42*(5), 392–397.

Davis, M. M., & Bjorkman, P. J. (1988). T-cell antigen receptor genes and t-cell recognition. *Nature*, *334*(6181), 395–402.

De Groot, A. S., Bosma, A., Chinai, N., Frost, J., Jesdale, B. M., Gonzalez, M. A., ... Saint-Aubin, C. (2001). From genome to vaccine: in silico predictions, ex vivo verification. *Vaccine*, *19*(31), 4385–4395.

De Groot, A. S., Sbai, H., Martin, B., & Berzofsky, J. A. (2002). Use of bioinformatics to predict mhc ligands and t-cell epitopes: application to epitope-driven vaccine design. *Methods in Microbiology*, *32*, 99–123.

De Groot, A. S., Sbai, H., Saint Aubin, C., McMurry, J., & Martin, W. (2002). Immuno-informatics: mining genomes for vaccine components. *Immunology and Cell Biology*, *80*(3), 255–269.

Fleckenstein, B., Kalbacher, H., Muller, C. P., Stoll, D., Halder, T., Jung, G., & Wiesmüller, K.-H. (1996). New ligands binding to the human leukocyte antigen class ii molecule drb1* 0101 based on the activity pattern of an undecapeptide library. *European Journal of Biochemistry*, *240*(1), 71–77.

Gardiner, C. W. (1985). *Handbook of stochastic methods* (Vol. 3). Springer Berlin.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*(6), 721–741.

Genomics. (2015). *what is a genome?* http://ghr.nlm.nih.gov/handbook/hgp/genome. ([Online; accessed 12-May-2015])

Goldstein, B., Faeder, J. R., & Hlavacek, W. S. (2004). Mathematical and computational models of immune-receptor signalling. *Nature Reviews Immunology*, *4*(6), 445–456.

Groves, D., Lever, W., & Makinodan, T. (1969). Stochastic model for the production of antibody-forming cells.

Hamer, W. (1906). *Epidemic disease in england: the evidence of variability and of persistency of type, ser. milroy lectures.* Bedford Press.

Hammer, J., Bono, E., Gallazzi, F., Belunis, C., Nagy, Z., & Sinigaglia, F. (1994). Precise prediction of major histocompatibility complex class ii-peptide interaction based on peptide side chain scanning. *The Journal of experimental medicine*, *180*(6), 2353–2358.

Hraba, T., & Doležal, J. (1990). Model-based analysis of cd4+ lymphocyte dynamics in hiv infected individuals. *Immunobiology*, *181*(1), 108–118.

Huse, S. M., Welch, D. M., Morrison, H. G., & Sogin, M. L. (2010). Ironing out the wrinkles in the rare biosphere through improved otu clustering. *Environmental microbiology*, *12*(7), 1889–1898.

illumina. (2015). *introduction to next generation sequencing technology .* http://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf. ([Online; accessed 1-May -2015])

Institute, N. H. G. R. (2015). *DNA Sequencing Costs.* http://www.genome.gov/sequencingcosts/. ([Online; accessed 1-April -2015])

Janeway, C. A., Travers, P., Walport, M., & Shlomchik, M. J. (2001). Innate immunity.

Jesdale, B. M., Deocampo, G., Meisell, J., Beall, J., Marinello, M. J., Chicz, R. M., & De Groot, A. S. (1997). Matrix-based prediction of mhc-binding peptides: the epimatrix algorithm, reagent for hiv research. *Vaccines*, 57–64.

Kauffman, S. A., Weinberger, E. D., & Perelson, A. S. (1988). Maturation of the immune response via adaptive walks on affinity landscapes. *Theoretical Immunology I. Addison Wesley*.

Kepler, T. B., & Perelson, A. S. (1993). Cyclic re-entry of germinal center b cells and the efficiency of affinity maturation. *Immunology today*, *14*(8), 412–415.

Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. , *115*(772), 700–721.

Levin, S. A., Dushoff, J., & Plotkin, J. B. (2004). Evolution and persistence of influenza a and other diseases. *Mathematical biosciences*, *188*(1), 17–28.

Link, W. A., & Barker, R. J. (2009). *Bayesian inference: with ecological applications*. Academic Press.

Lipford, G. B., Hoffman, M., Wagner, H., & Heeg, K. (1993). Primary in vivo responses to ovalbumin. probing the predictive value of the kb binding motif. *The Journal of Immunology*, *150*(4), 1212–1222.

lookfordiagnosis. (2015). *Receptors, antigen, t-cell, gamma-delta (Antigen Receptors, T-Cell, gamma-delta.* `http://www.lookfordiagnosis.com/mesh_info.php?term=receptors%2C+antigen%2C+t-cell%2C+gamma-delta&lang=1`. ([Online; accessed 12-May-2015])

Lütkebohle, I. (2008). *BWorld Robot Control Software.* `http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/`. ([Online; accessed 19-July-2008])

Macken, C. A., & Perelson, A. S. (1989). Protein evolution on rugged landscapes. *Proceedings of the National Academy of Sciences*, *86*(16), 6191–6195.

Marchalonis, J., & Gledhill, V. (1968). Elementary stochastic model for the induction of immunity and tolerance.

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

May, R. M., Anderson, R. M., & McLean, A. R. (1988). Possible demographic consequences of hiv/aids epidemics. i. assuming hiv infection always leads to aids. *Mathematical Biosciences*, *90*(1), 475–505.

Meister, G. E., Roberts, C. G., Berzofsky, J. A., & De Groot, A. S. (1995). Two novel t cell epitope prediction algorithms based on mhc-binding motifs; comparison of predicted and published epitopes from mycobacterium tuberculosis and hiv protein sequences. *Vaccine*, *13*(6), 581–591.

Merrill, S. J. (1989). Modeling the interaction of hiv with cells of the immune system. In *Mathematical and statistical approaches to aids epidemiology* (pp. 371–385). Springer.

Microbiology, N. R. (2015). *High-throughput sequencing platforms.* `http://www.nature.com/nrmicro/journal/v10/n9/fig_tab/nrmicro2850_F1.html`. ([Online; accessed 1-April -2015])

Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., & Snyder, M. (2008). The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, *320*(5881), 1344–1349.

nature. (2015). *Immulogy.* `http://www.nature.com/subjects/immunology`. ([Online; accessed 2-May-2015])

Ndifon, W., Gal, H., Shifrut, E., Aharoni, R., Yissachar, N., Waysbort, N., ... Friedman, N. (2012). Chromatin conformation governs t-cell receptor jβ gene segment usage. *Proceedings of the National Academy of Sciences*, *109*(39), 15865–15870.

Nelson, G. W., & Perelson, A. S. (1992). A mechanism of immune escape by slow-replicating hiv strains. *JAIDS Journal of Acquired Immune Deficiency Syndromes*, *5*(1), 82–93.

NIAID. (2015). *Understanding the immune system: How it works.* `http://www.nsta.org/publications/press/extras/files/debatable/theimmunesystem.pdf`. ([Online; accessed 1-April -2015])

Parker, K. C., Bednarek, M. A., & Coligan, J. E. (1994). Scheme for ranking potential hla-a2 binding peptides based on independent binding of individual peptide side-chains. *The Journal of Immunology*, *152*(1), 163–175.

Percus, J. K., Percus, O. E., & Perelson, A. S. (1993). Predicting the size of the t-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination. *Proceedings of the National Academy of Sciences*, *90*(5), 1691–1695.

Perelson, A. S., & DeLisi, C. (1980). Receptor clustering on a cell surface. i. theory of receptor cross-linking by ligands bearing two chemically identical functional groups. *Mathematical Biosciences*, *48*(1), 71–110.

Perelson, A. S., Kirschner, D. E., & De Boer, R. (1993). Dynamics of hiv infection of cd4+ t cells. *Mathematical biosciences*, *114*(1), 81–125.

Perelson, A. S., & Weisbuch, G. (1997). Immunology for physicists. *Reviews of modern physics*, *69*(4), 1219.

Qi, Q., Liu, Y., Cheng, Y., Glanville, J., Zhang, D., Lee, J.-Y., ... Goronzy, J. J. (2014). Diversity and clonal selection in the human t-cell repertoire. *Proceedings of the National Academy of Sciences*, *111*(36), 13139–13144.

Quince, C., Lanzén, A., Curtis, T. P., Davenport, R. J., Hall, N., Head, I. M., ... Sloan, W. T. (2009). Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature methods*, *6*(9), 639–641.

Quince, C., Lanzen, A., Davenport, R. J., & Turnbaugh, P. J. (2011). Removing noise from pyrosequenced amplicons. *BMC bioinformatics*, *12*(1), 38.

Reeder, J., & Knight, R. (2010). Rapidly denoising pyrosequencing amplicon reads by exploiting rank-abundance distributions. *Nature methods*, *7*(9), 668–669.

Riedel, S. (2005). Edward jenner and the history of smallpox and vaccination. *Proceedings (Baylor University. Medical Center)*, *18*(1), 21.

Robins, H. S., Campregher, P. V., Srivastava, S. K., Wacher, A., Turtle, C. J., Kahsai, O., ... Carlson, C. S. (2009). Comprehensive assessment of t-cell receptor β-chain diversity in $\alpha\beta$ t cells. *Blood*, *114*(19), 4099–4107.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140.

Robinson, M. D., & Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, *23*(21), 2881–2887.

Rodbard, D. (1973). Mathematics of hormone-receptor interaction. In *Receptors for reproductive hormones* (pp. 289–326). Springer.

Rosen, M. J., Callahan, B. J., Fisher, D. S., & Holmes, S. P. (2012). Denoising pcr-amplified metagenome data. *BMC bioinformatics*, *13*(1), 283.

Rosenfeld, R., Vajda, S., & DeLisi, C. (1995). Flexible docking and design. *Annual review of biophysics and biomolecular structure*, *24*(1), 677–700.

Ross, R. (1911). The prevention of malaria, with addendum on the theory of happenings. *Murray, London*.

Sanger, F., Coulson, A., Friedmann, T., Air, G., Barrell, B., Brown, N., ... Smith, M. (1978). The nucleotide sequence of bacteriophage φx174. *Journal of molecular biology*, *125*(2), 225–246.

Sette, A., & Rappuoli, R. (2010). Reverse vaccinology: developing vaccines in the era of genomics. *Immunity*, *33*(4), 530–541.

Sette, A., Sidney, J., Oseroff, C., del Guercio, M.-F., Southwood, S., Arrhenius, T., ... Grey, H. M. (1993). Hla dr4w4-binding motifs illustrate the biochemical basis of degeneracy and specificity in peptide-dr interactions. *The Journal of Immunology*, *151*(6), 3163–3170.

Stenberg, M., & Nygren, H. (1988). Kinetics of antigen-antibody reactions at solid-liquid interfaces. *Journal of immunological methods*, *113*(1), 3–15.

Stigler, S. M. (1983). Who discovered bayes's theorem? *The American Statistician*, *37*(4a), 290–296.

sumanas, i. (2015). *high-throughput sequencing.* `http://www.sumanasinc.com/webcontent/animations/content/highthroughput2.html`. ([Online; accessed 1-April -2015])

Thomas-Vaslin, V., Altes, H. K., de Boer, R. J., & Klatzmann, D. (2008). Comprehensive assessment and mathematical modeling of t cell popu-

lation dynamics and homeostasis. *The Journal of Immunology*, *180*(4), 2240–2250.

Venables, W., & Ripley, B. (2002). *Modern applied statistics using s*. Springer, New York, NY, USA,.

Wang, L., Feng, Z., Wang, X., Wang, X., & Zhang, X. (2010). Degseq: an r package for identifying differentially expressed genes from rna-seq data. *Bioinformatics*, *26*(1), 136–138.

Warren, R. L., Freeman, J. D., Zeng, T., Choe, G., Munro, S., Moore, R., ... Holt, R. A. (2011). Exhaustive t-cell repertoire sequencing of human peripheral blood samples reveals signatures of antigen selection and a directly measured repertoire size of at least 1 million clonotypes. *Genome research*, *21*(5), 790–797.

Wikipedia. (2015a). *Antibody*. `https://en.wikipedia.org/wiki/Antibody`. ([Online; accessed 12-May-2015])

Wikipedia. (2015b). *Computatinal Immunology*. `http://en.wikipedia.org/wiki/Computational_immunology`. ([Online; accessed 19-April -2015])

Wikipedia. (2015c). *Dna sequencing*. `http://en.wikipedia.org/wiki/DNA_sequencing`. ([Online; accessed 12-May-2015])

Wikipedia. (2015d). *Likelihood*. `http://en.wikipedia.org/wiki/Likelihood_function`. ([Online; Accessed April 2015])

Wikipedia. (2015e). *OTU*. `https://en.wikipedia.org/wiki/Operational_taxonomic_unit`. ([Online; accessed 1-April -2015])

Wikipedia. (2015f). *Polymarase Chain Reaction*. `http://en.wikipedia.org/wiki/Polymerase_chain_reaction`. ([Online; Accessed 2015-03-19])

Wikipedia. (2015g). *Tcell receptor.* `http://en.wikipedia.org/wiki/T_cell_receptor`. ([Online; accessed 1-April -2015])

Zhang, C., Anderson, A., & DeLisi, C. (1998). Structural principles that govern the peptide-binding motifs of class i mhc molecules. *Journal of molecular biology*, *281*(5), 929–947.

Zinkernagel, R. M., & Doherty, P. C. (1997). The discovery of mhc restriction. *Immunology today*, *18*(1), 14–17.

APPENDICES

Appendix A

R code for generation of samples of different sizes using the *distr* package

```
#Loadiing the distr package
require(distr)

#initialize the distribution's parameters and its desired support
N <- 10; p <- 0.001; f <- 0.3; s <- seq(0,1e6,1)

#probabiity  of the clonotype's copy number conditioned on parameters N,f
prb <- function(s,N,f,p){
  j <- 0:min(c(s,N))
  return (sum(exp(lgamma(N+s-j+1)- lgamma(N-j+1)-lgamma(s-j+1)-lgamma(j+1)
                  j*log(f)+(N-j)*log(1-f)-(N+s-j)*log(p+(1-p)*f)+log(N)+
}
#define the distribution
mydist <- r(DiscreteDistribution (supp=s,prob= sapply(s,prb,N,f,p)))

#generate n=100 random numbers.
set.seed(77)

x <- r(mydist)(100)

#Obtaining the sample means for x,x^2 and x^3.
meanx <- mean(x)
```

```
meanx2 <- mean(x^2)

meanx3 <- mean(x^3)


#Estimating N from the samples

#quadratic is of the form aN^2+c=0

a <- -.5*meanx^3-7/2*meanx+3/2*meanx2^2/meanx-meanx3

c <- .5*meanx^3

w <- -c/a

estimate_N <- sqrt(w)


#Estimating f from the samples

estimate_f <- (estimate_N*(meanx^2+meanx-meanx2)+meanx^2)/(a*e*meanx)


#Estimating p from the samples

estimate_p <- estimate_N*estimate_f/meanx


#Estimating p with f constant at f=0.3

estimate_p_f <-  estimate_N*.3/meanx


estimate_N;estimate_f; estimate_p; estimate_p_f
```

Appendix B

Metropolis Hastings algorithm implemented in R

```
#Metrapolis Hastings algrothim


#Initialising the chain

x_0 <- 15

x_b <- x_0
```

```
#vector containing the generated chains
f <- c(x_b)

#loop for generating 1000 states x_t
for(i in 2:1000){
  #generating two independent U(0,1) random numbers in step 1
  h <- runif(2,0,1)
  u1 <- h[1]
  u2 <- h[2]
  #generating the candidate value x_c in step 2
  x_c <- x_b+1*2*(u1-1/2)
  #calculating the ratio r in step 3
  r <- exp(-.5*x_c^2)/exp(-.5*x_b^2)
  #step 4
  if(u2<r){x_a <- x_c}else{x_a <- x_b}
  #appending the generated value x_a to the vector of vaues
  f[i] <- x_a
  x_b <- x_a
}

#plotting
plot(f,type="l",xlab="",ylab = "")

#determining the auto correation
z <- acf(f, lag.max = 1, type = c("correlation"),plot = F)
```

Appendix C

R code for estimating parameters N and p of the negative binomial

```
#loading the coda package for calcuating credible intervals
library(coda)

#loading the distr package for defining probability distributions
require(distr)

#initialize the distribution's parameters and its desired support
N <- 10; p <- 0.01; s <- seq(N,1e6,1)

#define the distribution
mydist <- DiscreteDistribution (supp=s,prob =
 exp(lchoose(s-1,s-N) + N*log(p) + (s-N)*log(1-p) ))

#generate 1000 random numbers.
set.seed(45)
y <- r(mydist)(1000)

#Generation of p
getp <- function(N,p){
  alpha_0 <- 1
  beta_0 <- 1
  alpha <- length(y)*N+alpha_0
  bet <- beta_0-N*length(y)+sum(y)
  w <- rbeta(1,alpha,bet)
```

```
  return(w)
}

####################
#target distribution for N
fullcond.N <- function(N,p,lambda){
  n <- length(y)
  #log-scale transformation
  w <- -n*lgamma(exp(N))-sum(lgamma(y-
  exp(N)+1))+n*exp(N)*(log(p)-log(1-p))-
    exp(N)*log(lambda)-lgamma(exp(N)+1)
  return(w)
}
###################################
#metrapolis hastings for N
MH <- function(N_b,p,A){
  lambda <- 1
  k <- runif(2,0,1)
  u1 <- k[1]
  u2 <- k[2]
  N_c <- N_b+A*2*(u1-1/2)
  r <- exp(fullcond.N(N_c,p,lambda) - fullcond.N(N_b,p,lambda))
  if(u2<r){N_a <- N_c}else{N_a <- N_b}
  return(N_a)
}
#single step for the gibbs sampler
#theta=[p,N]
gibbstep1 <- function(theta,A){
  #updating p
```

```
  theta['p'] <- getp(theta['N'],theta['p'])
  #updating N
  d <- MH(log(theta['N']),theta['p'],A)
  theta['N'] <- exp(d)
  return(theta)
}

#gibbs sampler
gibbstepn <- function(init.va,n,A){
  h <- list()
  theta <- c(p=init.va[1],N=init.va[2])
  for(i in 1:n){
    theta <-  gibbstep1(theta,A)
    h[[i]] <- theta
  }
  return(h)
}
############################################################

############################################################

#running the gibbs sampler for 110000 iterations with
 tuning parameter A=0.035
d <- gibbstepn(c(0.05,8),110000,0.035)
g <- unlist(d)

#Obtaining MCMC for N
N <- g[seq(0,220000,2)]
#thinning N
```

```
N <- N[seq(100,110000,100)]
plot(N,type="l")
z <- acf(N, lag.max = 200,type = c("correlation"),plot = TRUE)
mean(N)

#Obtaining MCMC for p
p <- g[seq(1,220000,2)]
#thinning p
p <- p[seq(101,110000,100)]
plot(p,type="l")
z <- acf(p, lag.max = 200,type = c("correlation"),plot = TRUE)
mean(p)
#################################################################
#
#
#################################################################################
#getting lag 1 autocorrelation and mean
A <- seq(0.01,.06,by=0.001)
#lag_1p <- c()
lag_1N <- c()
meanp <- c()
meanN <- c()
for(j in 1:length(A)){
  d <- gibbstepn(c(0.001,10),5000,A[j])
  g <- unlist(d)
  ####################
  #obtaining samples for N
  ####################
  N <- g[seq(0,10000,2)]
```

```
  #1000 burn in
  N <- N[1001:5000]
  #obtaining mean of N
  meanN[j] <- mean(N)
  #Obtaining 1 lag for N
  z1 <- acf(N, lag.max = 200,type = c("correlation"),plot = FALSE)
  lag_1N[j] <- z1[1]$acf[1]
  ########################
  #obtaining samples for p
  ########################
  p <- g[seq(1,10000,2)]
  #1000 burn in
  p <- p[1001:5000]
  #obtaining mean for p
  meanp[j] <- mean(p)
}

#obtaining A with the least 1 lag autocorrelatiion
A[which.min(lag_1N)]



#obtaining 1 lag plots for N and p
plot (c(0.01,0.06),c(0.92,1.05),type="n", # sets the x
 and y axes scales

      xlab="A",ylab="1 lag autocorrelation",
      main = "lag 1 as a function of the stepsize for N") # adds titles to

lines(A,lag_1N,col="black",lwd=2) # adds a curve for N
```

```
legend("topright", # places a legend at the appropriate place
       c("N"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate
       symbols (lines)

       lwd=c(2,2),col=c("black")) # gives the legend lines the correct col
################################################
################################################

#obtaining mean and 95% credible interval for different sample sizes.

MEANN <- c()
MEANp <- c()

CIN <- list()
CIp <- list()
for(i in 1:10){
  set.seed(45)
  y <- r(mydist)(i*100)
  d <- gibbstepn(c(0.05,8),1000,0.035)
  g <- unlist(d)
  #Obtaining mcmc for N
  N <- g[seq(0,2000,2)]
  #thinning N
  #N <- N[seq(100,110000,100)]
  MEANN[i] <- mean(N)
  CIN[[i]] <- HPDinterval(mcmc(N), 0.95)[seq(1,2,1)]
```

```
  #Obtaining mcmc for p
  p <- g[seq(1,2000,2)]
  #thinning p
  #p <- p[seq(101,110000,100)]
  MEANp[i] <- mean(p)
  CIp[[i]] <- HPDinterval(mcmc(p), 0.95)[seq(1,2,1)]
}
#########################################################




df <- data.frame(sampe_size =seq(100,1000,100),
                 F =MEANp,
                 L =unlist(CIp)[seq(1,20,2)],
                 U =unlist(CIp)[seq(0,20,2)])


ggplot(df, aes(x = sampe_size, y = F)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymax = U, ymin = L))+
  xlab("sampe size") +
  ylab("mean of p")
```

Appendix D

R code for estimating N, f and p of simulated values from the generated probability distribution in S24

```
#Summation function in the probability equation
diCalc <- function(a,N,f,p){
```

```
  j <- 0:min(c(a,N))
  return(sum(exp(lgamma(N+a-j+1)-lgamma(N-j+1)-
  lgamma(j+1)-lgamma(a-j+1)+j*(log(p+(1-p)*f)
  -log((1-p)*(1-f)))-log(N+a-j))))
}

#full condition for f
fcf <- function(N,f,p,x){
  #summation
  b <- sapply(x,diCalc,N,f,p)

  #multiplying by thr prior
  theta0 <- 1;gamma0 <- 1
  fc <- (theta0 +sum(x)-1)*log(f)+(gamma0+length(x)*N-1)*log(1-
  f)-(length(x)*N+sum(x))*log(p+(1-p)*f)+sum(log(b))
  return(fc)
}

#full condition for p
fcp <- function(N,f,p,x){
  #summation
  b <- sapply(x,diCalc,N,f,p)
  #multiplying by thr prior
  alpha0 <- 1;beta0 <- 1
  fc <- (alpha0 +length(x)*N-1)*log(p)+(beta0+sum(x)-1)*log(1-
  p)-(length(x)*N+sum(x))*log(p+(1-p)*f)+sum(log(b))
  return(fc)
}
#full condition for N
```

```
fcN <- function(N,f,p,x){
  #summation
  b <- sapply(x,diCalc,N,f,p)
  #multiplying by thr prior
  fc <-  length(x)*log(N)+N*length(x)*(log(p)+log(1-f)-
  log(p+(1-p)*f))+sum(log(b))
  return(fc)
}

#metrapolis hastings for f
MHf <- function(N,f_b,p,x,A1){
  u <- runif(1,0,1)
  f_c <- f_b+A1*2*(u-1/2)
  r <- exp(fcf(N,f_c,p,x)-fcf(N,f_b,p,x)+dbeta(f_b,A1*f_c,A1*(1-f_c),
  log=TRUE)-dbeta(f_c,A1*f_b,A1*(1-f_b),log=TRUE))
    if(u<r){f_a <- f_c}else{f_a <- f_b}
  return(f_a)
}

#metrapolis hastings for N
MHN <- function(N_b,f,p,x,A1){
  k <- runif(2,0,1)
  u1 <- k[1]
  u2 <- k[2]
  N_c <- exp(log(N_b)+A1*2*(u1-1/2))
  r <- exp(fcN(N_c,f,p,x)-fcN(N_b,f,p,x))
  if(u2<r){N_a <- N_c}else{N_a <- N_b}
  return(N_a)
}
```

```
#metrapolis hastings for p
MHp <- function(N,f,p_b,x,A1){
  u <- runif(1,0,1)
  p_c <- rbeta(1,A1*p_b,A1*(1-p_b))
  r <- exp(fcp(N,f,p_c,x)-fcp(N,f,p_b,x)+dbeta(p_b,A1*p_c,A1*(1-p_c),
  log=TRUE)
  -dbeta(p_c,A1*p_b,A1*(1-p_b),log=TRUE))
  if(u<r){p_a <- p_c}else{p_a <- p_b}
  return(p_a)
}

N_b <- 1;p_b <- .0017;f_b <- 0.25;h <- list()
ptm <- proc.time()
for(i in 1:5000){
  N_b <- MHN(N_b,f_b,p_b,x,0.09)
  p_b <- MHp(N_b,f_b,p_b,x,2e5)
  f_b <- MHf(N_b,f_b,p_b,x,0.05)
  h[[i]] <- c(N_b,f_b,p_b)
}
#unlisting g
g <- unlist(h)
proc.time() - ptm

#Obtaining mcmc for N
Na <- g[seq(1,15000,3)]
plot(Na,type="l", xlab = "t",ylab = "N", main = "History plot of N")
z <- acf(Na, lag.max = 500,type = c("correlation"),plot
 = TRUE, main = "Autocorrelation plot for N")
```

```
mean(Na)

#Obtaining mcmc for p
pa <- g[seq(3,15000,3)]
plot(pa,type="l", xlab = "t",ylab = "p", main = "History plot of p")
z <- acf(pa, lag.max = 300,type = c("correlation"),plot
 = TRUE, main = "Autocorrelation plot for p")
mean(pa)

#Obtaining mcmc for f
fa <- g[seq(2,15000,3)]
plot(fa,type="l", xlab = "t",ylab = "f", main = "History plot of f")
z <- acf(fa, lag.max = 300,type = c("correlation"),plot
 = TRUE, main = "Autocorrelation plot for f")
mean(fa)


#Inference of p and N with f=0.3 (contsant)
N_b <- 3;p_b <- .0012;f_b <- 0.3;h <- list()
ptm <- proc.time()
for(i in 1:5000){
  N_b <- MHN(N_b,f_b,p_b,k,0.06)
  p_b <- MHp(N_b,f_b,p_b,k,3e6)
  h[[i]] <- c(N_b,f_b,p_b)
}
g <- unlist(h)
proc.time() - ptm

#Obtaining mcmc for N
```

```
Na <- g[seq(1,15000,3)]
#Na<- Na[seq(1,1000,60)]
plot(Na,type="l", xlab = "t",ylab = "N", main =
 "History plot of N with f=0.3")
z <- acf(Na, lag.max = 500,type = c("correlation"),plot
 = TRUE, main = "Autocorrelation plot for N with f=0.3")
mean(Na)

#Obtaining mcmc for p
pa <- g[seq(3,15000,3)]
plot(pa,type="l", xlab = "t",ylab = "p", main =
 "History plot of p with f=0.3")
z <- acf(pa, lag.max = 500,type = c("correlation"),plot
 = TRUE, main = "Autocorrelation plot for p with f=0.3")
mean(pa)
#############################################################################

#############################################################################
#obtaining mean and 95% credible interval for different sample sizes.

#single step Gibbs sampler
gibbstep1 <- function(theta,x,A){
  #updating N
  theta['N'] <- MHN(theta['N'],theta['f'] ,theta['p'],x,A[1])
  #updating p
  theta['p'] <- MHp(theta['N'],theta['f'] ,theta['p'],x,A[2])
  #updating f
  #to be commented out when f is constant
  theta['f'] <- MHf(theta['N'],theta['f'] ,theta['p'],x,A[3])
```

```
  return(theta)
}

#multiple step Gibbs sampler.
gibbstepn <- function(init.va,n,x,A){
  h <- list()
  theta <- c(N=init.va[1],p=init.va[2],f=init.va[3])
  h[[1]] <- theta
  for(i in 1:n){
    theta <-  gibbstep1(theta,x,A)
    h[[i+1]] <- theta
  }
  return(h)
}

#Require necessary packages
require(coda)

require(distr)

# the probability distribution of the clonotype cooy number
prb <- function(s,N,f,p){
  j <- 0:min(c(s,N))
  return (sum(exp(lgamma(N+s-j+1)- lgamma(N-j+1)-lgamma(s-j+1)-
  lgamma(j+1)+(s-j)*log(1-p)+(s-j)*log(f)+
                    j*log(f)+(N-j)*log(1-f)-(N+s-
                    j)*log(p+(1-p)*f)+log(N)+N*log(p)-log(N+s-j))))
}
```

```
#simuating data using given parameter values for N,,p and f
N <- 10;p <- 0.001;f <- 0.3;
s <- seq(0,5e6,1)
x <- r(DiscreteDistribution (supp=s,prob= sapply(s,prb,N,f,p)))(1000)

#lists t contain mean and 95% credible interva values for each of the 3 pa
MEANN <- c()
MEANp <- c()
MEANf <- c()

CIN <- list()
CIp <- list()
CIf <- list()

for(i in 1:10){
  y <- x[1:(i*100)]
  d <- gibbstepn(c(10,0.001,0.3),3000,y,c(0.06,3e6,0.05))
  g <- unlist(d)
  #Obtaining mcmc for N
  N <- g[seq(1,9000,3)]
  #extracting the mean and credible interval
  MEANN[i] <- mean(N)
  CIN[[i]] <- HPDinterval(mcmc(N), 0.95)[seq(1,2,1)]

  #Obtaining mcmc for p
  p <- g[seq(2,9000,3)]
  #extracting the mean and credible interval
  MEANp[i] <- mean(p)
  CIp[[i]] <- HPDinterval(mcmc(p), 0.95)[seq(1,2,1)]
```

```
  #Obtaining mcmc for f
  f <- g[seq(3,9000,3)]
  #extracting the mean and credible interval
  MEANf[i] <- mean(f)
  CIf[[i]] <- HPDinterval(mcmc(f), 0.95)[seq(1,2,1)]
}
#########################################################



df <- data.frame(sample_size =seq(100,1000,100),
                 F =MEANN,
                 L =unlist(CIN)[seq(1,20,2)],
                 U =unlist(CIN)[seq(0,20,2)])

ggplot(df, aes(x = sample_size, y = F)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymax = U, ymin = L))+
  xlab("sample size") +
  ylab("mean of N")

df <- data.frame(sampe_size =seq(100,1000,100),
                 F =MEANp,
                 L =unlist(CIp)[seq(1,20,2)],
                 U =unlist(CIp)[seq(0,20,2)])

ggplot(df, aes(x = sampe_size, y = F)) +
  geom_point(size = 4) +
```

```
  geom_errorbar(aes(ymax = U, ymin = L))+
  xlab("sampe size") +
  ylab("mean of p")

df <- data.frame(sample_size =seq(100,1000,100),
                 F =MEANf,
                 L =unlist(CIf)[seq(1,20,2)],
                 U =unlist(CIf)[seq(0,20,2)])

ggplot(df, aes(x = sample_size, y = F)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymax = U, ymin = L))+
  xlab("sample size") +
  ylab("mean of f")
```