University of Louisville
ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2017

Likelihood-based methods for analysis of copy number variation using next generation sequencing data.

Udika Iroshini Bandara
University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Part of the Biostatistics Commons

Recommended Citation

Bandara, Udika Iroshini, "Likelihood-based methods for analysis of copy number variation using next generation sequencing data." (2017). *Electronic Theses and Dissertations.* Paper 2778.
https://doi.org/10.18297/etd/2778

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

LIKELIHOOD-BASED METHODS FOR ANALYSIS OF COPY NUMBER VARIATION USING NEXT GENERATION SEQUENCING DATA

By

Udika Iroshini Bandara
BSc., University of Sri Jayewardenepura, Sri Lanka, 2006
M.A., University of Louisville, Kentucky, USA, 2014

A Dissertation
Submitted to the Faculty of the
College of Arts and Sciences of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in
Industrial and Applied Mathematics

Department of Mathematics
University of Louisville
Louisville, KY

August 2017

LIKELIHOOD-BASED METHODS FOR ANALYSIS OF COPY NUMBER VARIATION USING NEXT GENERATION SEQUENCING DATA

Submitted by

Udika Iroshini Bandara

A Dissertation Approved on

July 21, 2017

by the following Dissertation Committee:

Dr. Ryan Gill, Dissertation Director

Dr. Jiaxu Li

Dr. Riten Mitra

Dr. Prasanna Sahoo

Dr. Cristina Tone

DEDICATION

This dissertation is dedicated
to my mother Prema Bandara, who has been helping me in numerous ways to
succeed in this challenging task,
my deceased father Herath Bandara, my husband Kasun Fernando
and
my loving son Kemith Fernando.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Dr. Ryan Gill for the great support that I received related to my research, for his understanding, patience, motivation, immense knowledge and for all the guidance that he gave me through his valuable comments.

I would also like to thank the rest of my thesis committee, Dr. Riten Mitra, Dr. Cristina Tone, Dr. Jiaxu Li, and deceased Prof. Prasanna Sahoo, for their insightful comments and advice.

My deepest gratitude also goes to my friends who are like my family, and who have been helping me in countless ways to make this succeed.

Finally, I am grateful to my family, my mother, my loving sisters, and my brother for their great support. Also, I would like to express my sincere thanks to my husband and my loving son, for bearing all the difficulties and helping me to succeed in this thesis and my life in general.

ABSTRACT

LIKELIHOOD-BASED METHODS FOR ANALYSIS OF COPY NUMBER VARIATION USING NEXT GENERATION SEQUENCING DATA

Udika Iroshini Bandara

July 21, 2017

A Copy Number Variation (CNV) detection problem is considered using Circular Binary Segmentation (CBS) procedures, including newly developed procedures based on likelihood ratio tests with the parametric bootstrap for models based on discrete distributions for count data (Poisson and negative binomial) and a widely-used `DNAcopy` package. Results from the literature concerning maximum likelihood estimation for the negative binomial distribution are reviewed. The Newton-Raphson method is used to find the root of the derivative of the profile log likelihood function when applicable, and it is proven that this method converges to the true MLE, if the starting point for the Newton-Raphson is selected appropriately and the MLE exists. Simulation studies are conducted to examine the performance of the CBS procedures under various scenarios. Also, the procedures are applied to a real data example based on the baboon endogenous viral genome.

TABLE OF CONTENTS

CHAPTER

1. INTRODUCTION 1

2. BACKGROUND 7

  2.1 What is DNA? 8

    2.1.1 Directionality 9

    2.1.2 Chemical Bases 10

  2.2 Next Generation Sequencing 12

    2.2.1 Illumina Sequencing Technology 13

  2.3 BWA - Burrows-Wheeler Aligner 16

  2.4 Copy Number Variation 17

3. NEGATIVE BINOMIAL DISTRIBUTION 19

  3.1 Probability Mass Function 20

  3.2 Maximum Likelihood Estimation 21

  3.3 Newton-Raphson Method 38

4. CNV DETECTION METHODS 45

  4.1 Likelihood Based Methods 46

    4.1.1 Change Point Analysis 46

    4.1.2 Maximum Likelihood Estimation 48

    4.1.3 Statistical Distributions 49

    4.1.4 MLEs for the Negative Binomial Model 50

    4.1.5 MLEs for the Poisson Model 55

  4.2 Circular Binary Segmentation 60

        4.2.1 Likelihood Ratio Test with Parametric Bootstrap . . . 62
    4.3 CBS Procedures for CNV Detection and Estimation 62
        4.3.1 CBS Procedures 63
        4.3.2 Simulations Under Poisson Models 63
        4.3.3 Simulations Under Negative Binomial Models 69
5. COMPARISON OF METHODS 78
    5.1 MetaSim 79
        5.1.1 Importing Genome Sequences to the Database 80
        5.1.2 Simulation of Illumina Short Reads 82
        5.1.3 Aligning Reads to the Reference Genome 84
    5.2 Copy Number Analysis of MetaSim Data 85
6. CONCLUSIONS 93
    6.1 Conclussion 93
    6.2 Discussion: Advantages and Drawbacks 95
    6.3 Future Work 98
REFERENCES 100
APPENDIX 110
    A.1 MetaSim Output for Illumina Short Reads 110
        A.1.1 Simulator Log (.mprf file) 110
        A.1.2 Fasta File 111
    A.2 BWA output 112
    A.3 R Code 115
CURRICULUM VITAE 129

LIST OF TABLES

Table 4.1. Results for simulation under Poisson scenario *pN1*. 64

Table 4.2. Results for simulation under Poisson scenario *pH1*. 65

Table 4.3. Results for simulation under Poisson scenario *pF1*. 66

Table 4.4. Results for simulation under Poisson scenario *pN2*. 66

Table 4.5. Results for simulation under Poisson scenario *pH2*. 67

Table 4.6. Results for simulation under Poisson scenario *pF2*. 68

Table 4.7. Results for simulation under negative binomial scenario *nbN1*. 70

Table 4.8. Results for simulation under negative binomial scenario *nbH1*. 70

Table 4.9. Results for simulation under negative binomial scenario *nbF1*. . 71

Table 4.10. Results for simulation under negative binomial scenario *nbN2*. 72

Table 4.11. Results for simulation under negative binomial scenario *nbH2*. 73

Table 4.12. Results for simulation under negative binomial scenario *nbF2*. . 74

Table 4.13. Results for simulation under negative binomial scenario *nbN3*. 74

Table 4.14. Results for simulation under negative binomial scenario *nbH3*. 75

Table 4.15. Results for simulation under negative binomial scenario *nbF3*. . 76

Table 5.1. Model estimates for MetaSim simulated reads based on *DNAcopy*. 87

Table 5.2. Model estimates for aggregated reads based on *Poisson CBS*. . 88

Table 5.3. Model estimates for MetaSim simulated reads based on *Poisson CBS*. 89

Table 5.4. Model estimates for aggregated reads based on *Negative Binomial CBS*. 90

Table 5.5. Model estimates for MetaSim simulated reads based on *Negative Binomial CBS*. 91

LIST OF FIGURES

Figure 1.1. Copy Number Variations 2
Figure 2.1. Structure of DNA. 9
Figure 2.2. DNA directionality. 10
Figure 2.3. Nitrogenous Bases 11
Figure 2.4. Library Preparation. 14
Figure 2.5. Cluster Amplification. 14
Figure 2.6. Sequencing 15
Figure 2.7. Alignment and Data Analysis 15
Figure 2.8. Duplication and Deletion. 18
Figure 3.1. Graph of $f(r)$ for an example in the case when $\bar{x} > \hat{\sigma}^2$. 26
Figure 3.2. Graph of $f(r)$ for an example in the case when $\bar{x} = \hat{\sigma}^2$. 27
Figure 3.3. Graph of $f(r)$ for an example in the case when $\bar{x} < \hat{\sigma}^2$. 28
Figure 3.4. Illustration of a convex function with a root at r^*. 39
Figure 4.1. Change point model. 48
Figure 4.2. Change point model corresponding to the negative binomial distribution. 50
Figure 4.3. Change point model corresponding to the Poisson distribution. 56
Figure 5.1. The screenshot of baboon endogenous virus in the GUI mode. 80
Figure 5.2. Baboon endogenous virus strain M7 proviral DNA. 81
Figure 5.3. Linear baboon endogenous virus tandem duplication strain M7 proviral DNA. 82
Figure 5.4. Screenshot of fasta file with short reads. 84

Figure 5.5. Screenshot of sam file with aligned reads. 85
Figure 5.6. Number of reads starting at each genomic position. 86
Figure 5.7. Fitted mean values for the *DNAcopy*, *Poisson CBS*, and *Negative Binomial CBS* methods. 92
Figure A.1. Reads Simulator Log 110
Figure A.2. Screenshot of fasta file with short reads. 112
Figure A.3. Screenshot of BWA and samtools commands in Bio-linux terminal. 113
Figure A.4. Screenshot of sam file with aligned reads. 114

CHAPTER 1
INTRODUCTION

In 2002, Charles Lee, a cytogeneticist at Brigham and Women's Hospital in Boston, was trying to identify the changes in chromosomes in individuals, who were previously characterized with chromosomal imbalance. So, he used normal healthy individuals as the control group, but he was repeatedly unsuccessful in these experiments, because he found major aberrations in the gene sequence of normal healthy people. What was more confusing to him was that, some of these individuals in the control group carried more copies than the individuals in the experimental group, but they were perfectly healthy people. In late 2003, he met Steve Scherer, a Canadian scientist who studies genetic variation in human disease at the Hospital for Sick Children. He was also experiencing the same weird phenomenon in normal healthy patients. They collaborated their research together to measure the number of occurrences of the large scale copy number variations across the human genome. To investigate these copy number variants, they applied array-based comparative genomic hybridization (arrayCGH) to the genomes of all unrelated individuals. Meanwhile, Michael Wigler, a molecular geneticist at Cold Spring Harbor Laboratory in New York, was also observing major variations in chromosomes in healthy normal people, using a different technology (Check 2005). In 2004, both sets of researchers published their findings that indicated large-scale variations in copy number contributing substantially to genomic variation between normal humans and susceptibility to some genetic diseases and both sets of researchers argued for a more dynamic human genome structure (Iafrate et al. 2004, Sebat et al. 2004, Lobo 2008). These large variations in DNA segments are called copy number variants and they are described as a DNA segment of one kilo base(kb) or larger that is present at a variable copy number in comparison with a reference genome (Redon et al. 2006). Hence, year 2004 was a landmark year of genetic studies, because of this new discovery related to the human genome, which is now leading researchers to believe that copy number variations (CNVs) are as important a component of genomic diversity as single nucleotide polymorphisms (SNPs – a variation in a single nucleotide that occurs at a specific position in the genome differs between members of a species) (Lobo 2008). These copy number variants can be seen in the forms illustrated in Figure 1.1.

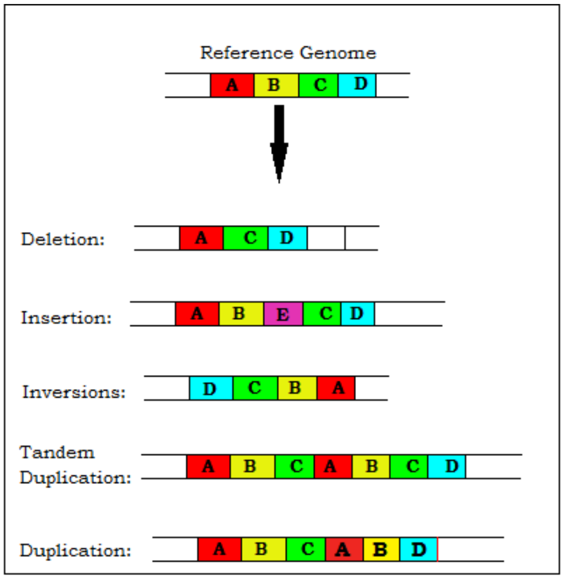


Figure 1.1: Copy Number Variations

With the huge development of DNA sequencing technologies, scientists have more resources to increase research on genetic variations in the human population as well as in other mammalian species. Because of the major breakthrough of finding copy number variations (CNVs) in 2004, scientists began to link CNVs in the genome to human health and complex diseases. Not all the CNVs have influence on diseases. It is found that some CNVs in healthy population have no apparent influence on phenotype, for example, Enrique Gonzalez, a researcher at Veterans Administration Research Center for AIDS and HIV-1 Infection, and his colleagues found that individuals who have extra copies of CCL3L1 can protect an individual against contracting HIV and developing acquired immunodeficiency syndrome (AIDS) more effectively than individuals, who carried fewer copy number variants encoding CCL3L1 than average, were significantly more susceptible to HIV and AIDS (Gonzalez et al. 2005). That means, if an individual with extra copies became infected by HIV, the individual will have very slow progress towards full-blown AIDS (Check 2005). Similarly, other copy number variants carried by healthy individuals that seem to have no function might actually be evolutionarily retained in populations if they provide a selective advantage (Lobo 2008). There are as many as 40 other CNVs which have been definitely linked with complex diseases. Also, scientists found some evidence that whether CNVs have a detectable phenotypic effect might be influenced by interaction with additional genetic or environmental factors (Clancy 2008).

Discovering genes in which copy number is associated with diseases such as cancer has the potential to provide diagnostic tools for these diseases. For instance, overexpression of the ERBB2 gene has been associated with certain types of breast cancer (Pollack et al. 1999). Moreover, more aggressive forms of breast cancer are correlated with having a high number of copies of the ERBB2 gene (Peiró et al. 2004). CNVs have been detected in genetic regions related to Alzheimer's disease and schizophrenia (Freeman et al. 2006; Redon et al. 2006). Prader-Willi syndrome and Angelman syndrome have been connected with the imprinted chromosome 15 region (Redon et al. 2006). Down's syndrome is known to occur when there are three copies of chromosome 21 (see Hattori et al. 2000 and the references therein). Spinal muscle atrophy and DiGeorge syndrome have been connected with CNVs in chromosome 22 (Redon et al. 2006).

With all these great discoveries, a new era has begun in gene science, and overall, in the medical science field. Researchers have more opportunities to find hidden secrets behind human genomes, such as copy number variation, and essential facts about our evolution. Scientists now can screen patients with genetic diseases and compare them with healthy patients in a control group to attempt to discover which CNVs are actually associated with disease and which are instead common in the overall population. Consequently, this could provide new discoveries of previous unknown relationships between genes and diseases (Lobo 2008). Many scientists have been applying statistical approaches to develop new statistical tools to identify the copy number variations more efficiently and accurately. Therefore, it is worthwhile to look at this aspect with a statistical eye. Throughout this work, we explore this aspect in a statistical way and will apply the likelihood based methods to identify the copy number variation (tandem duplication region) in a viral genome. We will illustrate the flow of our work as follows.

Before analyzing any problem in general, it is worthwhile to explore the background information and the past literature. We present this in Chapter 2, in a way such that, it begins with discussing deoxyribonucleic acid (DNA) and its shape, the structure of a DNA including directionality and nitrogenous bases as background information regarding DNA, and also, we will briefly present the information regarding DNA sequencing, essentially more on Next Generation Sequencing and its process. Finally, we will give information on the Burrows-Wheeler Aligner, an alignment software tool, and concepts related to copy number variation, more on tandem duplication, and deletion.

Many researchers have been using statistical distributions for analyzing CNV detection problems. It is more natural to use discrete distributions, for modeling the counts of short DNA sequences (these are called as read counts) mapping to a long genome sequence. Under the assumption that reads are randomly and independently sampled from any location of the target genome with equal probability, it is often assumed that the distribution of the count reads that map into a specific location of the reference genome can be approximated by a Poisson distribution. However, some authors revealed that read counts generated by some instrumental equipments (such as Illumina Genome Analyzer), follow a Poisson distribution with a slight overdispersion (Bentley et al. 2008, Yoon et al. 2009). Alberto Magi and his colleagues showed clear evidence that the read counts generated by high throughput sequencing technologies can be modeled by a negative binomial distribution (Magi et al. 2012). Therefore, in Chapter 3, we will describe one form of the negative binomial distribution, which is the Poisson-gamma mixture, in detail. There we present the maximum likelihood estimates (MLE) of negative binomial parameters and review the results concerning the existence and the uniqueness of the MLE from past literature. In particular, we carefully reformulate the important results from Simonsen (1976). Then, we extend these results to make new statements about the shape of the profile likelihood function for the negative binomial distribution. Also, we describe the Newton-Raphson method and apply it to find the roots of the derivative of the log likelihood function when applicable. Moreover, we use our results about the shape of the profile likelihood function to prove that, the algorithm will definitely converge to true MLE, if the starting point for the Newton-Raphson is selected appropriately and the MLE exists.

Statisticians and bioinformaticians have been widely using change point analysis for inventing computational tools for detecting CNVs. In Chapter 4, we start by considering the problem of detecting CNVs with a simple change point model, which has two changes, and applying the likelihood based methods to estimate the MLEs for two cases (Poisson and negative binomial distributions). This includes estimating the means of each of the continuous segments, and the change point locations. Next, we extend our consideration to more than two changes by discussing a well known algorithm called Circular Binary Segmentation (CBS). We describe a widely-used R package `DNAcopy` package (Seshan and Olshen 2017) which uses the CBS algorithm, and we develop CBS procedures using the likelihood ratio tests based on the Poisson and negative binomial models and describe a parametric bootstrap procedure for making decisions on whether to reject hypotheses at each step. Finally, we perform simulation studies under various scenarios to examine the performance of these CBS procedures.

Chapter 5 presents a comparison of the methods that we discussed in Chapter 4, for real data, by considering the baboon endogenous virus strain M7 proviral DNA as the reference genome. We generate Illumina short reads from the test genome, which is created by adding a tandem duplication region to the reference genome, using a reads simulator called MetaSim. There we simulate Illumina short reads, and each is 36 bases long, using the empirical error model. Assuming that the locations of the simulated reads are unknown, we use the BWA aligner to attempt to align the reads to the reference genome and then apply the CBS procedures to analyze the resulting read counts for copy number variation.

Chapter 6 describes some conclusions, discussion and the future work related to the current research. Discussion and output from the MetaSim and code used for finding MLEs and performing the CBS procedure are provided in the Appendix.

CHAPTER 2
BACKGROUND

The young Swiss doctor Friedrich Miescher, who was working in the laboratory of Felix Hoppe-Seyler at the University of Tbingen in the winter of 1868-1869, performed experiments on the chemical composition of leukocytes that lead to the discovery of DNA (Dahm 2008). Leslie Pray (2008) mentioned that 1869 was a landmark year in genetic research, because of this enormous discovery of Miescher, and now we continue to make great strides in understanding the human genome and the importance of DNA to life and health.

The middle of the twentieth century was a great period of some of the most fundamental discoveries in DNA research (Dahm 2008). In 1944, Avery and his colleagues were the first ones who identified DNA as genetic material (Avery et al. 1944). At the end of this decade, Erwin Chargaff and his group studied the composition and structure of nucleic acids and discovered that the base composition of DNA varies between species (Chargaff et al. 1949, Chargaff 1950, Chargaff 1951).

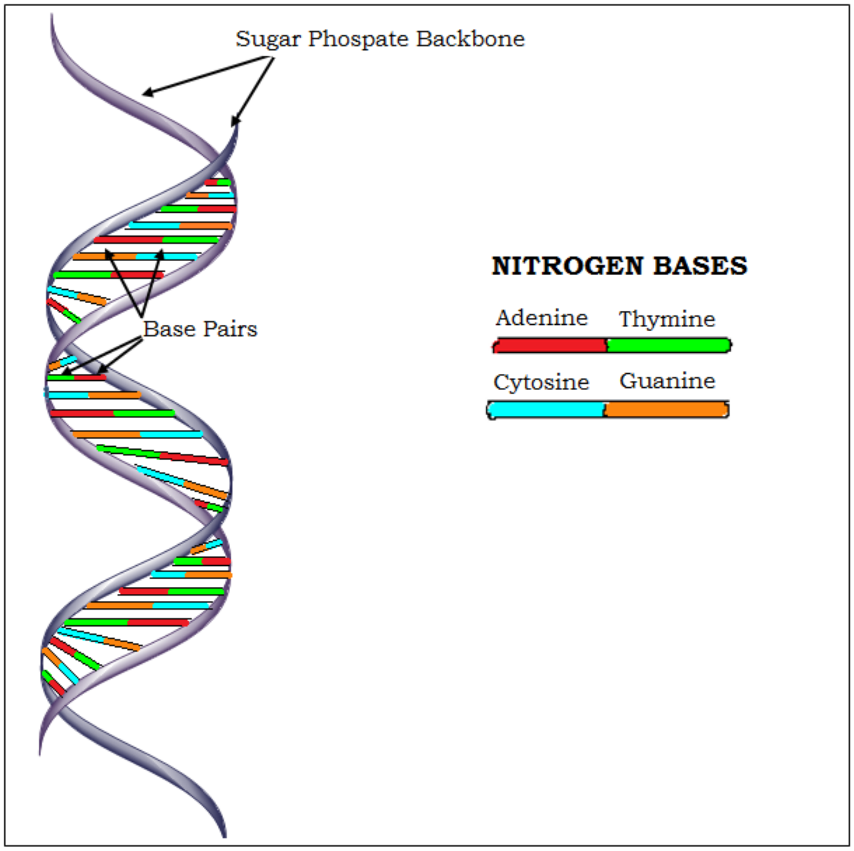
For the first time in the history, in 1953, James Watson and Francis Crick discovered the double helix, the twisted-ladder structure of DNA, which is now accepted as the first correct double-helix model of the DNA (Watson and Crick 1953). This was a huge milestone in the history of genetics and inspired the modern molecular biology, and also a great help to understand the concepts behind the genetic code and protein synthesis. These ground-breaking discoveries helped to build new technologies, such as genetic engineering, rapid gene sequencing, etc., which are today's multi million dollar bio-technology industries.

2.1 What is DNA?

DNA (deoxyribonucleic acid) is the genetic material passed from generation to generation in all organisms. Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people. It is a complex doubly stranded molecule present in the nucleus of all living cells. DNA is often referred to as the "building block of life" (Mitra et al. 2014).

Albrecht Kossel, a German biochemist and pioneer in the study of genetics, was the first one who was able to isolate and name the five nucleobases, which are adenine [A], guanine [G], cytosine [C], thymine [T], and uracil [U] (Jones 1953). It is now widely accepted that DNA contains only four chemical bases called adenine [A], guanine [G], cytosine [C], and thymine [T] (Pray 2008). The DNA sequence is essentially a collection of these nitrogen bases. Those nitrogenous bases are the places, where all the biological information about a living organism are stored in. The order, or sequence, of these bases determines the information to make proteins.

DNA bases pair up with each other, A with T and C with G, inside the helix, and these units are called base pairs. Each base is also attached to a sugar molecule (5- carbon sugar called deoxyribose) and a phosphate molecule. Together, a base, a 5-carbon sugar, and a phosphate molecule are called a nucleotide. The nucleotides are joined to one another in a chain by molecule bonds,which is a chemical bond that involves the sharing of electron pairs between atoms, between the sugar of one nucleotide and the phosphate of the next, resulting in an alternating sugar-phosphate backbone. Nucleotides are arranged in two long strands that form a spiral, called a double helix. The structure of the double helix is somewhat like a ladder, with the base pairs resembling the steps of the ladder. The structure of the DNA is shown in the Figure 2.1.

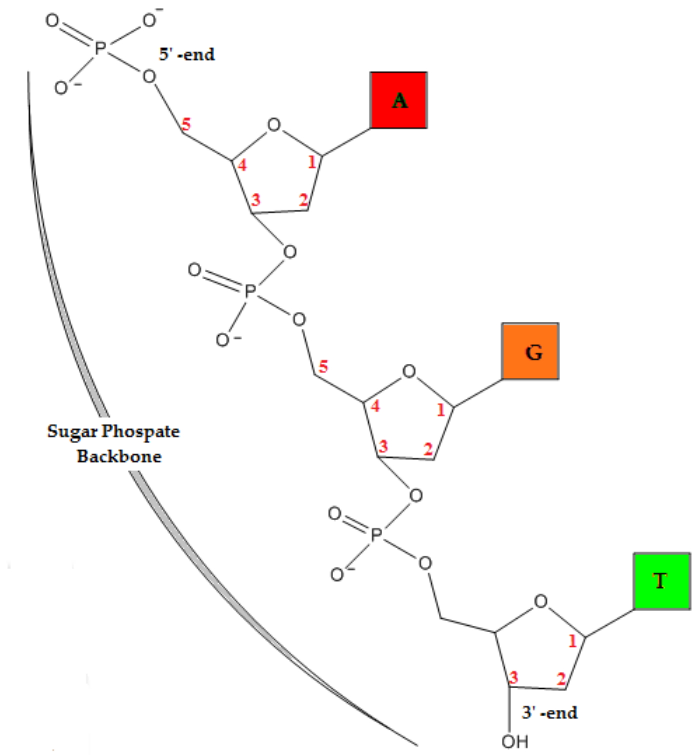


This picture is redrawn from the Genetic Home reference-U.S. National Library of Medicine.

Figure 2.1: Structure of DNA.

2.1.1 Directionality

Each strand in the backbone is associated with a direction from top to bottom. This direction is determined by the ending and starting carbons. The directions are commonly referred to as 5'- end or 3'- end. The 5' or 3' indicate the carbon numbers in the DNA sugar backbone. The 5'-carbon is attached to a phosphate group and the 3'-carbon is attached to a hydroxyl group. As shown in the Figure 2.2, five carbons in the sugar-phosphate backbone are numbered starting from the "o" in the clockwise direction. So, the carbon which has the base attached is called "1" and the next carbon is "2" and so on.



This picture is adapted from Directionality(molecular biology) Wikipedia.

Figure 2.2: DNA directionality.

2.1.2 Chemical Bases

Chargaff (1950) concluded that the amount of adenine [A] is usually similar to the amount of thymine [T], and the amount of guanine [G] usually approximates the amount of cytosine [C]. In other words, the total amount of purines [A + G] and the total amount of pyrimidines [C + T] are usually nearly equal, which is now known as “Chargaff’s rule ”. But he could not imagine the explanation of these relationships, specifically, that A is bound to T and C is bound to G within the molecular structure of DNA. Watson and Crick (Watson and Crick 1953) discovered that, “A” fit together perfectly with “T” and “C” with “G”, with each pair held together by hydrogen bonds (Pray 2008). That is, A forms 2 hydrogen bonds with T on the opposite strand and G forms 3 hydrogen bonds with C. This is because purines (A and G) always bind with pyrimidines (T and C). Due to this relationship, the sequence of bases on one strand uniquely determined the bases on the opposite strand (Watson and Crick 1953). The length of a DNA fragment is generally determined by the number of base pairs it has in kBp (kilo base pairs) or mBp (mega base pairs). The following images in Figure 2.3 show the structures of the four nitrogenous bases.

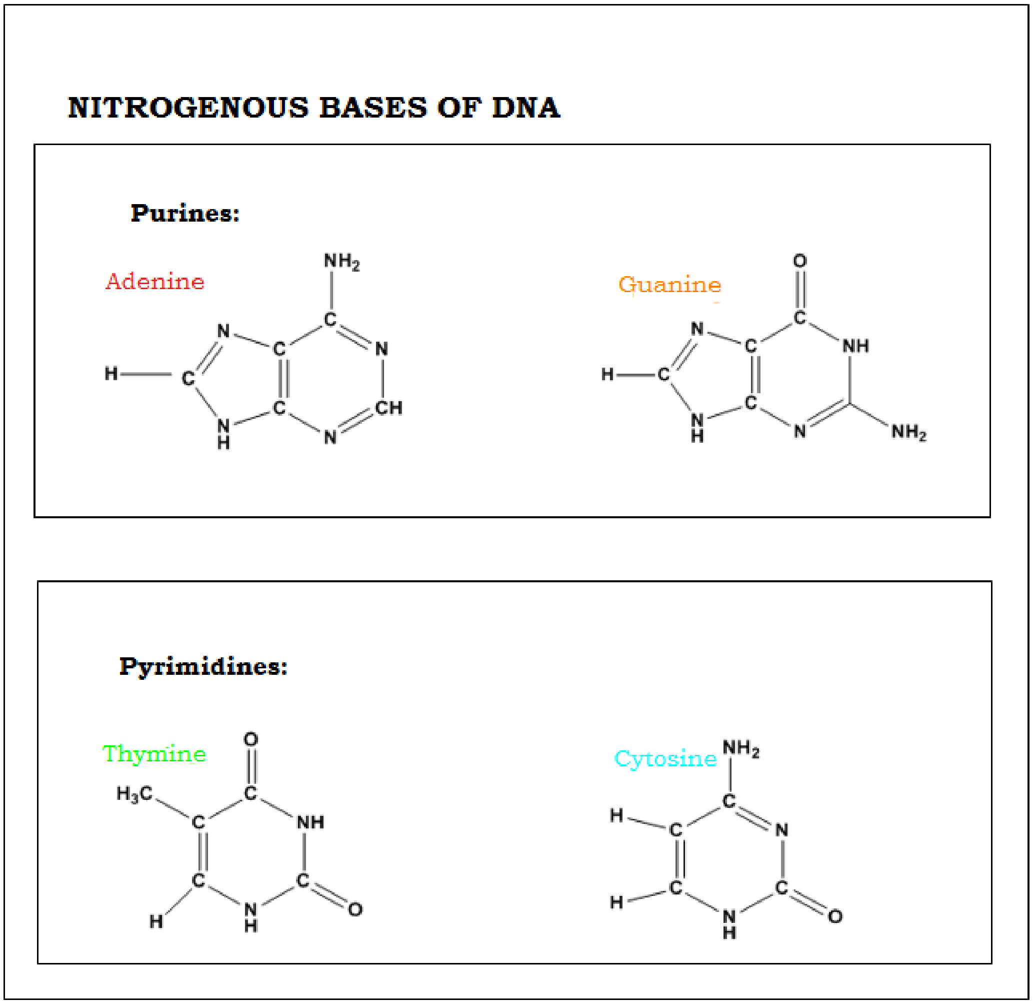


Figure 2.3: Nitrogenous Bases

After the great discoveries of many secrets behind DNA, scientists began to look for the DNA sequencing technologies, which is the process of determining the precise order of nucleotides within a DNA molecule. The first method for determining DNA sequences was found by Ray Wu at Cornell University in 1970 (*DNA sequencing wikipedia*). In 1977, Frederick Sanger, a British biochemist, introduced the "Sanger Method" which was a major breakthrough and allowed long stretches of DNA to be rapidly and accurately sequenced. This DNA sequencing information has been widely used in the bio-medical field, in a great deal to identify and diagnose various kinds of genetic diseases, such as Down syndrome, cancers etc.(Machado and Menck 1997). With the exceptional development of biological and medical research, the demand of having a fast and accurate DNA sequencing has risen. So, much research has been administered to discover fast, easy and accurate DNA sequencing technologies. Nowadays, the next generation sequencing technology is currently meeting this demand in an enormous way.

2.2 Next Generation Sequencing

Next-generation sequencing (NGS), also known as high-throughput sequencing, is the general term used to describe a variety of modern sequencing technologies. NGS technologies provide a sensitive and accurate alternative approach for accessing genomic variations. The quality, speed and affordability give NGS a significant advantage over the other DNA sequencing technologies.(Hurd and Nelson 2009, Su et al. 2011, Wang et al. 2014). In the past few years, because of the rapid development in NGS technology, many sequencing platforms have been released. Some of these include:

- Roche 454 sequencing
- Ion torrent: Proton / PGM sequencing
- SOLiD sequencing

- Illumina HiSeq
- Illumina MiSeq

These relatively new technologies enable DNA and RNA to be sequenced more rapidly than the older DNA sequencing (Sanger sequencing). Also, NGS is much less expensive and the price has continued to decrease as the technology has been further developed. Basically, the NGS process includes a combination of template preparation, sequencing and imaging, and genome alignment and assembling (Metzker 2010).

2.2.1 Illumina Sequencing Technology

The NGS technologies using the Illumina platform employ a massively parallel Sequencing by Synthesis (SBS) methodology which involves sequencing the ends of millions, or even billions, of DNA fragments (called reads) in parallel and performing read assembly for analysis (Chaitankar et al. 2016). Illumina NGS workflows consist of 4 basic steps. They are:

- Library Preparation
- Cluster Amplification
- Sequencing
- Allignment and Data Analysis

Fgirues 2.4, 2.5, 2.6, and 2.7 present pictorial illustrations of the above steps. All the images are adapted from An Introduction to Next Generation Sequencing Technology (`www.illumina.com/technology/next-generation-sequencing.html`).

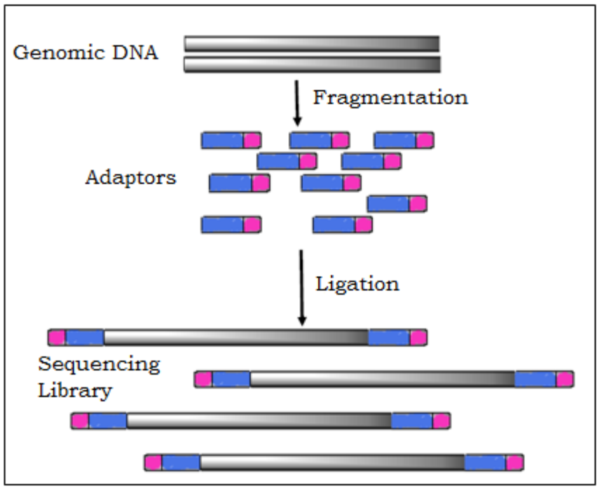


Figure 2.4: Library Preparation.

Library preparation begins with the extraction and purification of genomic DNA. The extracted DNA is then broken into several overlapping fragments followed by 5' and 3' adapter ligation as illustrated in Figure 2.4. Adapter-ligated fragments are then PCR amplified and gel purified.

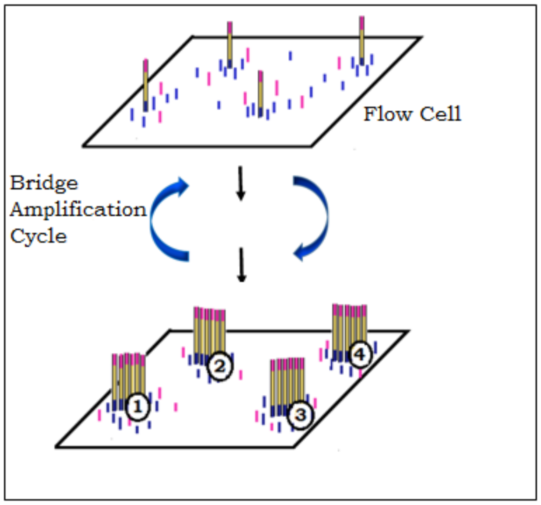


Figure 2.5: Cluster Amplification.

The library is loaded into a flow cell and each fragment is then amplified into distinct colonal clusters through bridge amplification. This generates thousands to millions of copies of a particular DNA sequence as illustrated in Figure 2.5.

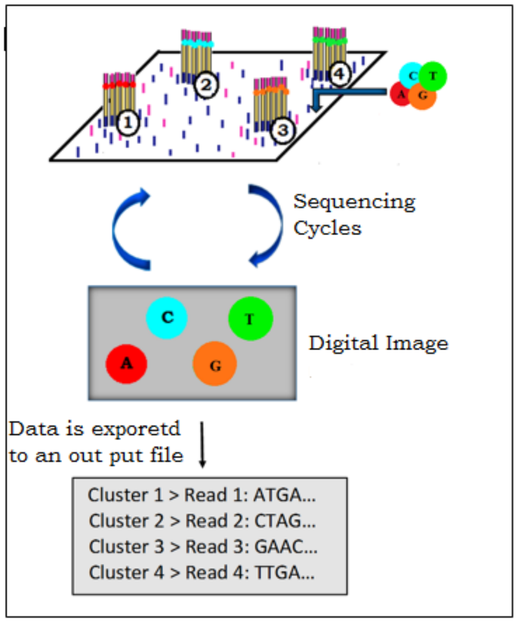

1
2
3
4
C
T
A
G
Sequencing Cycles
C
T
A
G
Digital Image
Data is exporetd to an out put file
Cluster 1 > Read 1: ATGA...
Cluster 2 > Read 2: CTAG...
Cluster 3 > Read 3: GAAC...
Cluster 4 > Read 4: TTGA...

Figure 2.6: Sequencing

In the sequencing process, the base pairs from the ends of the fragments are read. Each DNA strand within a cluster incorporates one of the nucleotides. This nucleotide is the same for all strands within a single cycle. In this process, non- incorporated molecules are washed away. A detecting device then records the fluorescent color corresponding to the sequenced base as illustrated in Figure 2.6.

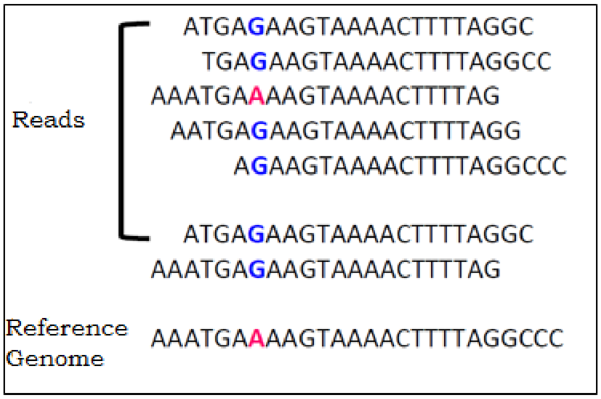


Figure 2.7: Alignment and Data Analysis

Finally, reads are aligned to a reference sequence with bioinformatics software. Figure 2.7 illustrates an example of a few reads being aligned to a reference genome. After alignment, differences between the reference genome and the newly sequenced reads can be identified.

With the fast development of DNA sequencing technologies, an enormous amount of reads can be generated. As a result of that, the necessity of fast and accurate read alignment software tools is required. In the past few decades, many read alignment software tools, such as the Burrows-Wheeler Alignment (BWA), Bowtie, AlignerBoost etc., have been developed and massively used in the bioinformatics field. Since we use BWA for our simulation study, we will illustrate it briefly in the following section.

2.3 BWA - Burrows-Wheeler Aligner

In 2009, Heng Li and Richard Durbin introduced a read alignment software package called the Burrows-Wheeler Alignment(BWA) tool (Li and Durbin 2009). The BWA tool is a new read alignment software package that is based on backward search with Burrows-Wheeler Transform (BWT). It aligns short sequencing reads against a large reference sequence such as the human genome, allowing mismatches and gaps. It consists of three algorithms:

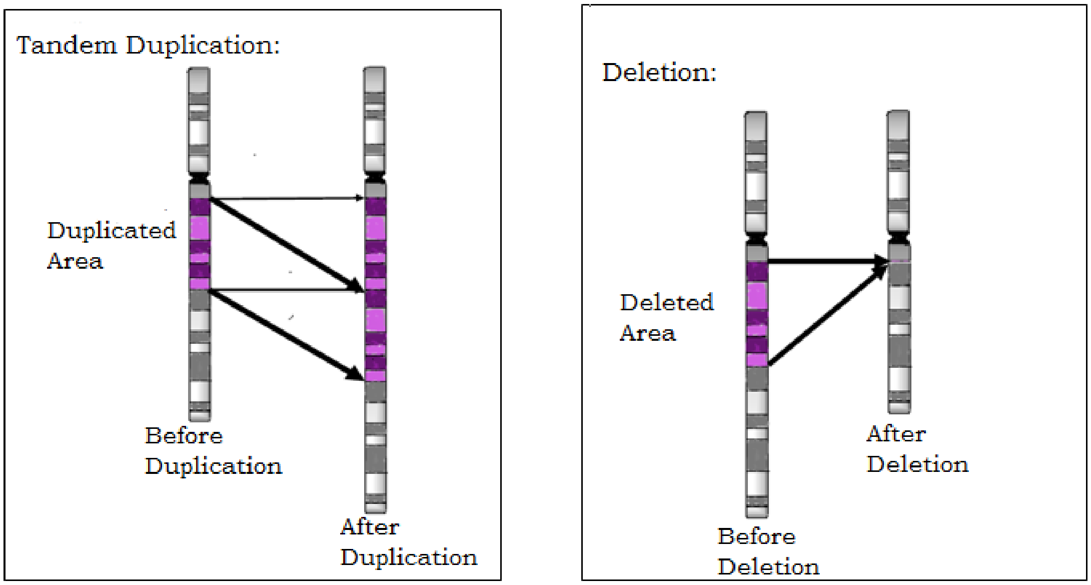
- BWA-backtrack (Li and Durbin 2009),
- BWA-SW (Li and Durbin 2010), and
- BWA-MEM (Li 2013).

BWA-backtrack is designed for Illumina sequence reads up to 100bp, while the BWA-MEM and BWA-SW for longer sequences ranged from 70bp to 1Mbp. BWA-MEM and BWA-SW share similar features such as long-read support and split alignment, but BWA-MEM (which is the latest) is generally recommended for high-quality queries as it is faster and more accurate. BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.

With the vast development of NGS technology, the term “Copy Number Variation (CNV)” is often associated with the genomic and medical fields, since it plays an important role in studies of susceptibility or resistance to complex diseases. CNVs can be found in both humans and other species. It is an unique identity for gene family expansion and diversification, which is a crucial evolutionary force.

2.4 Copy Number Variation

“ Copy Number Variation ” is often defined by the situation in which sections of the genome are repeated and the number of repeats in the genome varies between individuals in the human population. It is a form of structural variation in the genome. Copy number variation is often associated with complex disorders such as Down’s syndrome which is the duplication of part or all of chromosome 21, Schizophrenia (St. Clair 2009), Autism (Pinto et al. 2010), cancers (Shlien et al. 2009), etc. In many cases, the term CNV refers to the duplication or deletion of DNA segments larger than 1000 base pairs. These changes in the DNA segments of a gene may disturb its own activity level. For example, if a DNA segment of a gene is deleted, the cell may produce half as much protein as its normal activity level. Figure 2.8 shows the situation in which duplication or deletion occur in a gene. The duplicate regions can be located adjacent to each other (called tandem duplication) or one of the duplicate regions can be in its normal location and the other in a novel location on a different part of the same chromosome or even on another chromosome (Griffiths et al. 2000).



Figures are adapted from Copy Number Variation-wikipedia

Figure 2.8: Duplication and Deletion.

CHAPTER 3
NEGATIVE BINOMIAL DISTRIBUTION

Several papers have considered maximum likelihood estimation of the parameters of a negative binomial distribution. Page 367 of Anscombe (1950) correctly conjectured conditions under which the maximum likelihood estimate exists and conditions under which it is unique. Simonsen (1976) and Simonsen (1980) gave a detailed proof of Anscombe's conjecture. Levin and Reeds (1977) gave an alternate proof based on the variation diminishing property of Laplace transforms. Wang (1996) reviewed some later attempts to prove Anscombe's conjecture, pointing out a flaw in one later paper, and proved that the unbiased estimator of the parameter which is denoted by r in this chapter does not exist. Dai et al. (2013) proposed a fixed point iteration algorithm to attempt to find the MLE and proved that the algorithm converges to the MLE when the sample mean is no greater than 1.5.

In this chapter, we describe and examine computation of maximum likelihood estimates of the parameters of the negative binomial distribution using the Newton-Raphson method. First, we describe the negative binomial distribution, derive expressions for the maximum likelihood estimate (MLE) of its parameters based on a random sample of observations for this distribution, and review results concerning the existence and uniqueness of the MLE from past literature. Then we describe the Newton-Raphson method and apply it to find the root of the derivative of the likelihood function when applicable. Finally, if the starting point for the Newton-Raphson method is selected appropriately and the MLE exists, we prove that the algorithm is guaranteed to converge to the true MLE.

3.1 Probability Mass Function

The *probability mass function* (pmf) of a *negative binomial distribution* has the form:

$$P(X = x) = \frac{\Gamma(x + r)}{x!\Gamma(r)} \quad p^x (1 - p)^r, \tag{3.1}$$

where, $x \in \{0, 1, 2, \ldots\}$, $0 < p < 1$, and $r > 0$. Often, the parameter p is referred to as the *probability of success on a Bernoulli trial* and, if r is an integer, then x is the number of successes before the rth failure, and $(x + r)$ is the number of trials.

Note that this definition of the negative binomial distribution is more general than the typical definition in an introductory probability text since r is not restricted to positive integer values. One major application of the negative binomial distribution is that it can play a role as a mixture distribution of Poisson distribution with gamma mixing weights. In other words, the negative binomial distribution can be considered as a Poisson distribution where the Poisson parameter itself is a random variable and distributed as a gamma distribution. Thus, the negative binomial distribution is known as a Poisson-gamma mixture and the number of failures r does not necessarily need to be a non-negative integer. The following derivation clarifies the intution behind this statement.

Let X be a Poisson random variable with parameter Λ and, suppose that Λ has a gamma distribution with shape parameter $\alpha = r$ and rate parameter $\beta = \frac{1-p}{p}$. Then the joint density function of X and Λ is given by

$$P(X = x|\Lambda = \lambda) \cdot P(\Lambda = \lambda) = \frac{e^{-\lambda}\lambda^x}{x!} \cdot \lambda^{r-1} \frac{e^{-\lambda(1-p)/p}}{\left(\frac{p}{1-p}\right)^r \Gamma(r)}.$$

Then, the unconditional distribution of X can be obtained as follows.

$$P(X = x) = \int_0^\infty \frac{e^{-\lambda}\lambda^x}{x!} \cdot \lambda^{r-1} \frac{e^{-\lambda(1-p)/p}}{\left(\frac{p}{1-p}\right)^r \Gamma(r)} d\lambda$$

$$\begin{aligned} P(X=x) &= \frac{(1-p)^r}{x!\cdot p^r\cdot\Gamma(r)}\int_0^\infty e^{-\lambda\left(1+\frac{1-p}{p}\right)}\cdot\lambda^{(x+r-1)}d\lambda \\ &= \frac{(1-p)^r\cdot\Gamma(x+r)\cdot p^x}{x!\cdot\Gamma(r)}\int_0^\infty \frac{e^{\frac{-\lambda}{p}}\cdot\lambda^{(x+r-1)}}{p^x\cdot p^r\cdot\Gamma(x+r)}d\lambda \\ &= \frac{(1-p)^r\cdot\Gamma(x+r)\cdot p^x}{x!\cdot\Gamma(r)}\int_0^\infty \frac{e^{\frac{-\lambda}{p}}\cdot\lambda^{(x+r-1)}}{p^{x+r}\cdot\Gamma(x+r)}d\lambda. \end{aligned}$$

Note that, the integral is equal to 1, since the integrand is the probability density function of the gamma distribution. Hence, the unconditional distribution of X is given by

$$P(X=x) = \frac{\Gamma(x+r)}{x!\cdot\Gamma(r)}\cdot p^x(1-p)^r.$$

If we simplify (3.1) further, we will get the following form of the pmf:

$$\begin{aligned} P(X=x) &= \frac{\Gamma(x+r)}{x!\Gamma(r)}\quad p^x\,(1-p)^r \\ &= \frac{(x+r-1)\cdots r\cdot\Gamma(r)}{x!\Gamma(r)}\quad p^x\,(1-p)^r \\ &= \frac{(x+r-1)\cdots r}{x!}\quad p^x\,(1-p)^r\,. \end{aligned}$$

The natural logarithm of the pmf is

$$\ln P(X=x) = x\ln p + r\ln(1-p) + \sum_{\nu=0}^{x-1}\ln(r+\nu) - \ln(x!).$$

3.2 Maximum Likelihood Estimation

Let $x_1, x_2, \ldots, x_n$ be a random sample of observations from independent and identically distributed (iid) random variables $X_1, X_2, \ldots, X_n$ which have the pmf given by (3.1). Then the log likelihood function can be written as

$$l(p,r) = \sum_{i=1}^{n}\left\{x_i\ln p + r\ln(1-p) + \sum_{\nu=0}^{x_i-1}\ln(r+\nu) - \ln(x_i!)\right\}. \tag{3.2}$$

Differentiating the log-likelihood with respect to p, we obtain

$$\frac{\partial l}{\partial p}(p,r) = \sum_{i=1}^{n} \Big\{ \frac{x_i}{p} - \frac{r}{1-p} \Big\}.$$

Solving $\partial l / \partial p = 0$ gives the maximizer for p, say $\hat{p}(r)$, as a function of r:

$$\frac{\partial l}{\partial p}(\hat{p}(r), r) = 0$$

$$\frac{nr}{1-\hat{p}(r)} = \frac{\sum_{i=1}^{n} x_i}{\hat{p}(r)}$$

$$\hat{p}(r) \cdot nr = \sum_{i=1}^{n} x_i - \hat{p}(r) \sum_{i=1}^{n} x_i$$

$$\hat{p}(r) \Big(nr + \sum_{i=1}^{n} x_i \Big) = \sum_{i=1}^{n} x_i$$

$$\hat{p}(r) = \frac{\sum_{i=1}^{n} x_i}{nr + \sum_{i=1}^{n} x_i}$$

$$\hat{p}(r) = \frac{\bar{x}}{r + \bar{x}} \tag{3.3}$$

where, $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ is the sample mean of $x_1, \ldots, x_n$. For each fixed $r > 0$ $l(p,r)$ is maximized at $\hat{p}(r)$ since $\frac{\partial^2 l}{\partial p^2}(p,r) = -\sum_{i=1}^{n} \left\{ \frac{x_i}{p^2} + \frac{r}{(1-p)^2} \right\} < 0$ for all $p \in (0,1)$.

Substituting (3.3) into (3.2), we can write the profile log-likelihood function for r as

$$l(\hat{p}(r), r) = \sum_{i=1}^{n} \Big\{ x_i \ln \hat{p}(r) + r \ln(1 - \hat{p}(r)) + \sum_{\nu=0}^{x_i - 1} \ln(r + \nu) - \ln(x_i!) \Big\}.$$

and differentiate it to obtain

$$\begin{aligned} h(r) &= \frac{\partial l(\hat{p}(r), r)}{\partial r} \\ &= \frac{\hat{p}'(r)}{\hat{p}(r)} \sum_{i=1}^{n} x_i - nr \frac{\hat{p}'(r)}{1 - \hat{p}(r)} + n \ln(1 - \hat{p}(r)) + \sum_{i=1}^{n} \left(\sum_{\nu=0}^{x_i - 1} \frac{1}{r + \nu} \right) \end{aligned}$$

$$h(r) = n \ln(1 - \hat{p}(r)) + \sum_{i=1}^{n} \left(\sum_{\nu=0}^{x_i - 1} \frac{1}{r + \nu} \right), \tag{3.4}$$

since

$$\begin{aligned}
\frac{\hat{p}'(r)}{\hat{p}(r)} \sum_{i=1}^{n} x_i - nr \frac{\hat{p}'(r)}{1 - \hat{p}(r)} &= \hat{p}'(r) \left[\frac{n\bar{x}}{\hat{p}(r)} - \frac{nr}{1 - \hat{p}(r)} \right] \\
&= n\hat{p}'(r) \left[\frac{\bar{x}}{\bar{x}/(r + \bar{x})} - \frac{r}{r/(r + \bar{x})} \right] \\
&= n\hat{p}'(r) \left[(r + \bar{x}) - (r + \bar{x}) \right] \\
&= 0.
\end{aligned}$$

By substituting equation (3.3) into equation (3.4), we get

$$\begin{aligned}
h(r) &= n \ln \left(1 - \frac{\bar{x}}{r + \bar{x}} \right) + \sum_{i=1}^{n} \left(\sum_{\nu=0}^{x_i - 1} \frac{1}{r + \nu} \right) \\
&= n \ln \left(\frac{r}{r + \bar{x}} \right) + \sum_{i=1}^{n} \left(\sum_{\nu=0}^{x_i - 1} \frac{1}{r + \nu} \right) \\
&= n \ln \left(\frac{1}{1 + \bar{x}/r} \right) + \sum_{i=1}^{n} \left(\sum_{\nu=0}^{x_i - 1} \frac{1}{r + \nu} \right) \\
&= -n \ln \left(1 + \frac{\bar{x}}{r} \right) + \sum_{i=1}^{n} \left(\sum_{\nu=0}^{x_i - 1} \frac{1}{r + \nu} \right) \\
&= \sum_{i=1}^{n} \left(\frac{1}{r} + \frac{1}{r+1} + \frac{1}{r+2} + \ldots + \frac{1}{r + x_i - 1} \right) - n \ln \left(1 + \frac{\bar{x}}{r} \right) \\
&= \sum_{i=1}^{n} \left(\sum_{\nu=1}^{x_i} \frac{1}{r + \nu - 1} \right) - n \ln \left(1 + \frac{\bar{x}}{r} \right) \\
&= \sum_{\nu=1}^{k} \left(\sum_{i=1}^{n} \frac{I(x_i \geq \nu)}{r + \nu - 1} \right) - n \ln \left(1 + \frac{\bar{x}}{r} \right) \\
&= \sum_{\nu=1}^{x_i} \left(\sum_{i=1}^{n} I(x_i \geq \nu)(r + \nu - 1)^{-1} \right) - n \ln \left(1 + \frac{\bar{x}}{r} \right)
\end{aligned}$$

$$h(r) = \sum_{\nu=1}^{k} N_\nu (r + \nu - 1)^{-1} - n \ln \left(1 + \frac{\bar{x}}{r} \right) \tag{3.5}$$

where, $k = \max(x_1, x_2, \ldots, x_n)$ and $N_\nu = \sum_{i=1}^{n} I(x_i \geq \nu)$.

Then solving the equation $h(r) = 0$ provides the maximum likelihood estimate for r. Then, letting $\hat{r}$ denote the solution, we have by (3.5)

$$h(\hat{r}) = \sum_{\nu=1}^{k} N_\nu (\hat{r} + \nu - 1)^{-1} - n \ln \left(1 + \frac{\bar{x}}{\hat{r}} \right) = 0.$$

We introduce a function $f(r)$, such that, $f(r) = \frac{1}{n} h(r)$. Then,

$$f(r) = \sum_{\nu=1}^{k} \frac{N_\nu}{n} (r + \nu - 1)^{-1} - \ln \left(1 + \frac{\bar{x}}{r} \right). \tag{3.6}$$

After some simulation studies, we observed that $f(r) = 0$ has a unique solution whenever $\hat{\sigma}^2 > \bar{x}$ and has no solution whenever $\hat{\sigma}^2 \leq \bar{x}$;
where, $\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2$.

Here are three examples of small data sets which demonstrate the possible shapes of the graph of $f(r)$ (shown in Figures 3.1, 3.2, and 3.3). The R code and functions are included for computing $f(r)$ and creating the graphs with the examples.

Case I: The sample mean $\bar{x}$ is larger than the sample variance $\hat{\sigma}^2$. To illustrate this case, suppose that we observe $x_1 = 1$, $x_2 = 4$, and $x_3 = 2$. As shown in the output from the code below, $\bar{x} = \frac{7}{3} > \frac{14}{9} = \hat{\sigma}^2$. It appears that f is a positive decreasing function from the graph of $f(r)$ shown in Figure 3.1.

```
> x=c(1, 4, 2)
> n=length(x)
> x
```

```
[1] 1 4 2
> n
[1] 3
> max(x)
[1] 4
> mean(x)
[1] 2.333333
> sigma2=sum((x-mean(x))^2)/n
> sigma2
[1] 1.555556
> calc_Nu <- function(v){
+ t = 0
+ for (i in 1:n){
+  if (x[i] >= v){
+  t = t+1
+ }
+ }
+ return(t)
+ }
> l=function(x,r.delta=.01,r.max=100,...){
+ r=seq(r.delta,r.max,by=r.delta)
+ f=rep(0,length(r))
+ for (j in 1:length(r)){
+ S=rep(0,max(x))
+ for (v in 1:max(x)){
+ S[v] = (calc_Nu(v))/(r[j]+v-1)
+ }
+ f[j]= (sum(S)/n) - log(1 +(mean(x)/r[j]))
+ }
> plot(r,f, main=expression(paste("Graph of f(r) when ",   bar(x),
+ " > ",hat(sigma)^2 )),type="l",...)
+ abline(h=0,lty=2)
+ }
> l(x,xlim=c(0,80),ylim=c(-0.0025,0.01))
```

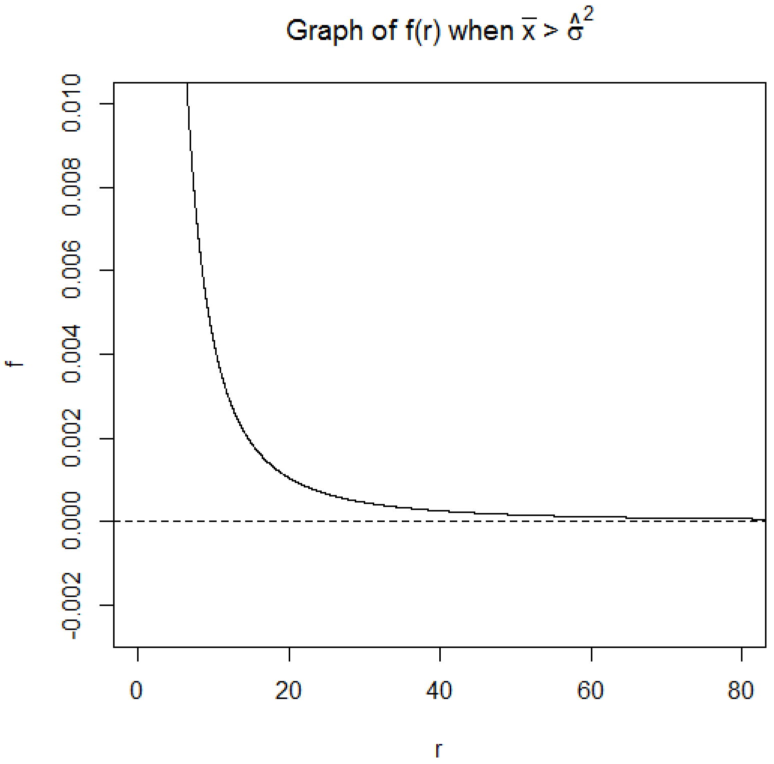


Figure 3.1: Graph of $f(r)$ for an example in the case when $\bar{x} > \hat{\sigma}^2$.

Case II: The sample mean $\bar{x}$ equals the sample variance $\hat{\sigma}^2$.

Next, we consider the example where $x_1 = 3$, $x_2 = 0$, and $x_3 = 3$. As shown in the output from the code below, $\bar{x} = 2 = \hat{\sigma}^2$. Here, we use the same R functions which are shown in Case I to calculate the function N_ν and plot the graph of $f(r)$. Just as in Case I, it appears that f is a positive decreasing function from the graph of $f(r)$ shown in Figure 3.2.

```
> x=c(3,0,3)
> n=length(x)
> x
[1] 3 0 3
> n
[1] 3
> max(x)
[1] 3
> mean(x)
[1] 2
> sigma2=sum((x-mean(x))^2)/n
> sigma2
[1] 2
```

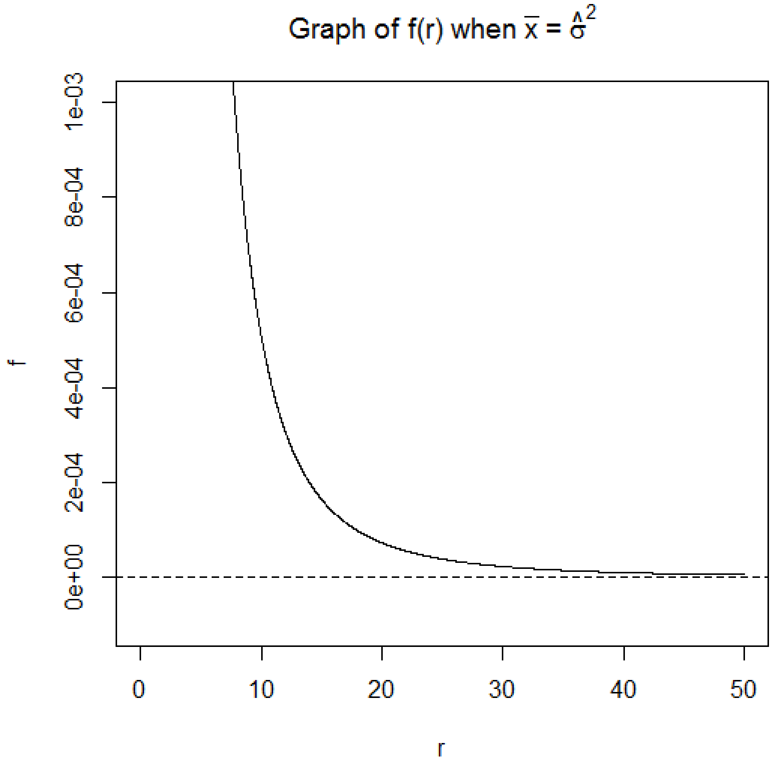


Figure 3.2: Graph of $f(r)$ for an example in the case when $\bar{x} = \hat{\sigma}^2$.

Case III: The sample mean $\bar{x}$ is less than the sample variance $\hat{\sigma}^2$.

Next, we consider the example where $x_1 = 1$, $x_2 = 2$, and $x_3 = 6$. As shown in the output from the code below, $\bar{x} = 3 < \frac{14}{3} = \hat{\sigma}^2$. Here, we use the same R functions which is shown in Case I to calculate the function N_ν and plot the graph of $f(r)$. The shape of the graph of $f(r)$ is shown in Figure 3.3. Here it appears that the non-linear equation $f(r) = 0$ has a unique solution.

```
> x=c(1, 2, 6)
> n=length(x)
> x
[1] 1 2 6
> n
[1] 3
> max(x)
[1] 6
```

```
> mean(x)
[1] 3
> sigma2=sum((x-mean(x))^2)/n
> sigma2
[1] 4.666667
```

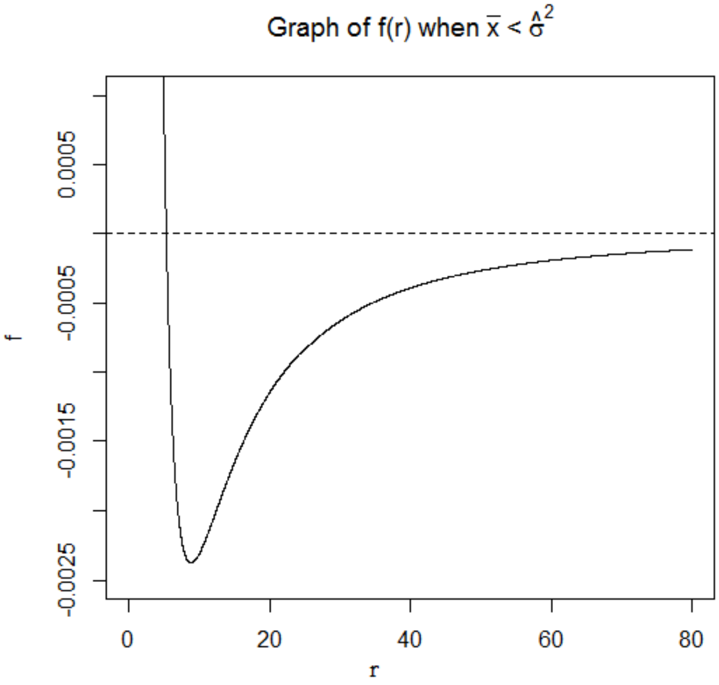


Figure 3.3: Graph of $f(r)$ for an example in the case when $\bar{x} < \hat{\sigma}^2$.

The limiting behavior of $f(r)$ as $r \to 0$ and $r \to \infty$ can be computed directly. First, we will show $\lim\limits_{r\to\infty} f(r) = 0$.

$$\begin{aligned}
\lim_{r\to\infty} f(r) &= \lim_{r\to\infty} \left\{ \sum_{\nu=1}^{k} \frac{N_\nu}{n}(r+\nu-1)^{-1} - \ln\left(1+\frac{\bar{x}}{r}\right)\right\} \\
&= \lim_{r\to\infty} \sum_{\nu=1}^{k} \frac{N_\nu}{n}(r+\nu-1)^{-1} - \lim_{r\to\infty}\left\{\ln\left(1+\frac{\bar{x}}{r}\right)\right\} \\
&= \lim_{r\to\infty} \sum_{\nu=1}^{k} \frac{N_\nu}{n}\frac{1}{(r+\nu-1)} - \lim_{r\to\infty}\left\{\ln\left(1+\frac{\bar{x}}{r}\right)\right\} \\
&= \sum_{\nu=1}^{k} \frac{N_\nu}{n}\cdot 0 - \ln(1+0) \\
&= 0.
\end{aligned}$$

Next, we will show that $\lim\limits_{r \to\ 0^+} f(r) = \infty$.

$$\begin{aligned}
\lim_{r \to\ 0^+} f(r) &= \lim_{r \to\ 0^+} \left\{ \sum_{\nu=1}^{k} \frac{N_\nu}{n} (r + \nu - 1)^{-1} - \ln\left(1 + \frac{\bar{x}}{r}\right) \right\} \\
&= \lim_{r \to\ 0^+} \left\{ \sum_{\nu=1}^{k} \frac{N_\nu}{n} \frac{1}{(r + \nu - 1)} - \ln\left(1 + \frac{\bar{x}}{r}\right) \right\} \\
&= \lim_{r \to\ 0^+} \left\{ \ln e^{\left\{ \sum_{\nu=1}^{k} \frac{N_\nu}{n} \frac{1}{(r+\nu-1)} \right\}} - \ln\left(1 + \frac{\bar{x}}{r}\right) \right\} \\
&= \lim_{r \to\ 0^+} \ln \left\{ \frac{e^{\left\{ \sum_{\nu=1}^{k} \frac{N_\nu}{n} \frac{1}{(r+\nu-1)} \right\}}}{\left(1 + \frac{\bar{x}}{r}\right)} \right\} \\
&= \ln \left\{ \lim_{r \to\ 0^+} \frac{e^{\left\{ \frac{N_1}{nr} + \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(r+C_\nu)} \right\}}}{\left(1 + \frac{\bar{x}}{r}\right)} \right\}
\end{aligned}$$

where, $C_\nu = (\nu - 1)$ for $\nu = 2, 3, \ldots, k$. Since direct substitution yields the indeterminant form $\frac{\infty}{\infty}$, using L'Hôpital's Rule, we get the following.

$$\begin{aligned}
\lim_{r \to\ 0^+} f(r) &= \ln \left\{ \lim_{r \to\ 0^+} \frac{e^{\left\{ \frac{N_1}{nr} + \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(r+C_\nu)} \right\}} \cdot \left\{ \frac{-N_1}{nr^2} - \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(r+C_\nu)^2} \right\}}{\left(\frac{-\bar{x}}{r^2}\right)} \right\} \\
&= \ln \left\{ \lim_{r \to\ 0^+} \frac{e^{\left\{ \frac{N_1}{nr} + \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(r+C_\nu)} \right\}} \cdot \frac{-1}{r^2} \cdot \left\{ \frac{N_1}{n} + \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(1+\frac{C_\nu}{r})^2} \right\}}{\left(\frac{-1}{r^2}\right)\bar{x}} \right\} \\
&= \ln \left\{ \lim_{r \to\ 0^+} \frac{e^{\left\{ \frac{N_1}{nr} + \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(r+C_\nu)} \right\}} \cdot \left\{ \frac{N_1}{n} + \sum_{\nu=2}^{k} \frac{N_\nu}{n} \frac{1}{(1+\frac{C_\nu}{r})^2} \right\}}{\bar{x}} \right\}
\end{aligned}$$

Then by applying the limits, we get the form $\ln \left\{ \frac{e^{\infty} \cdot \frac{N_1}{n}}{\bar{x}} \right\} = \ln(\infty)$, so we have

$\lim\limits_{r \to\ 0^+} f(r) = \infty$.

Simonsen (1976,1980) considered the question of solving the equation $f(r) = 0$, with different notation. There, the equation was written as

$$\sum_{s=1}^{k} \frac{N_s}{N_0} (x + s - 1)^{-1} - Log\left(1 + \frac{m}{x}\right) = 0 \tag{3.7}$$

where, $m = \dfrac{1}{N_0}\sum_{s=1}^{k} N_s$. In (3.7), the index s is used instead of index ν, variable x instead of variable r, and N_0 instead of n in (3.4). For the remainder of this chapter, we will try to avoid confusion by converting the notation from Simonsen (1976) to the notation introduced herein. The following results concerning the existence and uniqueness of a solution to (3.7) were proved in great detail in Simonsen (1976,1980).

THEOREM 3.1. (Simonsen)
Let $\mathcal{S} = \dfrac{1}{N_0}\sum_{\nu=1}^{k} N_\nu(\nu-1)$ *and* $C(r) = \dfrac{\bar{x}}{\sum_{\nu=1}^{k} N_\nu (r+\nu-1)^{-2}} - r.$

(a) If $k = 1$ *or if* $k \geq 2$ *and* $m^2 \geq 2\mathcal{S}$*, then* $f(r) > 0$ *for all* $r > 0$*.*

(b) If $k \geq 2$ *and* $m^2 < 2\mathcal{S}$*, then the following statements are true.*

(i) There is a positive number r^* *such that* $f(r) > 0$ *when* $r < r^*$, $f(r^*) = 0$, *and* $f(r) < 0$ *when* $r > r^*$*.*

(ii) There is a number $\xi > r^*$ *such that* $f'(r) < 0$ *when* $r < \xi$, $f'(\xi) = 0$ *and* $f'(r) > 0$ *when* $r > \xi$*.*

(iii) $\dfrac{1}{\bar{x}} r\,(r + C(r))\,(r+\bar{x}) f'(r) = C(r) - \bar{x}.$

(iv) $C'(r) > 0$ *for all* $r > 0$*.*

(v) $C(r) > 0$ *for all* $r > 0$*.*

Proof: For (a), see statement (i) in Section 4 of Simonsen (1976). For (b-i), see statement (ii) in Section 4 and equation (5.4) of Simonsen (1976). For (b-ii), see statement (B) in Section 5 of Simonsen (1976). Equation (b-iii) is equivalent to equation (4.3) of Simonsen (1976). The inequality (b-iv) is equivalent to inequality (3.2) which is proved in Section 3 of Simonsen (1976). Since $f(r)$ is increasing by (b-iv), the inequality (b-v) follows from (3.3) of Simonsen (1976) which states that $\lim_{r\to 0} C(r) = 0$ and $\lim_{r\to\infty} C(r) = \dfrac{2\mathcal{S}}{\bar{x}}$.□

The following result relates the quantities m and $\mathcal{S}$ with the sample mean

$\bar{x} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i$ and sample variance $\hat{\sigma}^2 = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2$.

THEOREM 3.2. *If x_i are nonnegative integers for $i = 1, \ldots, n$, then*

(a) $m = \bar{x}$

and

(b) $\hat{\sigma}^2 = 2\mathcal{S} + \bar{x} - \bar{x}^2$.

Furthermore, if $k = 0$ or $k = 1$, then $\hat{\sigma}^2 \leq \bar{x}$.

Proof: First, we see that

$$\begin{aligned}
\sum_{i=1}^{n} x_i &= \sum_{i=1}^{n}\sum_{s=1}^{k} I(x_i \geq s) \\
&= \sum_{s=1}^{k}\sum_{i=1}^{n} I(x_i \geq s) \\
&= \sum_{s=1}^{k} N_s.
\end{aligned}$$

Dividing both sides by $N_0 = n$, we obtain (a).

To prove (b), we see that

$$\begin{aligned}
\sum_{i=1}^{n} x_i^2 &= \sum_{i=1}^{n}\left(\sum_{s=1}^{k} I(x_i \geq s)\right)^2 \\
&= \sum_{i=1}^{n}\sum_{s=1}^{k}\sum_{t=1}^{k} I(x_i \geq s)I(x_i \geq t) \\
&= \sum_{i=1}^{n}\sum_{s=1}^{k}\sum_{t=1}^{k} I\left(x_i \geq \max(s,t)\right) \\
&= \sum_{i=1}^{n}\sum_{s=1}^{k}\left(\sum_{t=1}^{s} I(x_i \geq s) + \sum_{t=s+1}^{k} I(x_i \geq t)\right) \\
&= \sum_{i=1}^{n}\sum_{s=1}^{k}\left(sI(x_i \geq s) + \sum_{t=s+1}^{k} I(x_i \geq t)\right) \\
&= \sum_{s=1}^{k}\left(s\sum_{i=1}^{n} I(x_i \geq s) + \sum_{t=s+1}^{k}\sum_{i=1}^{n} I(x_i \geq t)\right) \\
&= \sum_{s=1}^{k}\left(sN_s + \sum_{t=s+1}^{k} N_t\right)
\end{aligned}$$

$$\sum_{i=1}^{n} x_i^2 = \sum_{s=1}^{k} sN_s + \sum_{s=1}^{k-1} \sum_{t=s+1}^{k} N_t. \tag{3.8}$$

Now, it is shown by induction that

$$\sum_{s=1}^{k-1} \sum_{t=s+1}^{k} N_t = \sum_{s=1}^{k} N_s(s-1) \text{ for any nonnegative } k. \tag{3.9}$$

First, it is true for $k = 2$ since $\sum_{s=1}^{1} \sum_{t=s+1}^{2} N_t = \sum_{s=1}^{2} N_s(s-1) = N_2$. (It is also trivially true for $k = 0$ and $k = 1$.) Now suppose it is true for $k = j$; that is, suppose $\sum_{s=1}^{j-1} \sum_{t=s+1}^{j} N_t = \sum_{s=1}^{j} N_s(s-1)$. Then it follows that

$$\begin{aligned}
\sum_{s=1}^{j} \sum_{t=s+1}^{j+1} N_t &= \sum_{s=1}^{j-1} \sum_{t=s+1}^{j} N_t + \sum_{s=1}^{j-1} N_{j+1} + \sum_{t=j+1}^{j+1} N_t \\
&= \sum_{s=1}^{j} N_s(s-1) + N_{j+1}(j-1) + N_{j+1} \\
&= \sum_{s=1}^{j} N_s(s-1) + N_{j+1}j \\
&= \sum_{s=1}^{j+1} N_s(s-1)
\end{aligned}$$

which proves (3.9).

Substituting (3.9) into (3.8), we get

$$\begin{aligned}
\sum_{i=1}^{n} x_i^2 &= \sum_{s=1}^{k} sN_s + \sum_{s=1}^{k} N_s(s-1) \\
&= 2\sum_{s=1}^{k} N_s(s-1) + \sum_{s=1}^{k} N_s.
\end{aligned}$$

Then it follows that

$$\begin{aligned}
\hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^{n} \Big(x_i - \bar{x}\Big)^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} x_i^2 - \bar{x}^2 \\
&= \frac{1}{n}\Big(2\sum_{s=1}^{k} N_s(s-1) + \sum_{s=1}^{k} N_s\Big) - \bar{x}^2
\end{aligned}$$

$$
\begin{aligned}
\hat{\sigma}^2 &= \frac{1}{n}\Big(2n\mathcal{S} + \sum_{i=1}^{n} x_i\Big) - \bar{x}^2 \\
&= 2\mathcal{S} + \bar{x} - \bar{x}^2
\end{aligned}
$$

which proves (b). $k = 0$ implies that $\mathcal{S} = 0$, since the term $\mathcal{S}$ does not exist, and $\bar{x} = 0$. Then, it immediately follows from (b) that

$$\hat{\sigma}^2 = \bar{x} - \bar{x}^2 = 0 = \bar{x}.$$

Since $k = 1$ implies that $\mathcal{S} = 0$, it immediately follows from (b) that

$$\hat{\sigma}^2 = \bar{x} - \bar{x}^2 \leq \bar{x}.$$

□

The following theorem proves some useful statements about the shape of $f(r)$ where it is positive in terms of k, $\bar{x}$, and $\hat{\sigma}^2$. This includes a statement about the concavity of $f(r)$ which is not discussed in Simonsen (1976).

THEOREM 3.3. (Shape of the positive portion of the profile likelihood function $f(r)$.)

If $k > 0$, then the following properties hold for $f(r)$.

(a) If $\hat{\sigma}^2 \leq \bar{x}$, then $f(r)$ is positive for all $r > 0$.

(b) If $\hat{\sigma}^2 > \bar{x}$, then $f(r) > 0$, $f'(r) < 0$, and $f''(r) > 0$ when $0 < r < r^*$ where, r^* is the positive number in Theorem 3.1(b-i).

Proof: If $k \geq 2$ and $\hat{\sigma}^2 \leq \bar{x}$, then $m^2 = \bar{x}^2 = 2\mathcal{S} + \bar{x} - \hat{\sigma}^2 \geq 2\mathcal{S}$ so Theorem 3.1(a) implies that (a) holds. If $k = 1$ (in which case, Theorem 3.2 implies that $\hat{\sigma}^2 \leq \bar{x}$), then Theorem 3.1(a) also implies that (a) holds.

If $\hat{\sigma}^2 > \bar{x}$, then $k > 1$ and $m^2 = \bar{x}^2 = 2\mathcal{S} + \bar{x} - \hat{\sigma}^2 < 2\mathcal{S}$ by Theorem 3.2. By Theorem 3.1(b) parts (i) and (ii), $f(r) > 0$ when $r < r^*$ and $f'(r) < 0$ when $r < r^*$ since $r^* < \xi$. Thus, $f(r)$ is positive and decreasing when $0 < r < r^*$. To see that

$f(r)$ is concave upward on this interval, note that

$$A(r)f'(r) = C(r) - \bar{x} \tag{3.10}$$

where,

$$A(r) = \frac{1}{\bar{x}} r(r + C(r))(r + \bar{x})$$

is a product of positive increasing functions by Theorem 3.1(b) parts (iii), (iv), and (v). As a consequence of the product rule for differentiation, the derivative of a product of positive increasing functions is positive, so $A'(r) > 0$. Differentiating both sides of (3.10), we obtain

$$A'(r)f'(r) + A(r)f''(r) = C'(r). \tag{3.11}$$

Solving (3.11) for $f''(r)$, it is seen that

$$f''(r) = \frac{C'(r) - A'(r)f'(r)}{A(r)}. \tag{3.12}$$

If $0 < r < r^*$, then $f''(r) > 0$ since $C'(r) > 0$, $A'(r) > 0$, $f'(r) < 0$, and $A(r) > 0$, and hence, f is concave upward on this interval. □

It is also important to examine the behavior of the negative binomial distribution as the parameter (r) goes to infinity, whereas the probability of success (p) goes to zero. In particular, let us parametrize the mean of the distribution (3.1) such that

$$\mu = r\frac{p}{1-p}, \quad \text{which implies that} \quad p = \frac{\mu}{\mu + r}.$$

Then, under this parametrization, the pmf for the distribution (3.1) can be expressed as

$$P(X = x) = \frac{\Gamma(x + r)}{x!\Gamma(r)} \quad \left(\frac{\mu}{\mu + r}\right)^x \left(1 - \frac{\mu}{\mu + r}\right)^r$$

$$
\begin{aligned}
P(X=x) &= \frac{\Gamma(x+r)}{x!\Gamma(r)} \quad \left(\frac{\mu}{\mu+r}\right)^x \left(\frac{r}{\mu+r}\right)^r \\
&= \frac{\Gamma(x+r)}{x!\Gamma(r)} \quad \left(\frac{\mu}{\mu+r}\right)^x \frac{1}{\left(\frac{\mu}{r}+1\right)^r} \\
&= \frac{\mu^x}{x!} \cdot \frac{\Gamma(x+r)}{\Gamma(r)(\mu+r)^x} \cdot \frac{1}{\left(\frac{\mu}{r}+1\right)^r}.
\end{aligned}
$$

Now, if we let $r \to \infty$ we will show that $\dfrac{\Gamma(x+r)}{\Gamma(r)(\mu+r)^x}$ converges to 1, and $\dfrac{1}{\left(\frac{\mu}{r}+1\right)^r}$ converges to $e^{-\mu}$.
Since

$$
\begin{aligned}
\frac{\Gamma(x+r)}{\Gamma(r)(\mu+r)^x} &= \frac{(x+r-1)!}{\Gamma(r)(\mu+r)^x} \\
&= \frac{(x+r-1)\cdots r\cdot\Gamma(r)}{\Gamma(r)(\mu+r)^x} \\
&= \frac{(x+r-1)\cdots r}{(\mu+r)^x} \\
&= \frac{r^x\cdot\left(\frac{x-1}{r}+1\right)\cdots 1}{r^x\cdot\left(\frac{\mu}{r}+1\right)},
\end{aligned}
$$

it follows that

$$
\lim_{r\to\infty} \frac{\Gamma(x+r)}{\Gamma(r)(\mu+r)^x} = \lim_{r\to\infty} \frac{\left(\frac{x-1}{r}+1\right)\cdots 1}{\left(\frac{\mu}{r}+1\right)} = 1.
$$

Next, let $y = \lim\limits_{r\to\infty} \dfrac{1}{\left(\frac{\mu}{r}+1\right)^r}$. Then, we have

$$
\begin{aligned}
\ln y &= \lim_{r\to\infty} \ln\left(\frac{1}{\left(\frac{\mu}{r}+1\right)^r}\right) \\
&= \lim_{r\to\infty} -\ln\left(\left(\frac{\mu}{r}+1\right)^r\right) \\
&= \lim_{r\to\infty} -r\cdot\ln\left(\frac{\mu}{r}+1\right) \\
&= \lim_{r\to\infty} -\frac{\ln\left(\frac{\mu}{r}+1\right)}{\frac{1}{r}}.
\end{aligned}
$$

Direct substitution yields the indeterminant form $\frac{0}{0}$. So, by applying the L'Hopital's rule, we get

$$\ln y = \lim_{r\to\infty} -\frac{\frac{1}{\left(\frac{\mu}{r}+1\right)} \cdot \left(-\frac{\mu}{r^2}\right)}{\left(\frac{-1}{r^2}\right)} = \lim_{r\to\infty} -\frac{\mu}{\left(\frac{\mu}{r}+1\right)} = -\mu,$$

which implies that $y = e^{-\mu}$. Thus, we have

$$\lim_{r\to\infty} \frac{1}{\left(\frac{\mu}{r}+1\right)^r} = e^{-\mu},$$

and hence,

$$\begin{aligned} \lim_{r\to\infty} P(X=x) &= \frac{\mu^x}{x!} \cdot 1 \cdot e^{-\mu} \\ &= \frac{\mu^x \cdot e^{-\mu}}{x!}, \end{aligned}$$

which is the pmf of a Poisson random variable with mean μ. Hence, the pmf of a negative binomial distribution with $p = \frac{\mu}{\mu+r}$ approaches the pmf of a Poisson distribution with mean μ when r is large.

This connection between the negative binomial and Poisson distributions is helpful in understanding the supremum of the likelihood function when the MLE does not exist. The following theorem discusses the existence and uniqueness of the MLE as well as the supremum/maximum of the likelihood function. Note that the supremum in part (a) is the maximum value of the likelihood function if $x_1, \ldots, x_n$ is an iid sample from a Poisson distribution, and that in the case when $x_1 = \ldots = x_n = 0$, we define $0 \ln 0$ to be 0.

THEOREM 3.4. (Maximum likelihoood estimation for the negative binomial distribution.)

(a) If $\hat{\sigma}^2 \le \bar{x}$, then there is no maximizer of (3.2) and

$$\sup_{p,r} l(p,r) = n\bar{x}\ln\bar{x} - n\bar{x} - \sum_{i=1}^{n} \ln x_i! \quad .$$

(b) If $\hat{\sigma}^2 > \bar{x}$, then the unique MLE of (p,r) is $(\hat{p},\hat{r})$ where, $\hat{p} = \dfrac{\bar{x}}{r^* + \bar{x}}$ and $\hat{r} = r^*$ with r^* being the positive number such that $f(r^*) = 0$, and $\max\limits_{p,r} l(p,r) = l(\hat{p},\hat{r})$.

Proof: If $\hat{\sigma}^2 \leq \bar{x}$, then either $k = 0$ or $k > 0$. If $k = 0$, then the supremum of $l(p, r) = nr \ln(1 - p)$ is clearly 0 by fixing r and letting $p \to 0$, but it is not attained since $p \in (0, 1)$. Otherwise, if $k > 0$ and $\hat{\sigma}^2 \leq \bar{x}$, then $f(r)$ is positive for all r by Theorem 3.3(a) and thus (3.4) is positive for all r. Hence, there is no maximizer of $l(p, r)$ since $l(\hat{p}(r), r)$ is increasing for all r, and

$$\begin{aligned}
\sup_{p,r} l(p, r) &= \sup_{r} \max_{p} l(p, r) \\
&= \sup_{r} l(\hat{p}(r), r) \\
&= \lim_{r\to\infty} l(\hat{p}(r), r)
\end{aligned}$$

$$\begin{aligned}
&= \lim_{r\to\infty} \sum_{i=1}^{n} \left\{ x_i \ln \hat{p}(r) + r \ln(1 - \hat{p}(r)) + \sum_{\nu=0}^{x_i-1} \ln(r + \nu) - \ln x_i! \right\} \\
&= \lim_{r\to\infty} \sum_{i=1}^{n} \left\{ x_i \ln \frac{\bar{x}}{r + \bar{x}} + r \ln \frac{r}{r + \bar{x}} + \sum_{\nu=0}^{x_i-1} \ln(r + \nu) - \ln x_i! \right\} \\
&= \lim_{r\to\infty} \left\{ \sum_{i=1}^{n} x_i \ln \bar{x} + nr \ln \frac{r}{r + \bar{x}} - \sum_{i=1}^{n} x_i \ln(r + \bar{x}) + \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln(r + \nu) - \ln x_i! \right\} \\
&= \lim_{r\to\infty} \left\{ n\bar{x} \ln \bar{x} + nr \ln \frac{r}{r + \bar{x}} - \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln(r + \bar{x}) + \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln(r + \nu) - \ln x_i! \right\} \\
&= \lim_{r\to\infty} \left\{ n\bar{x} \ln \bar{x} + nr \ln \frac{r}{r + \bar{x}} + \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln \frac{r + \nu}{r + \bar{x}} - \ln x_i! \right\} \\
&= n\bar{x} \ln \bar{x} - n\bar{x} - \sum_{i=1}^{n} \ln x_i!
\end{aligned}$$

since

$$\lim_{r\to\infty} r \ln \frac{r}{r + \bar{x}} = \lim_{r\to\infty} \frac{-\ln\left(1 + \frac{\bar{x}}{r}\right)}{\frac{1}{r}} = \lim_{r\to\infty} \frac{\frac{\bar{x}}{r^2}}{-\frac{1}{r^2}} = -n\bar{x}$$

and

$$\lim_{r\to\infty} \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln \frac{r + \nu}{r + \bar{x}} = \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln \left(\lim_{r\to\infty} \frac{r + \nu}{r + \bar{x}} \right) = \sum_{i=1}^{n} \sum_{\nu=0}^{x_i-1} \ln 1 = 0.$$

If $\hat{\sigma}^2 > \bar{x}$, then by Theorem 3.1(b)(i), there is a unique solution $\hat{r} = r^*$ to the equation $f(r) = 0$. Also, $f'(r^*) < 0$ by Theorem 3.1(b)(ii) so r^* maximizes the profile log-likelihood $l(\hat{p}(r), r)$. Thus, $(\hat{p}, \hat{r})$ maximizes $l(p, r)$ since $l(\hat{p}, \hat{r}) \geq l(\hat{p}(r), r) \geq l(p, r)$ for all r and p. Hence $(\hat{p}, \hat{r})$ is the unique MLE. □

3.3 Newton-Raphson Method

The Newton-Raphson method is a widely-used iterative algorithm for attempting to find the solution of an equation $\mathrm{f}(r) = 0$ using the function and its derviative. Note that in this section, we use the Roman f to denote a general function and the Italic notation f to denote the function defined in (3.6). Specifically, this method considers the first order **Taylor series** of the function $\mathrm{f}(r)$ at $\hat{r}_j$

$$\mathrm{f}(r) = \mathrm{f}(\hat{r}_j) + \mathrm{f}'(\hat{r}_j)(r - \hat{r}_j).$$

Now, solving the equation $\mathrm{f}(r) = 0$ yields

$$r = \hat{r}_j - \frac{\mathrm{f}(\hat{r}_j)}{\mathrm{f}'(\hat{r}_j)}. \tag{3.13}$$

The function

$$g(r) = r - \frac{\mathrm{f}(r)}{\mathrm{f}'(r)}$$

is called the **Newton-Raphson iteration function.** It is easy to see that $g(r) = r$, when $\mathrm{f}(r) = 0$. Thus, the Newton-Raphson iteration for finding the root of the equation $\mathrm{f}(r) = 0$ can be accomplished by finding the fixed point such that $g(r) = r$.

The Newton-Raphson method starts with an initial value $\hat{r}_0$, and updates this value using (3.13) to obtain

$$\hat{r}_{j+1} = g(\hat{r}_j) = \hat{r}_j - \frac{\mathrm{f}(\hat{r}_j)}{\mathrm{f}'(\hat{r}_j)} \tag{3.14}$$

for $j = 0, 1, 2, \ldots$. The red line in Figure 3.4 illustrates the idea behind the Newton-Raphson method; this is the tangent line to the curve $y = \mathrm{f}(r)$ at the point $(r_j, \mathrm{f}(r_j))$. The updated value $r_{j+1} = g(r_j)$ is the value r_{j+1} where the tangent line intersects the horizontal axis.

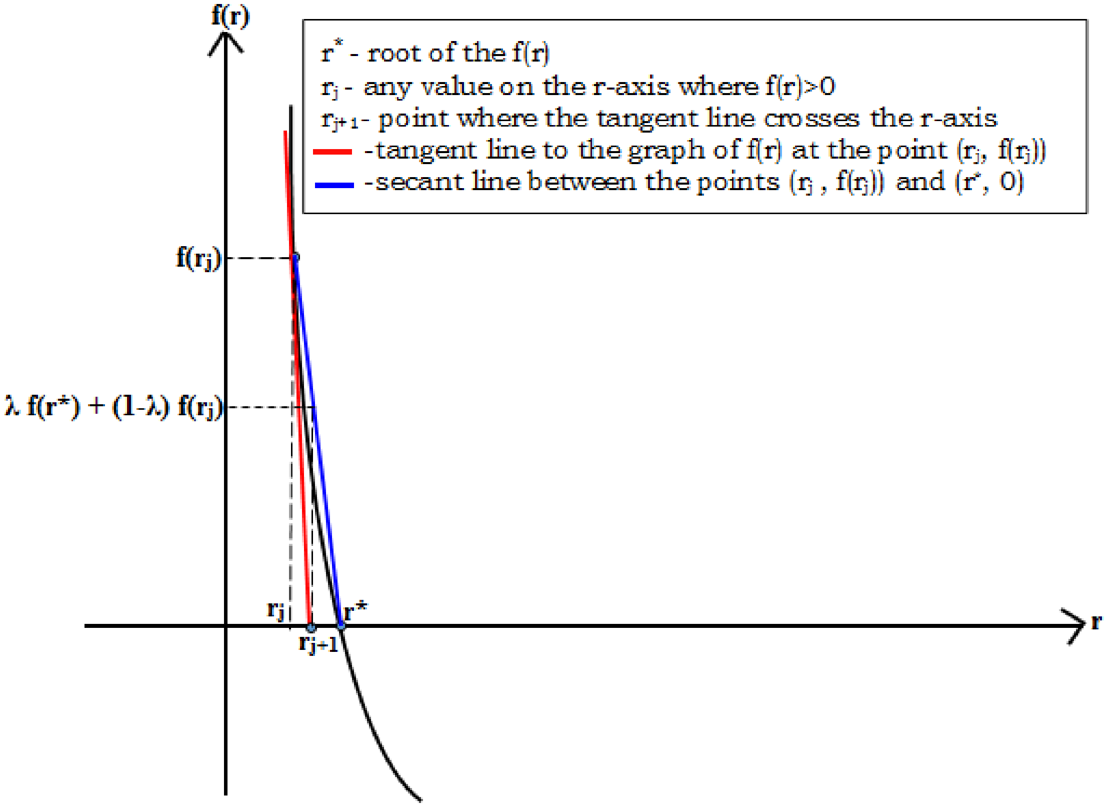

f(r)
r* - root of the f(r)
rj - any value on the r-axis where f(r)>0
rj+1 - point where the tangent line crosses the r-axis
🟥 -tangent line to the graph of f(r) at the point (rj, f(rj))
🟦 -secant line between the points (rj , f(rj)) and (r*, 0)
f(rj)
λ f(r*) + (1-λ) f(rj)
rj
r*
rj+1
r

Figure 3.4: Illustration of a convex function with a root at r^*.

In general, there is no guarantee that the Newton-Raphson method converges, but in some cases, results can be obtained based on convexity. See a general discussion of the Newton-Raphson method in, Mathews (1992), for further details.

We will need the following definition of a convex function on an interval.

Definition 3.1. *A function* f *is said to be* **convex** *on the interval* $[a, b]$ *if*

$$\mathrm{f}(\lambda x + (1-\lambda)y) \leq \lambda \mathrm{f}(x) + (1-\lambda)\mathrm{f}(y)$$

for any x *and* y *in* $[a, b]$ *and* λ *in the interval* $[0, 1]$.

The blue line in Figure 3.4 illustrates Definition 3.1; the points on the segment of the secant line between the points $(r_j, \mathrm{f}(r_j))$ and $(r^*, 0)$ are above the curve $y = \mathrm{f}(r)$.

Now, we consider a special case where a function is positive, decreasing, and convex on a certain interval. In light of Theorem 3.3, this result will be very important for studying the behavior of the Newton-Raphson method when applied to finding the solution to the likelihood equation for the negative binomial distribution.

THEOREM 3.5. *Suppose* f *is a differentiable function which is decreasing and convex on the interval* $(0, r^*]$, $\mathrm{f}(r^*) = 0$, *and* $r_0 \in (0, r^*]$. *Define the sequence*

$$r_{j+1} = r_j - \frac{\mathrm{f}(r_j)}{\mathrm{f}'(r_j)}$$

for all nonnegative integers j. *Then* $\{r_j\}_{j=0}^{\infty}$ *is a nondecreasing sequence and* $\lim_{j \to \infty} r_j = r^*$.

Proof: First, it is shown that $\{r_j\}_{j=0}^{\infty}$ is nondecreasing and bounded above by r^* by induction. By the assumption $r_0 \in (0, r^*]$, the basis step is satisfied. Now, for the inductive step, suppose $r_j \in (0, r^*]$. Since $\mathrm{f}(r_j) \geq 0$ and $\mathrm{f}'(r_j) < 0$, it is easily seen that

$$r_j \leq r_j - \frac{\mathrm{f}(r_j)}{\mathrm{f}'(r_j)} = r_{j+1}. \tag{3.15}$$

Since f is a convex function on $(0, r^*]$,

$$\mathrm{f}((1-\lambda)r_j + \lambda r^*) \leq (1-\lambda)\mathrm{f}(r_j) + \lambda \mathrm{f}(r^*)$$

for any $\lambda \in [0, 1]$. Let us define a function $\mathcal{H}(\lambda)$ such that;

$$\mathcal{H}(\lambda) = (1-\lambda)\mathrm{f}(r_j) + \lambda \mathrm{f}(r^*) - \mathrm{f}((1-\lambda)r_j + \lambda r^*). \tag{3.16}$$

Then, $\mathcal{H}(\lambda) \geq 0$.

Since the function f is continuous and differentiable, $\mathcal{H}'(\lambda)$ can be computed as follows:

$$\mathcal{H}'(\lambda) = -\mathrm{f}(r_j) + \mathrm{f}(r^*) - (r^* - r_j)\mathrm{f}'((1-\lambda)r_j + \lambda r^*).$$

Then, it follows that

$$\lim_{\lambda \to 0} \mathcal{H}'(\lambda) = -\mathrm{f}(r_j) + \mathrm{f}(r^*) - (r^* - r_j)\mathrm{f}'(r_j). \tag{3.17}$$

Now we will show that $\lim_{\lambda \to 0} \mathcal{H}'(\lambda) \geq 0$. By the definition of a derivative at a given point,

$$\lim_{\lambda \to 0} \mathcal{H}'(\lambda) = \lim_{\lambda \to 0} \frac{\mathcal{H}(\lambda) - \mathcal{H}(0)}{\lambda - 0}.$$

Then by (3.16), we know that $\mathcal{H}(0) = 0$. Hence,

$$\lim_{\lambda \to 0} \mathcal{H}'(\lambda) = \lim_{\lambda \to 0} \frac{\mathcal{H}(\lambda)}{\lambda} \geq 0 \tag{3.18}$$

since $\mathcal{H}(\lambda) \geq 0$ and $\lambda \in [0, 1]$. Then by combining (3.17) and (3.18), we get

$$-\mathrm{f}(r_j) + \mathrm{f}(r^*) - (r^* - r_j)\mathrm{f}'(r_j) \geq 0 \tag{3.19}$$

which implies that

$$\mathrm{f}(r^*) - \mathrm{f}(r_j) \geq (r^* - r_j)\mathrm{f}'(r_j)$$

$$\mathrm{f}(r^*) \geq \mathrm{f}(r_j) + (r^* - r_j)\mathrm{f}'(r_j). \tag{3.20}$$

Since $\mathrm{f}'(r_j) < 0$, (3.20) implies that

$$\begin{aligned} \mathrm{f}(r^*) &\geq \mathrm{f}(r_j) - (r^* - r_j)|\mathrm{f}'(r_j)| \\ -\mathrm{f}(r^*) &\leq -\mathrm{f}(r_j) + (r^* - r_j)|\mathrm{f}'(r_j)| \\ \frac{-\mathrm{f}(r^*)}{|\mathrm{f}'(r_j)|} &\leq \frac{-\mathrm{f}(r_j)}{|\mathrm{f}'(r_j)|} + r^* - r_j. \end{aligned}$$

Then $\mathrm{f}(r^*) = 0$ since r^* is the root of $\mathrm{f}(r)$. Then employing this result to the above inequality, we get

$$\begin{aligned} 0 &\leq \frac{-\mathrm{f}(r_j)}{|\mathrm{f}'(r_j)|} + r^* - r_j \\ r_j &\leq \frac{-\mathrm{f}(r_j)}{|\mathrm{f}'(r_j)|} + r^* \\ r_j + \frac{\mathrm{f}(r_j)}{|\mathrm{f}'(r_j)|} &\leq r^*. \end{aligned}$$

Since $\mathrm{f}'(r_j) < 0$ the above inequality can be written as

$$r_j - \frac{\mathrm{f}(r_j)}{\mathrm{f}'(r_j)} \le r^*, \tag{3.21}$$

and combining (3.21) with (3.15), we have

$$r_j \le r_{j+1} \le r^*,$$

and the inductive step is proved.

Since $\{r_j\}_{j=0}^{\infty}$ is nondecreasing and bounded above, $\{r_j\}_{j=0}^{\infty}$ converges to some value (say, $r_\infty \le r^*$) such that, $r_j - r_{j+1} \longrightarrow r_\infty - r_\infty = 0$ as $j \to \infty$. Now, $\mathrm{f}(r_j) = \mathrm{f}'(r_j)\big(r_j - r_{j+1}\big)$. We will show that $\lim\limits_{j\to\ \infty} \mathrm{f}(r_j) = 0$. Since f is a convex function on $(0, r^*]$, $\mathrm{f}'(r)$ is increasing and $\mathrm{f}'(r_0) \le \mathrm{f}'(r_j) \le \mathrm{f}'(r^*) < 0$ for all j.
So, it follows that

$$\begin{aligned}\lim_{j\to\ \infty} \mathrm{f}(r_j) &= \lim_{j\to\ \infty} \mathrm{f}'(r_j)\big(r_j - r_{j+1}\big)\\ &= -\lim_{j\to\ \infty} \mathrm{f}'(r_j)\big(r_{j+1} - r_j\big).\end{aligned}$$

Then by the Squeeze Theorem,

$$-\mathrm{f}'(r_0)\lim_{j\to\ \infty}\big(r_{j+1} - r_j\big) \ge \lim_{j\to\ \infty} \mathrm{f}(r_j) \ge -\mathrm{f}'(r^*)\lim_{j\to\ \infty}\big(r_{j+1} - r_j\big),$$

so that

$$0 \ge \lim_{j\to\ \infty} \mathrm{f}(r_j) \ge 0,$$

which implies that $\lim\limits_{j\to\ \infty} \mathrm{f}(r_j) = 0$. By the continuity of f, it follows that

$$\mathrm{f}\left(\lim_{j\to\infty} r_j\right) = \mathrm{f}(r_\infty) = 0,$$

and since r^* is the unique root of f in $(0, r^*]$, r_∞ must be r^*; that is, $\lim\limits_{j\to\infty} r_j = r^*$. □

Note that, if f is a twice differentiable function (as is the case for (3.6) with the negative binomial model), then f is convex on an interval $[a, b]$ if and only if

$f''(x)$ is nonnegative for all x in $[a, b]$; in this case, we could also prove the previous result based on the second derivative.

Now we apply Theorem 3.5 to the problem of finding the MLE for the negative binomial distribution.

THEOREM 3.6. (Convergence of the Newton-Raphson method for finding the MLE for the negative binomial distribution.)
Let f be the function defined in (3.6) and suppose $\hat{\sigma}^2 > \bar{x}$. If $\hat{r}_0$ is selected such that $f(\hat{r}_0) > 0$, then the Newton-Raphson iteration

$$\hat{r}_{j+1} = \hat{r}_j - \frac{f(\hat{r}_j)}{f'(\hat{r}_j)}$$

for $j = 0, 1, 2, \ldots$ converges to r^* where, r^* is the unique root of f.

Proof: By Theorem 3.3(b), f is a differentiable function which is positive, decreasing, and convex on the interval $(0, r^*)$ and by Theorem 3.1(b-i), r^* is the unique root of f. Also, by Theorem 3.1(b-i), if $f(\hat{r}_0) > 0$, then $\hat{r}_0$ must be in the interval $(0, r^*)$. Thus, by Theorem 3.5, $\hat{r}_j \to r^*$ as $j \to \infty$. □

A natural question raised by the statement of Theorem 3.6 is how can we choose the initial value $\hat{r}_0$ for the Newton-Raphson method to guarantee that $f(\hat{r}_0) > 0$. Of course, a trial-and-error type method could be used to eventually find an appropriate value, but it is of interest to determine if it is possible to find a closed form for $\hat{r}_0$ that always works. In practice, this is important from a computational standpoint since it eliminates the need to try several starting values to guarantee that the method will find the MLE when it exists. The following result gives a closed form for a starting value that is guaranteed to work.

THEOREM 3.7. *Let f be the function defined in (3.6).*
If $k > 1$, then $f\left(\dfrac{N_1^2}{2n(n\bar{x} - N_1)}\right) > 0$.

Proof: It suffices to find a value of r such that

$$\rho t - \ln(1+t) > 0 \tag{3.22}$$

where, $\rho = \dfrac{N_1}{n\bar{x}} = \dfrac{N_1}{N_1 + \ldots + N_k}$ and $t = \dfrac{\bar{x}}{r}$. The inequality (3.22) is equivalent to

$$e^{\rho t} > 1 + t. \tag{3.23}$$

Since $e^{\rho t} = 1 + \rho t + \frac{1}{2}\rho^2 t^2 + \ldots > 1 + \rho t + \rho^2 t^2$, a value $\hat{t}_0 > 0$ which satisfies the inequality (3.23) can be obtained by solving

$$1 + \rho \hat{t}_0 + \frac{1}{2}\rho^2 (\hat{t}_0)^2 = 1 + \hat{t}_0. \tag{3.24}$$

Solving (3.24) for t, we obtain

$$\hat{t}_0 = \frac{2(1-\rho)}{\rho^2},$$

(note $\rho \in (0,1)$ which shows that $t > 0$) and consequently,

$$\hat{r}_0 = \frac{\bar{x}}{\hat{t}_0} = \frac{\bar{x}\left(\frac{N_1}{n\bar{x}}\right)^2}{2\left(1 - \frac{N_1}{n\bar{x}}\right)} = \frac{N_1^2}{2n\left(n\bar{x} - N_1\right)}.$$

Thus, we see that

$$\begin{aligned} f(\hat{r}_0) &= \sum_{\nu=1}^{k} \frac{N_\nu}{n} (\hat{r}_0 + \nu - 1)^{-1} - \ln\left(1 + \frac{\bar{x}}{\hat{r}_0}\right) \\ &> \frac{N_1}{n\hat{r}_0} - \ln\left(1 + \frac{\bar{x}}{\hat{r}_0}\right) \\ &= \rho\hat{t}_0 - \ln(1 + \hat{t}_0) \\ &> 0. \end{aligned}$$

□

CHAPTER 4
CNV DETECTION METHODS

In the earliest days of the cytogenetics, scientists used some traditional approaches to detect CNVs in human DNA, such as, Karyotyping, which is a test to identify and evaluate the size, shape, and number of chromosomes in a sample of body cells, (Mayall et al. 1984, Bender and Kastenbaum 1969) and fluorescence in situ hybridization (FISH) (Langer-Safer et al. 1982). Then, Kallioniemi et al. (1993) introduced a rapid new method for detecting and mapping DNA amplification in tumors, called comparative genome hybridization (CGH). In 1998, Pinkel and colleagues developed array comparative genome hybridization (aCGH) which is now widely used to identify CNVs using micro-arrays (Pinkel et al. 1998). In 2003, genome-wide detection of CNVs was achieved using more accurate (aCGH) and single-nucleotide polymorphism (SNP) array approaches (Carter 2007); these approaches, however, have suffered from several inherent drawbacks, including hybridization noise, limited coverage for genome, low resolution, and difficulty in detecting novel and rare mutations (Snijders el al. 2001, Shendure and Ji 2008, Zhao et al. 2013).

In the past few years, the NGS technology brought revolutionary breakthroughs in the bio-medical field and is used in various fields of life science (Schuster 2008). Recently, a variety of CNV detection techniques were proposed, such as, CNV-seq, CNVnator, readDepth, EWT, SegSeq, etc. Some researchers have been studying these various kinds of CNV detection methods and comparing their strengths and weaknesses (Duan et al. 2013, Zhao et al. 2013).

4.1 Likelihood Based Methods

In this section, mathematically, we analyze our problem as a simple change point problem, and then we evaluate the maximum likelihood estimates (MLEs) of means of each of the continuous segments and the MLEs of change point locations, by considering the data as discrete independent random variables. MLEs for the parameters of the Negative Binomial Distribution and the Poisson Distribution are illustrated, in order to estimate the means of the segments.

4.1.1 Change Point Analysis

The change point problem always refers to the problems of identifying changes at an unknown time and estimating their locations in a series of events. Usually in the change point analysis, we first try to detect whether there is any change in the observed data, and if there is any, then estimate the number of changes and their corresponding locations. The idea of the change point problem can be summarized as described in Chen and Gupta (2011).

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent random variables with probability distribution functions $F_1, F_2, \ldots, F_n$ respectively. Then, in general, the change point problem is to test the following two hypotheses, null (H_0) versus alternative (H_1), where,

$$H_0 : F_1 = F_2 = \ldots = F_n$$

$versus$

$$H_1 : F_1 = \ldots = F_{m_1} \neq F_{m_1+1} = \ldots = F_{m_2} \neq F_{m_2+1} = \ldots F_{m_k} \neq F_{m_k+1} = \ldots = F_n$$

where, $1 \leq m_1 < m_2 < \ldots < m_k < n$, k is the number of changes, and $m_1, m_2, \ldots, m_k$ are the corresponding locations of the changes that have to be estimated. If the distributions $F_1, F_2, \ldots, F_n$ belong to a common parametric family $F(\boldsymbol{\theta})$, then the change point problem is to test

$$H_0 : \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \ldots = \boldsymbol{\theta}_n = \boldsymbol{\theta}$$

$$Vs.$$

$$H_1 : \boldsymbol{\theta}_1 = \ldots = \boldsymbol{\theta}_{m_1} \neq \boldsymbol{\theta}_{m_1+1} = \ldots = \boldsymbol{\theta}_{m_2} \neq \boldsymbol{\theta}_{m_2+1} = \ldots \boldsymbol{\theta}_{m_k} \neq \boldsymbol{\theta}_{m_k+1} = \ldots = \boldsymbol{\theta}_n$$

where, $\boldsymbol{\theta}_i; i = 1, 2, \ldots, n$ are the population parameters, and $\boldsymbol{\theta}$ is an unknown parameter that needs to be estimated along with k, and $m_1, m_2, \ldots, m_k$. This hypothesis testing attempts to reveal the existence of any change point, number of change point(s) and its(their) location(s).

A special multiple change points problem is the epidemic change point problem, which is defined by testing the following hypothesis,

$$H_0 : \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \ldots = \boldsymbol{\theta}_n = \boldsymbol{\theta}$$

$$Vs.$$

$$H_1 : \boldsymbol{\theta}_1 = \ldots = \boldsymbol{\theta}_{\alpha-1} = \boldsymbol{\mu}_1 \neq \boldsymbol{\theta}_\alpha = \ldots = \boldsymbol{\theta}_{\beta-1} = \boldsymbol{\mu}_2 \neq \boldsymbol{\theta}_\beta = \ldots = \boldsymbol{\theta}_n = \boldsymbol{\mu}_1$$

where, $1 < \alpha < \beta \leq n + 1$, and $\boldsymbol{\mu}_1$, and $\boldsymbol{\mu}_2$ are unknown. This epidemic change point problem is of great practical interest, especially in bio-medical studies.

Nowadays, scientists in the bio-medical field often use the change point inference methods to identify the copy number variation in the DNA segments. The detection of CNVs in DNA or RNA is actually a change point problem, where the read counts change in bins corresponding to copy number change. This has been widely used in many CNV detection packages such as CNV-TV by Duan et al. (2013), SeqBBS by Li et al. (2013), ReadDepth by Miller et al. (2011), PSCC by Li et al. (2014), etc. Some frequently used methods for change point estimation in the bio-informatics literature include the Bayesian test (BIC-Seq by Xi et al. 2011), maximum likelihood ratio test (m-HMM by Wang et al. 2014), non-parametric test (CNAseg by Ivanko et al. 2011, BreakDancer by Chen et al. 2009), and so on.

In this section, we first consider a simple parametric change point model under the epidemic alternative by making the following assumptions. In this model, basically we are estimating the locations of the two change points where the tandem duplication region of the genome occurs. But this model also covers the situation of single change point model when the case of $\beta = n + 1$, where α and β are the change point locations.

Suppose we observe n number of reads. Let X_i represent the starting genomic position for the ith read, so that the read counts are independent random variables. Assume that $X_1, \ldots, X_{\alpha-1}, X_\beta, \ldots, X_n$ are iid random variables with probability mass function (pmf) $P_{\boldsymbol{\theta}_0}(X = x)$, and $X_\alpha, \ldots, X_{\beta-1}$ are iid random variables with pmf $P_{\boldsymbol{\theta}_1}(X = x)$. Here $\alpha \in \{2, \ldots, n - q + 1\}$ and $\beta \in \{\alpha + q, \ldots, n + 1\}$ where, $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are parameters of the outside segments and the middle segment respectively, and are q-dimensional parameters. This change point model is illustrated in Figure 4.1.

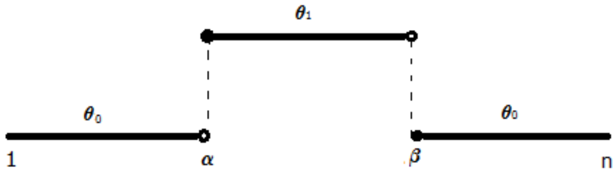


Figure 4.1: Change point model.

4.1.2 Maximum Likelihood Estimation

In this subsection, we give a detailed description of the method of estimating the maximum likelihood estimator of the parameters of each segment and the MLEs of change point locations of the above model.

The log-likelihood function for the above model based on observed data

$x_1, \dots, x_n$ is,

$$\mathcal{L}(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \alpha, \beta) = \sum_{i=1}^{\alpha-1} \ln P_{\boldsymbol{\theta}_0}(X = x_i) + \sum_{i=\alpha}^{\beta-1} \ln P_{\boldsymbol{\theta}_1}(X = x_i) + \sum_{i=\beta}^{n} \ln P_{\boldsymbol{\theta}_0}(X = x_i). \tag{4.1}$$

Then, the maximum likelihood estimate(MLE) of $\boldsymbol{\theta}_0$, $\boldsymbol{\theta}_1$, α, and β can be obtained as follows. For a particular subset (say A), the MLE of the parameters is given by

$$\hat{\boldsymbol{\theta}}_A = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i \in \boldsymbol{A}} \ln P_{\boldsymbol{\theta}}(X = x_i). \tag{4.2}$$

Then, the MLE of α and β is determined by

$$(\hat{\alpha}, \hat{\beta}) = \underset{\alpha,\beta}{\operatorname{argmax}}\, \mathcal{L}(\hat{\boldsymbol{\theta}}_{\{1,\dots,\alpha-1,\beta,\dots,n\}}, \hat{\boldsymbol{\theta}}_{\{\alpha,\dots,\beta-1\}}, \alpha, \beta). \tag{4.3}$$

That is, we evaluate (4.1) by replacing the parameters with their corresponding MLEs, for all possible values of α and β and then find the best value for α and β as $(\hat{\alpha}, \hat{\beta})$, which will maximize the evaluated function. Finally, the MLE of the parameters of the outside segments and the MLE of the parameters of the inside segments are determined by

$$\hat{\boldsymbol{\theta}}_0 = \hat{\boldsymbol{\theta}}_{\left\{1,\dots,\hat{\alpha}-1,\hat{\beta},\dots,n\right\}}, \quad \text{and} \quad \hat{\boldsymbol{\theta}}_1 = \hat{\boldsymbol{\theta}}_{\left\{\hat{\alpha},\dots,\hat{\beta}-1\right\}}. \qquad jhgf \tag{4.4}$$

4.1.3 Statistical Distributions

In our work, we map the short sequencing reads to the long reference genome and we are interested in the starting genomic position of each read. This means, the inputs for the statistical analysis are discrete non-negative integer values (count data). So, one way to look at the appropriate statistical distributions for analyzing these data is, usually, the negative binomial distribution and the Poisson distribution are often used to model the count data. Another way to look at this aspect is the probability of the starting genomic position of a given read that to be mapped to any genomic position of the reference genome is rather small compared to the number of reads we observe (n), then it can be well approximated by the Poisson distribution. However, the Poisson assumption may not be as appropriate as the negative binomial distribution when biological replicates are available and in the presence of overdispersion, that is, when the variance is larger than or equal to the mean (Dong et al. 2016).

Now, we derive the maximum likelihood estimators of each of the parameters for the corresponding distributions.

4.1.4 MLEs for the Negative Binomial Model

There are two mathematically equivalent formulations of the negative binomial distribution. One is the traditional form, which is, the negative binomial distribution estimates the probability of having a number of failures until a specified number of successes occur. The other definition is much more useful in the sequencing data, which is the negative binomial distribution can be defined as a Poisson-gamma mixture (see the discussion of the Poisson-gamma mixture in the Section 3.1). Therefore, here we consider the model in the second definition as the probability mass function of the negative binomial distribution. All the work of deriving the MLEs of the negative binomial parameters (r, p), is shown in Section 3.2.

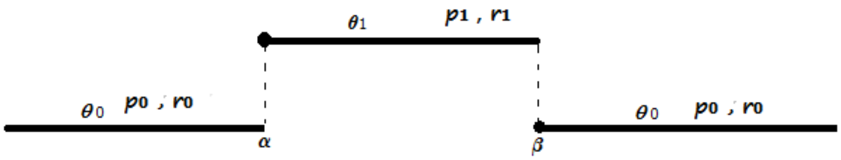


Figure 4.2: Change point model corresponding to the negative binomial distribution.

Consider the simple change point model illustrated in Figure 4.2. Assuming that each segment from the above change point model is independent, let

$X_1, X_2, \ldots, X_n$ be iid random variables from the distribution (3.1). Then the likelihood function $L(x|p_0, r_0, p_1, r_1, \alpha, \beta)$ for this model is given by,

$$L(x|p_0, r_0, p_1, r_1, \alpha, \beta) = \prod_{i=1}^{\alpha-1} \frac{\Gamma(x_i + r_0)}{x_i!\Gamma(r_0)} \quad {p_0}^{x_i}\,(1-p_0)^{r_0} \cdot \prod_{i=\alpha}^{\beta-1} \frac{\Gamma(x_i + r_1)}{x_i!\Gamma(r_1)} \quad {p_1}^{x_i}\,(1-p_1)^{r_1}$$
$$\cdot \prod_{i=\beta}^{n} \frac{\Gamma(x_i + r_0)}{x_i!\Gamma(r_0)} \quad {p_0}^{x_i}\,(1-p_0)^{r_0}\,. \tag{4.5}$$

Employing the results of equation (3.2) to the equation (4.5), the log likelihood function for the negative binomial change point model ($\mathcal{L}$) can be written as

$$\mathcal{L}(p_0, r_0, p_1, r_1, \alpha, \beta) = \sum_{i=1}^{\alpha-1} \Big\{ x_i \ln p_0 + r_0 \ln(1-p_0) + \sum_{\nu=0}^{x_i-1} \ln(r_0+\nu) - \ln(x_i!) \Big\}$$
$$+ \sum_{i=\alpha}^{\beta-1} \Big\{ x_i \ln p_1 + r_1 \ln(1-p_1) + \sum_{\nu=0}^{x_i-1} \ln(r_1+\nu) - \ln(x_i!) \Big\}$$
$$+ \sum_{i=\beta}^{n} \Big\{ x_i \ln p_0 + r_0 \ln(1-p_0) + \sum_{\nu=0}^{x_i-1} \ln(r_0+\nu) - \ln(x_i!) \Big\}. \tag{4.6}$$

Letting

$$l_{outside}(p_0, r_0, \alpha, \beta) = \sum_{i=1}^{n_0} \Big\{ x^*_{0,i} \ln p_0 + r_0 \ln(1-p_0) + \sum_{\nu=0}^{x^*_{0,i}-1} \ln(r_0+\nu) - \ln(x^*_{0,i}!) \Big\},$$

and

$$l_{inside}(p_1, r_1, \alpha, \beta) = \sum_{i=1}^{n_1} \Big\{ x^*_{1,i} \ln p_1 + r_1 \ln(1-p_1) + \sum_{\nu=0}^{x^*_{1,i}-1} \ln(r_1+\nu) - \ln(x^*_{1,i}!) \Big\}$$

where, $n_0 = n - \beta + \alpha$, $n_1 = \beta - \alpha$,

$$x^*_{0,i} = \begin{cases} x_i & \text{if } i < \alpha \\ x_{i+\beta-\alpha} & \text{if } i \geq \alpha \end{cases},$$

for $i = 1, \ldots, n_0$, and $x^*_{1,i} = x_{i+\alpha-1}$ for $i = 1, \ldots, n_1$, (4.6) can be expressed as

$$\mathcal{L}(p_0, r_0, p_1, r_1, \alpha, \beta) = l_{outside}(p_0, r_0, \alpha, \beta) + l_{inside}(p_1, r_1, \alpha, \beta).$$

The following result describes the MLE of the parameters for the NB change point model.

THEOREM 4.1. *Suppose $X_1, \ldots, \ldots, X_n$ are independent random variables such that $X_1, \ldots, X_{\alpha-1}, X_\beta, \ldots, X_n$ are negative binomial with parameters r_0 and p_0 and $X_\alpha, \ldots, X_{\beta-1}$ are negative binomial with parameters r_1 and p_1. Let $x_1, \ldots, x_n$ be the realizations of $X_1, \ldots, X_n$. Let*

$$f_0(r, \alpha, \beta) = \sum_{\nu=1}^{k} \frac{N_{0,\nu}(\alpha, \beta)}{(n-\beta+\alpha)(r+\nu-1)} - \ln\left(1 + \frac{1}{r(n-\beta+\alpha)}\left(\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i\right)\right),$$

and

$$f_1(r, \alpha, \beta) = \sum_{\nu=1}^{k} \frac{N_{1,\nu}(\alpha, \beta)}{(\beta-\alpha)(r+\nu-1)} - \ln\left(1 + \frac{1}{r(\beta-\alpha)}\sum_{i=\alpha}^{\beta-1} x_i\right)$$

where, $k = \max\{x_1, \ldots, x_n\}$, $N_{0,\nu}(\alpha, \beta) = \sum_{i=1}^{\alpha-1} I(x_i \geq \nu) + \sum_{i=\beta}^{n} I(x_i \geq \nu)$, and $N_{1,\nu}(\alpha, \beta) = \sum_{i=\alpha}^{\beta-1} I(x_i \geq \nu)$. If it exists, then the maximum likelihood estimate of $(p_0, r_0, p_1, r_1, \alpha, \beta)$ is $(\hat{p}_0, \hat{r}_0, \hat{p}_1, \hat{r}_1, \hat{\alpha}, \hat{\beta})$ where, $\hat{r}_0(\alpha, \beta)$ is the solution to $f_0(r, \alpha, \beta) = 0$, $\hat{r}_1(\alpha, \beta)$ is the solution to $f_1(r, \alpha, \beta) = 0$,

$$\hat{p}_0(\alpha, \beta) = \frac{\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i}{(n-\beta+\alpha)\hat{r}_0(\alpha, \beta) + \sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i},$$

$$\hat{p}_1(\alpha, \beta) = \frac{\sum_{i=\alpha}^{\beta-1} x_i}{(\beta-\alpha)\hat{r}_1(\alpha, \beta) + \sum_{i=\alpha}^{\beta-1} x_i},$$

$(\hat{\alpha}, \hat{\beta}) = \underset{\alpha,\beta}{argmax}\, \mathcal{L}(\hat{p}_0(\alpha, \beta), \hat{r}_0(\alpha, \beta), \hat{p}_1(\alpha, \beta), \hat{r}_1(\alpha, \beta), \alpha, \beta)$ with $\mathcal{L}$ defined in (4.6), $\hat{p}_0 = \hat{p}_0(\hat{\alpha}, \hat{\beta})$, $\hat{r}_0 = \hat{r}_0(\hat{\alpha}, \hat{\beta})$, $\hat{p}_1 = \hat{p}_1(\hat{\alpha}, \hat{\beta})$, and $\hat{r}_1 = \hat{r}_1(\hat{\alpha}, \hat{\beta})$.

Proof: First, consider α and β to be fixed. There is a unique maximizer of $l_{outside}$ if and only if

$$\frac{1}{n_0}\sum_{i=1}^{n_0} x^*_{0,i} < \frac{1}{n_0}\sum_{i=1}^{n_0}\left(x^*_{0,i} - \frac{1}{n_0}\sum_{j=1}^{n_0} x^*_{0,j}\right)^2 \tag{4.7}$$

by Theorem 3.4. If it exists, Theorem 3.4 implies this unique maximizer is $(\hat{p}_0(\alpha,\beta), \hat{r}_0(\alpha,\beta))$ where $\hat{r}_0(\alpha,\beta)$ is the solution to $f_0(r,\alpha,\beta) = 0$ guaranteed by Theorem 3.1 and

$$\begin{aligned}\hat{p}_0(\alpha,\beta) &= \frac{\frac{1}{n_0}\sum_{i=1}^{n_0} x^*_{0,i}}{\hat{r}_0(\alpha,\beta) + \frac{1}{n_0}\sum_{i=1}^{n_0} x^*_{0,i}}\\ &= \frac{\sum_{i=1}^{n_0} x^*_{0,i}}{n_0\hat{r}_0(\alpha,\beta) + \sum_{i=1}^{n_0} x^*_{0,i}}\\ &= \frac{\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i}{(n-\beta+\alpha)\hat{r}_0(\alpha,\beta) + \sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i}.\end{aligned}$$

Similarly, there is a unique maximizer of l_{inside} if and only if

$$\frac{1}{n_1}\sum_{i=1}^{n_1} x^*_{1,i} < \frac{1}{n_1}\sum_{i=1}^{n_1}\left(x^*_{1,i} - \frac{1}{n_1}\sum_{j=1}^{n_1} x^*_{1,j}\right)^2, \tag{4.8}$$

and, if it exists, it is $(\hat{p}_1(\alpha,\beta), \hat{r}_1(\alpha,\beta))$.

Next, when we compute

$$\sup_{p_0,r_0,p_1,r_1,\alpha,\beta} \mathcal{L}(p_0,r_0,p_1,r_1,\alpha,\beta) =$$

$$\max_{\alpha,\beta}\left\{\sup_{p_0(\alpha,\beta),r_0(\alpha,\beta)} l_{outside}(p_0(\alpha,\beta), r_0(\alpha,\beta),\alpha,\beta) + \sup_{p_1(\alpha,\beta),r_1(\alpha,\beta)} l_{inside}(p_1(\alpha,\beta), r_1(\alpha,\beta),\alpha,\beta)\right\} \tag{4.9}$$

the supremums both exist only if (4.7) and (4.8) hold for at least one pair (α,β) which maximizes (4.9). □

Note that equation (4.9) can be used to compute the supremum of $\mathcal{L}(p_0,r_0,p_1,r_1,\alpha,\beta)$ even when the MLE does not exist. The supremums on the right side of (4.9) can be computed using Theorem 3.4 based on whether or not conditions (4.7) and (4.8) are satisfied.

The expected value of the distribution (3.1) can be obtained as shown in the following result.

THEOREM 4.2. *If X follows a negative binomial distribution with parameters r and p, then* $E(X) = \dfrac{rp}{1-p}$.

Proof: By the definition of the mean, we have

$$\begin{aligned}
E(X) &= \sum_{x=0}^{\infty} x \cdot \frac{\Gamma(x+r)}{x!\Gamma(r)} \quad p^x \,(1-p)^r \\
&= \sum_{x=0}^{\infty} x \cdot \frac{(x+r-1)!}{x\cdot(x-1)!\cdot(r-1)!} \quad p^x \,(1-p)^r \\
&= (1-p)^r \sum_{x=0}^{\infty} \frac{(x+r-1!)}{(x-1)!\cdot(r-1)!} \quad p^x \\
&= (1-p)^r \sum_{x=0}^{\infty} \frac{((x-1)+(r+1)-1)!}{(x-1)!\cdot(r-1)!} \quad p^x \\
&= (1-p)^r \sum_{x=0}^{\infty} \frac{r\cdot((x-1)+(r+1)-1)!}{(x-1)!\cdot r\cdot(r-1)!} \quad p\cdot p^{x-1} \\
&= (1-p)^r \cdot rp \sum_{x=0}^{\infty} \frac{((r+1)+(x-1)-1)!}{(x-1)!\cdot r!} \quad p^{x-1} \\
&= (1-p)^r \cdot rp \sum_{x=0}^{\infty} \binom{(r+1)+(x-1)-1}{(x-1)} \quad p^{x-1}.
\end{aligned}$$

Since $0 < p < 1$, by the Newton's generalized binomial theorem,

$$\sum_{x=0}^{\infty} \binom{(r+1)+(x-1)-1}{(x-1)} \quad p^{x-1} = \frac{1}{(1-p)^{r+1}}.$$

Thus, it follows that

$$\begin{aligned}
E(X) &= (1-p)^r \cdot rp \cdot \frac{1}{(1-p)^{r+1}} \\
&= \frac{rp}{(1-p)}. \quad \square
\end{aligned}$$

Therefore, by the invariance property of the MLE, the MLE of the mean is given by, $E(X) = \dfrac{\hat{r}\hat{p}}{(1-\hat{p})}$. Consequently, the MLE of the means of the outside segments is given by

$$\hat{\mu}_0 = \frac{\hat{r}_0 \cdot \hat{p}_0}{1-\hat{p}_0}$$

and the MLE of the mean of the inside segment is given by

$$\hat{\mu}_1 = \frac{\hat{r}_1 \cdot \hat{p}_1}{1-\hat{p}_1}.$$

4.1.5 MLEs for the Poisson Model

In the literature, many authors have demonstrated that, when working with the NGS data, the read counts follow the Poisson distribution. SeqCNV by Chen et al. (2017), worked with the simulated data by assuming that the number of reads for each target followed a Poisson distribution with the product of the affinity and length, and coordinated within the range being sampled. Under the assumption that the reads are randomly and independently sampled from any location of the test genome with equal probability, Yoon et al. (2009) has considered the read counts, that are mapped into a window of the reference genome, follows the Poisson distribution. Bentley et al. (2008) and Yoon et al. (2009) have reported that read counts generated by the Illumina Genome Analyzer platform follows a pattern of Poisson distribution with slight overdispersion. Ji and Chen (2015) mentioned that, the natural way to think about the re-alignment of the short reads to the genome is to view this process as a Poisson process of observing the number of reads mapped to a specific genomic region. If there are no CNVs, then the number of reads should follow a homogeneous Poisson process with a fixed average mean read count. When the Poisson process starts to depart from its homogeneous feature as indicated by a non-constant average mean count, then there is an indication of presence of CNVs in the genomic region. So, it is worthwhile to look at the MLEs for the means and change points under the simple epidemic alternative illustrated in Figure 4.3 when the reads follow the Poisson distribution. Here μ is the parameter of the Poisson distribution (which is the mean of the distribution) and δ is the height corresponding to the jump.

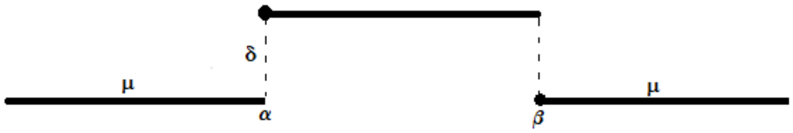


Figure 4.3: Change point model corresponding to the Poisson distribution.

Let $X_1, X_2, \ldots, X_n$ be iid random variables from the distribution with the probability mass function

$$P(X = x) = \frac{e^{-\mu}\mu^x}{x!} \tag{4.10}$$

where, $x \in \{0, 1, 2, \ldots\}$, and $\mu > 0$. Then the likelihood function $L(x|\mu, \delta, \alpha, \beta)$ for this model is given by,

$$L(x|\mu, \delta, \alpha, \beta) = \prod_{i=1}^{\alpha-1} \frac{e^{-\mu}\mu^{x_i}}{x_i!} \cdot \prod_{i=\alpha}^{\beta-1} \frac{e^{-(\mu+\delta)}(\mu+\delta)^{x_i}}{x_i!} \cdot \prod_{i=\beta}^{n} \frac{e^{-\mu}\mu^{x_i}}{x_i!}. \tag{4.11}$$

If we simplify (4.11) further, we can obtain the following simplified form of the likelihood function.

$$\begin{aligned} L(x|\mu, \delta, \alpha, \beta) &= \prod_{i=1}^{\alpha-1} \frac{e^{-\mu}\mu^{x_i}}{x_i!} \cdot \prod_{i=\alpha}^{\beta-1} \frac{e^{-\mu} \cdot e^{-\delta}(\mu+\delta)^{x_i}}{x_i!} \cdot \frac{\mu^{x_i}}{\mu^{x_i}} \cdot \prod_{i=\beta}^{n} \frac{e^{-\mu}\mu^{x_i}}{x_i!} \\ &= \prod_{i=1}^{\alpha-1} \frac{e^{-\mu}\mu^{x_i}}{x_i!} \cdot \prod_{i=\alpha}^{\beta-1} \left\{ \frac{e^{-\mu}\mu^{x_i}}{x_i!} \cdot e^{-\delta} \left(\frac{\mu+\delta}{\mu}\right)^{x_i} \right\} \cdot \prod_{i=\beta}^{n} \frac{e^{-\mu}\mu^{x_i}}{x_i!} \\ &= \prod_{i=1}^{n} \frac{e^{-\mu}\mu^{x_i}}{x_i!} \cdot \prod_{i=\alpha}^{\beta-1} e^{-\delta} \left(\frac{\mu+\delta}{\mu}\right)^{x_i}. \end{aligned}$$

Then, the log likelihood function can be written as follows.

$$\mathcal{L}(x|\mu, \delta, \alpha, \beta) = \sum_{i=1}^{n} \ln \left(\frac{e^{-\mu}\mu^{x_i}}{x_i!}\right) + \sum_{i=\alpha}^{\beta-1} \ln \left(e^{-\delta} \frac{\mu+\delta}{\mu}\right)^{x_i} \tag{4.12}$$

$$\mathcal{L}(x|\mu,\delta,\alpha,\beta) = -n\mu + \ln\mu\sum_{i=1}^{n}x_i - \sum_{i=1}^{n}\ln(x_i!) - (\beta-\alpha)\delta + [\ln(\mu+\delta) - \ln\mu]\sum_{i=\alpha}^{\beta-1}x_i. \tag{4.13}$$

THEOREM 4.3. *Suppose $X_1,\ldots,\ldots,X_n$ are independent random variables such that $X_1,\ldots,X_{\alpha-1},X_\beta,\ldots,X_n$ are Poisson with mean μ, and $X_\alpha,\ldots,X_{\beta-1}$ are Poisson with mean $\mu+\delta$. Let $x_1,\ldots,x_n$ be the realizations of $X_1,\ldots,X_n$. Then the maximum likelihood estimate of $(\mu,\delta,\alpha,\beta)$ is $(\hat{\mu},\hat{\delta},\hat{\alpha},\hat{\beta})$ where*

$$\hat{\mu}(\alpha,\beta) = \frac{1}{n-\beta+\alpha}\Big(\sum_{i=1}^{\alpha-1}x_i + \sum_{i=\beta}^{n}x_i\Big),$$

$$\hat{\delta}(\alpha,\beta) = \frac{1}{\beta-\alpha}\sum_{i=\alpha}^{\beta-1}x_i - \frac{1}{n-\beta+\alpha}\Big(\sum_{i=1}^{\alpha-1}x_i + \sum_{i=\beta}^{n}x_i\Big),$$

$(\hat{\alpha},\hat{\beta}) = \underset{\alpha,\beta}{argmax}\,\mathcal{L}(\hat{\mu}(\alpha,\beta),\hat{\delta}(\alpha,\beta),\alpha,\beta)$ with $\mathcal{L}$ defined in (4.13), $\hat{\mu} = \hat{\mu}(\hat{\alpha},\hat{\beta})$, and $\hat{\delta} = \hat{\delta}(\hat{\alpha},\hat{\beta})$.

Proof: Differentating the log likelihood function with respect to δ, we get,

$$\frac{\partial\mathcal{L}}{\partial\delta}(\mu,\delta) = -(\beta-\alpha) + \sum_{i=\alpha}^{\beta-1}x_i\Big(\frac{1}{\mu+\delta}\Big). \tag{4.14}$$

Partial differentiation of the log likelihood function with respect to μ yields,

$$\frac{\partial\mathcal{L}}{\partial\mu}(\mu,\delta) = -n + \frac{1}{\mu}\sum_{i=1}^{n}x_i + \Big(\frac{1}{\mu+\delta} - \frac{1}{\mu}\Big)\sum_{i=\alpha}^{\beta-1}x_i. \tag{4.15}$$

Now we set $\frac{\partial\mathcal{L}}{\partial\delta} = 0$ and $\frac{\partial\mathcal{L}}{\partial\mu} = 0$, and let $\tilde{\mu}$ and $\tilde{\delta}$ denote values of μ and δ which solve these pair of equations. Solving $\frac{\partial\mathcal{L}}{\partial\delta} = 0$ for $\beta-\alpha$ gives

$$\begin{aligned}\beta-\alpha &= \frac{1}{\tilde{\mu}+\tilde{\delta}}\sum_{i=\alpha}^{\beta-1}x_i\\ &= \frac{1}{\tilde{\mu}+\tilde{\delta}}\sum_{i=\alpha}^{\beta-1}x_i.\end{aligned}$$

Then,

$$\tilde{\mu} + \tilde{\delta} = \frac{1}{\beta - \alpha} \sum_{i=\alpha}^{\beta-1} x_i. \tag{4.16}$$

Using (4.15), $\dfrac{\partial \mathcal{L}}{\partial \mu} = 0$ gives

$$-n + \frac{1}{\tilde{\mu}} \sum_{i=1}^{n} x_i + \Big(\frac{1}{\tilde{\mu} + \tilde{\delta}} - \frac{1}{\tilde{\mu}}\Big) \sum_{i=\alpha}^{\beta-1} x_i = 0. \tag{4.17}$$

Then by applying the results of equation (4.16) to the equation (4.17), we get,

$$\begin{aligned}
-n + \frac{1}{\tilde{\mu}} \sum_{i=1}^{n} x_i + \Big((\beta - \alpha)\frac{1}{\sum_{i=\alpha}^{\beta-1} x_i} - \frac{1}{\tilde{\mu}}\Big) \sum_{i=\alpha}^{\beta-1} x_i &= 0 \\
-n + \frac{1}{\tilde{\mu}} \sum_{i=1}^{n} x_i + \beta - \alpha - \Big(\frac{\sum_{i=\alpha}^{\beta-1} x_i}{\tilde{\mu}}\Big) &= 0 \\
\sum_{i=1}^{n} x_i - \sum_{i=\alpha}^{\beta-1} x_i &= \tilde{\mu}(n - \beta + \alpha) \\
\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\alpha}^{\beta-1} x_i + \sum_{i=\beta}^{n} x_i - \sum_{i=\alpha}^{\beta-1} x_i &= \tilde{\mu}(n - \beta + \alpha) \\
\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i &= \tilde{\mu}(n - \beta + \alpha),
\end{aligned}$$

so that

$$\tilde{\mu} = \frac{1}{n - \beta + \alpha}\Big(\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i\Big) = \hat{\mu}(\alpha, \beta). \tag{4.18}$$

Then, substituting the equation (4.18) to the equation (4.16), we get

$$\tilde{\delta} = \frac{1}{(\beta - \alpha)} \sum_{i=\alpha}^{\beta-1} x_i - \frac{1}{n - \beta + \alpha}\Big(\sum_{i=1}^{\alpha-1} x_i + \sum_{i=\beta}^{n} x_i\Big) = \hat{\delta}(\alpha, \beta). \tag{4.19}$$

Now, we show that, for fixed α and β, $(\tilde{\mu}, \tilde{\delta})$ maximizes $\mathcal{L}$. To prove this, we show that, $\dfrac{\partial^2 \mathcal{L}}{\partial^2 \mu}(\alpha, \beta) < 0$ and the Hessian matrix $H(\mu, \delta)$ is negative definite, which is, the determinant of $H(\mu, \delta) > 0$. That is, $|H(\mu, \delta)| = |\nabla^2 \mathcal{L}(\mu, \delta)| > 0$.

First we show that, $\dfrac{\partial^2 \mathcal{L}}{\partial^2 \mu}(\alpha, \beta) < 0$. Partial derivative of (4.15) with respect to μ yields,

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial \mu^2}(\mu, \delta) &= -\frac{1}{\mu^2}\sum_{i=1}^{n} x_i - \frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i + \frac{1}{\mu^2}\sum_{i=\alpha}^{\beta-1} x_i \\
&= -\frac{1}{\mu^2}\sum_{i=1}^{\alpha-1} x_i - \frac{1}{\mu^2}\sum_{i=\alpha}^{\beta-1} x_i - \frac{1}{\mu^2}\sum_{i=\beta}^{n} x_i - \frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i + \frac{1}{\mu^2}\sum_{i=\alpha}^{\beta-1} x_i \\
&= -\frac{1}{\mu^2}\sum_{i=1}^{\alpha-1} x_i - \frac{1}{\mu^2}\sum_{i=\beta}^{n} x_i - \frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i \\
&< 0.
\end{aligned}$$

Now we show that, $|H(\mu, \delta)| = |\nabla^2 \mathcal{L}(\mu, \delta)| = \begin{vmatrix} \frac{\partial^2 \mathcal{L}(\mu,\delta)}{\partial \mu^2} & \frac{\partial^2 \mathcal{L}(\mu,\delta)}{\partial \mu \partial \delta} \\ \frac{\partial^2 \mathcal{L}(\mu,\delta)}{\partial \delta \partial \mu} & \frac{\partial^2 \mathcal{L}(\mu,\delta)}{\partial \delta^2} \end{vmatrix} > 0.$

We found that,

$$\frac{\partial^2 \mathcal{L}(\mu, \delta)}{\partial \mu \partial \delta} = \frac{\partial^2 \mathcal{L}(\mu, \delta)}{\partial \delta \partial \mu} = \frac{\partial^2 \mathcal{L}(\mu, \delta)}{\partial \delta^2} = -\frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i.$$

Then, by applying these results to the Hessian matrix, we get

$$\begin{aligned}
|H(\mu, \delta)| = \left(\frac{1}{\mu^2}\sum_{i=1}^{\alpha-1} x_i + \frac{1}{\mu^2}\sum_{i=\beta}^{n} x_i + \frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i\right) \cdot \left(\frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i\right) \\
- \left(\frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i\right) \cdot \left(\frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i\right).
\end{aligned}$$

By simplifying this further, we obtain

$$|H(\mu, \delta)| = \left(\frac{1}{\mu^2}\sum_{i=1}^{\alpha-1} x_i + \frac{1}{\mu^2}\sum_{i=\beta}^{n} x_i\right) \cdot \frac{1}{(\mu+\delta)^2}\sum_{i=\alpha}^{\beta-1} x_i > 0$$

which completes the proof. □

Hence, the MLE of the mean of the outside segments is given by,

$$\hat{\boldsymbol{\theta}}_0 = \hat{\mu}(\hat{\alpha}, \hat{\beta}) = \frac{1}{n - \hat{\beta} + \hat{\alpha}}\Big(\sum_{i=1}^{\hat{\alpha}-1} x_i + \sum_{i=\hat{\beta}}^{n} x_i\Big),$$

and the MLE of the mean of the inside segment is given by,

$$\hat{\boldsymbol{\theta}}_1 = \hat{\mu}(\hat{\alpha}, \hat{\beta}) + \hat{\delta}(\hat{\alpha}, \hat{\beta}) = \frac{1}{(\hat{\beta} - \hat{\alpha})} \sum_{i=\hat{\alpha}}^{\hat{\beta}-1} x_i.$$

4.2 Circular Binary Segmentation

Circular Binary Segmentation was proposed by Olshen and Venkatraman (2004) to identify DNA copy number changes in an aCGH database on the mean change point model. In aCGH experiments, the detection of CNVs, gains or losses, can be identified based on the ratio of the test sample intensity to the reference sample intensity as the ratio being higher or lower respectively. Let T_i and R_i be the test sample intensity and the corresponding reference sample intensity at the locus i respectively. The $\log_2 \frac{T_i}{R_i}$, is considered as a random variable used for the derivation of a copy number, which is assumed to follow a Gaussian distribution with mean 0 and constant variance σ^2. Then, deviations from the constant parameters (mean and variance) presented in $\log_2 \frac{T_i}{R_i}$ data may indicate a copy number change. Here, $\log_2 \frac{T_i}{R_i} = 0$ indicates no DNA copy number change at locus i, $\log_2 \frac{T_i}{R_i} < 0$ reveals a deletion at locus i, and $\log_2 \frac{T_i}{R_i} > 0$ signifies a duplication in the test sample at that locus.

CBS is a modification of binary segmentation. (Sen and Srivastava 1975). It is an estimation algorithm which uses a likelihood ratio statistic to test the null hypothesis of no change points in a sequence. If the null hypothesis is rejected, the sequence is split and the test is recursively applied to the resulting sub-segments until no additional changes are detected (Erdman and Emerson 2008). We summarized the CBS method as follows (Olshen and Venkatraman 2004);

Let $X_1, X_2, \ldots, X_n$ be the log ratios of the intensities of the n locis being tested, and let $S_i = X_1 + X_2 + \ldots + X_i, \quad 1 \leq i \leq n$, be the partial sums. When the data are normally distributed with a known variance, by considering the segment to be spliced at the two ends to form a circle, the likelihood ratio test statistic for testing the hypothesis, that the arc from $i+1$ to j and its complement have different means (modification of the likelihood ratio test statistic by Sen and Srivastava (1975)), is given by

$$Z_{ij} = \left\{ \frac{1}{(j-i)} + \frac{1}{(n-j+i)} \right\}^{-1/2} \cdot \left\{ \frac{(S_j - S_i)}{(j-i)} - \frac{(S_n - S_j + S_i)}{(n-j+i)} \right\}.$$

Then, the CBS is based on the statistic

$$Z_C = max_{1 \leq i < j \leq n} |Z_{ij}|.$$

Here, Z_C allows for both a single change ($j = n$) and the epidemic alternative ($j < n$). If the statistic exceeds an appropriate threshold level (critical value), then it declares a change. When the data are normal, this critical value can be computed using the Monte Carlo Simulations or the approximation given by Siegmund (1986) for the tail probability. If the null hypothesis is rejected the change-point(s) is (are) estimated to be i (and j) such that $Z_C = |Z_{ij}|$ and the procedure is applied recursively to identify all the changes.

When the data, X_i's, are not normal, Olshen and Venkatraman generalized the above procedure to the non-normal data by generating a reference distribution using a permutation approach, considering the X_i's are identically distributed under the null hypothesis of no change point. Let $X_1^*, X_2^*, \ldots, X_n^*$ be random permutation of the data and let $Z_C^* = max|Z_{ij}^*|$ be the statistic derived as above from the permuted data. Here, for the estimation of p-value, it requires large number of permutations and is computationaly intensive. Because of this computational complexity, Venkatraman and Olshen proposed a faster circular binary segmentation algorithm (Venkatraman and Olshen 2007).

4.2.1 Likelihood Ratio Test with Parametric Bootstrap

A standard test when the parametric form of a model is known is the (log-) likelihood ratio test. In change point problems, the sampling distribution of this statistic is complicated, and parametric bootstrap procedures are often used to estimate the p-value of the test. This section gives a general desription of the (log-) likelihood ratio test with a p-value estimated by the parametric boostrap.

Consider the general setting where, X_i has pmf $P_{\boldsymbol{\theta}}$ for $i = 1, \ldots, n$, and the likelihood function for $X_1, \ldots, X_n$ is $L(\boldsymbol{\theta})$ where, $\boldsymbol{\theta} \in \Theta$, and we wish to test the null hypothesis $H_0 : \boldsymbol{\theta} \in \Theta_0$ for some set Θ_0 against the alternative $H_1 : \boldsymbol{\theta} \in \Theta \cap \Theta_0'$. The log-likelihood ratio test statistic is then defined to be

$$\Lambda(X_1, \ldots, X_n) = \ln \frac{\sup\limits_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta})}{\sup\limits_{\boldsymbol{\theta} \in \Theta_0} L(\boldsymbol{\theta})}.$$

Given observed data $(x_1, \ldots, x_n)$, the null hypothesis H_0 is rejected when $\Lambda(x_1, \ldots, x_n)$ is sufficiently large.

The parametric bootstrap can be used to estimate the p-value for testing H_0 versus H_1. It uses the estimate of $\boldsymbol{\theta}$ under H_0 (say $\hat{\boldsymbol{\theta}}_0$) to generate B new bootstrap samples $(x_1^{(1)}, \ldots, x_n^{(1)})$,. . .,$(x_1^{(B)}, \ldots, x_n^{(B)})$; for any $i \in \{1, \ldots, n\}$, $X_i^{(b)}, b = 1, \ldots, B$, are iid random variables with pmf $P_{\hat{\boldsymbol{\theta}}_0}$. So, $\Lambda(X_1^{(b)}, \ldots, X_n^{(b)}), b = 1, \ldots, B$ is a sample from the sampling distribution of Λ under H_0, and an estimate of the p-value for testing H_0 versus H_1 is

$$\widehat{p\text{-value}} = \frac{1}{B} \sum_{b=1}^{B} I\left(\Lambda(x_1^{(b)}, \ldots, x_n^{(b)}) \geq \Lambda(x_1, \ldots, x_n)\right).$$

Finally, rejecting H_0 if the p-value is less than α gives us an approximate α-level test of H_0 versus H_1.

4.3 CBS Procedures for CNV Detection and Estimation

In this section, we consider several procedures for detection of copy number variation and estimation of the mean values for the read counts. Further, we perform various simulation studies to assess the performance of these procedures under different true models.

4.3.1 CBS Procedures

A widely-used method in the literature is the `segment` function in the `DNAcopy` package (Seshan and Olshen 2017). In this section, we use the default parameters for this function except for the size which we change to `alpha=0.05` to make it comparable to the other methods used in the simulation studies. In the simulations, we refer to this methods as the *DNAcopy* method.

We also consider procedures which perform circular binary segmentation using successive likelihood ratio tests based on the parametric bootstrap for two different underlying models for discrete count data. The simpler model assumes the Poisson model with the likelihood function defined in (4.11). We refer to the likelihood ratio test with the parametric bootstrap as the *Poisson CBS* method.

A more flexible model designed to account for overdispersion assumes the negative binomial model with the likelihood function defined in (4.5). We refer to this likelihood ratio test with the parametric bootstrap as the *Negative Binomial CBS* method.

4.3.2 Simulations Under Poisson Models

Here, we assume that the true underlying model is that $X_1, \ldots, X_n$ are independent Poisson random variables where, X_i has mean μ_i for $i = 1, \ldots, n$ under six scenarios. For each of the methods, we generate $R = 1000$ data sets and, for each data set and method, we tabulate the number of estimated continuous segments, and the root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_i - \mu_i)^2}$$

where, $\hat{\mu}_i$ is the estimated mean of X_i at index i.

The first scenario considered is one in which there is no copy number variation. We generate $R = 1000$ samples of size $n = 50$. For each sample, the data is simulated from the model

$X_i \sim \text{Poisson}(\mu = 10), i = 1, \ldots, 50.$

We refer to this simulation *pN1.* Some results are shown in Table 4.1.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.4259	.956	.010	.031	.001	.002	.000
Poisson CBS	.4341	.953	.002	.041	.001	.003	.000
Negative Binomial CBS	.4102	.966	.002	.031	.001	.000	.000

Table 4.1: Results for simulation under Poisson scenario *pN1.*

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. All of the procedures identified one segment about 95% of the time (all are slightly higher) so it appears that the nominal size specified for each test is reasonable. Interestingly, even though in the true model the random variables are Poisson, the average RMSE is highest for the *Poisson CBS* method and lowest for the *Negative Binomial CBS* method.

Next, in scenario *pH1*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{Poisson}(\mu = 10), i = 1, \ldots, 15 \\ \text{Poisson}(\mu = 15), i = 16, \ldots, 25 \\ \text{Poisson}(\mu = 10), i = 26, \ldots, 50 \end{cases}.$$

Some results are shown in Table 4.2.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	1.5995	.287	.003	.641	.026	.042	.001
Poisson CBS	1.5081	.205	.001	.756	.002	.033	.003
Negative Binomial CBS	1.5459	.296	.001	.683	.002	.018	.000

Table 4.2: Results for simulation under Poisson scenario *pH1*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Poisson CBS* method does best both in terms of correctly identifying the correct number of segments the highest proportion of times (.756) and having the smallest RMSE (1.5081).

Next, in scenario *pF1*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{Poisson}(\mu = 10), i = 1, \ldots, 15 \\ \text{Poisson}(\mu = 20), i = 16, \ldots, 25 \\ \text{Poisson}(\mu = 10), i = 26, \ldots, 50 \end{cases}.$$

Some results are shown in Table 4.3.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	1.3029	.003	.000	.901	.049	.043	.004
Poisson CBS	1.1945	.000	.000	.923	.008	.064	.005
Negative Binomial CBS	1.1554	.001	.000	.955	.004	.039	.001

Table 4.3: Results for simulation under Poisson scenario *pF1*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, it is again interestingly that the *Negative Binomial CBS* method does best in correctly idenifying the 3 segments the highest proportion of times (.955) and in having the smallest RMSE (1.1554). In this case though the *Poisson CBS* clearly is second best among the three methods.

Now, we consider scenarios with a smaller mean read count. In scenario *pN2*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$X_i \sim \text{Poisson}(\mu = 2.5), i = 1, \ldots, 50.$

Some results are shown in Table 4.4.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.1958	.973	.006	.020	.001	.000	.000
Poisson CBS	.2066	.958	.001	.040	.000	.001	.000
Negative Binomial CBS	.1982	.967	.001	.032	.000	.000	.000

Table 4.4: Results for simulation under Poisson scenario *pN2*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. The results for this scenario are similar to what was seen for scenario *pN1* in Table 4.1. Even though the true distribution is Poisson, the average RMSE for the *Poisson CBS* method is highest (.2066) though the proportion of times it fails to identify only one segment ($1 - .958 = .042$) is closest to the nominal size .05. However, *DNAcopy* does have the smallest average RMSE in this case, while the *Negative Binomial CBS* method had the lowest in scenario *pN1*.

Next, in scenario *pH2*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{Poisson}(\mu = 2.5), i = 1, \ldots, 15 \\ \text{Poisson}(\mu = 3.75), i = 16, \ldots, 25 \\ \text{Poisson}(\mu = 2.5), i = 26, \ldots, 50 \end{cases} .$$

Some results are shown in Table 4.5.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.5969	.818	.005	.162	.003	.012	.000
Poisson CBS	.6141	.799	.003	.187	.000	.011	.000
Negative Binomial CBS	.5883	.868	.001	.127	.000	.004	.000

Table 4.5: Results for simulation under Poisson scenario *pH2*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, although *Poisson CBS* does best in terms of having the highest proportion of times it correctly identifies three segments (.187), it is worst with the highest RMSE, even though the true model is Poisson. The *Negative Binomial CBS* method is best in terms of lowest average RMSE (.5883), though it is conservative and identifies three segments is lowest proportion of times among these three methods.

Next, in scenario *pF2*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{Poisson}(\mu = 2.5), i = 1, \ldots, 15 \\ \text{Poisson}(\mu = 5), i = 16, \ldots, 25 \\ \text{Poisson}(\mu = 2.5), i = 26, \ldots, 50 \end{cases} .$$

Some results are shown in Table 4.6.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.8693	.378	.003	.585	.016	.018	.000
Poisson CBS	.8344	.280	.001	.681	.002	.035	.001
Negative Binomial CBS	.8497	.387	.000	.596	.001	.016	.000

Table 4.6: Results for simulation under Poisson scenario *pF2*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. As opposed to scenario *pF1*, the *Poisson CBS* method does better than the *Negative Binomial CBS* method with the lower average RMSE (.8344) and the higher proportion of times it correctly identifies three segments. In both scenarios, both methods do better than *DNAcopy*.

Overall, for the Poisson scenarios, we see that the *Negative Binomial CBS* method does well under all scenarios, in some cases even doing better than the *Poisson CBS* method, which might be expected to do best it has a correct and more precise specification of the true model. The *DNAcopy* method also does reasonably well and appears robust among the scenarios considered; overall in these scenarios, it does not do as well as the *Negative Binomial CBS*, particularly in terms of average RMSE, but this should be expected since it does not make as strong of assumptions about the true model. Of course, it should be noted that the *Negative Binomial CBS* method has the highest computation time compared with the other CBS methods due to the iterative nature of the Newton-Raphson method.

4.3.3 Simulations Under Negative Binomial Models

In this subsection, we instead assume that the true underlying model is that $X_1, \ldots, X_n$ are independent negative binomial random variables where X_i has parameters r_i and p_i for $i = 1, \ldots, n$ under nine scenarios. For each of the methods, we generate $R = 1000$ data sets and, for each data set and method, we tabulate the number of estimated continuous segments and the root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_i - \mu_i)^2}$$

where, $\hat{\mu}_i$ is the estimated mean of X_i at index i.

First, we consider scenario *nbN1* in which there is no copy number variation and generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$X_i \sim \text{negative binomial}(r = 2.5, p = .8), i = 1, \ldots, 50.$

Note that in R the probability of success and failure is reversed, so the command that generates each sample is `rnbinom(50,prob=.2,size=2.5)`. Some results are shown in Table 4.7.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.9387	.962	.009	.029	.000	.000	.000
Poisson CBS	5.6732	.002	.001	.006	.002	.025	.964
Negative Binomial CBS	.9225	.949	.001	.046	.001	.003	.000

Table 4.7: Results for simulation under negative binomial scenario *nbN1*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Negative Binomial CBS* method has the lowest average RMSE (.9225) and *DNAcopy* is close with (5.6732); for both of these methods, the nominal size specified is reasonable. The model is misspecified for the *Poisson CBS* method, and it is seen that this causes the method to almost always overestimate the number of segments and, in most cases (96.4%), it found over five segments. Also, the average RMSE is very high (5.6732) relative to the other methods.

Next, in scenario *nbH1*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{negative binomial}(r = 2.5, p = .8), i = 1, \ldots, 15 \\ \text{negative binomial}(r = 3.75, p = .8), i = 16, \ldots, 25 \\ \text{negative binomial}(r = 2.5, p = .8), i = 26, \ldots, 50 \end{cases} .$$

Some results are shown in Table 4.8.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	2.4687	.866	.006	.115	.004	.009	.000
Poisson CBS	6.0836	.000	.000	.002	.004	.016	.978
Negative Binomial CBS	2.4630	.893	.003	.139	.001	.016	.002

Table 4.8: Results for simulation under negative binomial scenario *nbH1*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Negative Binomial CBS* method again has the smallest average RMSE (2.4630), *DNAcopy* is very close (2.4687), and the *Poisson CBS* method is much worse (6.0836). The *Negative Binomial CBS* method also correctly identifies the number of segments the highest proportion of the time (.139), while the *Poisson CBS* method again almost always find over five segments (97.8% of the time).

Next, in scenario *nbF1*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{negative binomial}(r = 2.5, p = .8), i = 1, \ldots, 15 \\ \text{negative binomial}(r = 5, p = .8), i = 16, \ldots, 25 \\ \text{negative binomial}(r = 2.5, p = .8), i = 26, \ldots, 50 \end{cases} .$$

Some results are shown in Table 4.9.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	3.9257	.564	.003	.389	.022	.021	.001
Poisson CBS	6.4815	.000	.000	.002	.001	.011	.986
Negative Binomial CBS	3.8534	.546	.003	.409	.007	.028	.007

Table 4.9: Results for simulation under negative binomial scenario *nbF1*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Negative Binomial CBS* method again has the smallest average RMSE (3.8534) and the *Poisson CBS* method is much worse (6.4815). The *Negative Binomial CBS* method also correctly identifies the number of segments the highest proportion of the time (.409), while the *Poisson CBS* method again almost always find over five segments (98.6% of the time).

Now, we consider scenarios where p is smaller so the variance is closer to the mean. In scenario *nbN2*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$X_i \sim$ negative binomial$(r = 10, p = .2), i = 1, \ldots, 50.$

Some results are shown in Table 4.10.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.2420	.954	.010	.032	.001	.002	.001
Poisson CBS	.3324	.847	.008	.127	.000	.005	.000
Negative Binomial CBS	.2439	.947	.003	.048	.000	.002	.000

Table 4.10: Results for simulation under negative binomial scenario *nbN2*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, *DNAcopy* interestingly has the lowest average RMSE (.2420) with the *Negative Binomial CBS* very close (.2439), and both appear to have reasonable nominal size. The overdispersion is not as severe and it is seen that the average RMSE for the *Poisson CBS* method (.3324) is not as bad as in scenarios with $p = .8$ and the proportion of times that the number of segments is not correctly determined to be one $(1 - .847 = .153)$ is closer to the nominal size.

Next, in scenario *nbH2*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{negative binomial}(r = 10, p = .2), i = 1, \dots, 15 \\ \text{negative binomial}(r = 15, p = .2), i = 16, \dots, 25 \\ \text{negative binomial}(r = 10, p = .2), i = 26, \dots, 50 \end{cases} .$$

Some results are shown in Table 4.11.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.6205	.865	.007	.115	.003	.009	.001
Poisson CBS	.7045	.668	.006	.290	.001	.034	.001
Negative Binomial CBS	.6162	.857	.007	.131	.000	.005	.000

Table 4.11: Results for simulation under negative binomial scenario *nbH2*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Negative Binomial CBS* method has the lowest average RMSE (.6162), *DNAcopy* is close (.6205), and the *Poisson CBS* method is not too much higher (.7045). Interestingly here, the *Poisson CBS* method does correctly identify three segments the highest proportion of times (.290) among the three methods.

Next, in scenario *nbF2*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{negative binomial}(r = 10, p = .2), i = 1, \dots, 15 \\ \text{negative binomial}(r = 20, p = .2), i = 16, \dots, 25 \\ \text{negative binomial}(r = 10, p = .2), i = 26, \dots, 50 \end{cases} .$$

Some results are shown in Table 4.12.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.9695	.500	.003	.468	.011	.018	.000
Poisson CBS	.9526	.242	.005	.639	.006	.097	.011
Negative Binomial CBS	.9463	.493	.001	.486	.003	.017	.001

Table 4.12: Results for simulation under negative binomial scenario *nbF2*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Negative Binomial CBS* method has the lowest average RMSE (.9463). Interestingly, the *Poisson CBS* method is next closest (.9526), though *DNAcopy* is also close (.9695). Again it is interesting that the *Poisson CBS* method does correctly identify three segments the highest proportion of times (.639) among the three methods.

Now, we consider scenarios with $p = .2$ where r is larger so the mean is larger. In scenario *nbN3*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$X_i \sim \text{negative binomial}(r = 40, p = .2), i = 1, \ldots, 50.$

Some results are shown in Table 4.13.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	.4832	.948	.010	.039	.000	.003	.000
Poisson CBS	.6785	.822	.008	.157	.001	.010	.002
Negative Binomial CBS	.5026	.934	.005	.060	.000	.000	.001

Table 4.13: Results for simulation under negative binomial scenario *nbN3*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, interestingly *DNAcopy* has the lowest average RMSE (.4832) and the proportion of times it does not correctly identify that there is only one segment ($1 - .948 = .052$) is closest to the nominal size. Again, the overdispersion does appear cause some problems for the *Poisson CBS* method in this setting with $p = .2$, but not nearly as severe as the problems when $p = .8$.

Next, in scenario *nbH3*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{negative binomial}(r = 40, p = .2), i = 1, \ldots, 15 \\ \text{negative binomial}(r = 60, p = .2), i = 16, \ldots, 25 \\ \text{negative binomial}(r = 40, p = .2), i = 26, \ldots, 50 \end{cases} .$$

Some results are shown in Table 4.14.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	1.7902	.395	.002	.547	.018	.037	.001
Poisson CBS	1.7455	.167	.005	.697	.005	.110	.016
Negative Binomial CBS	1.7617	.408	.000	.567	.003	.000	.022

Table 4.14: Results for simulation under negative binomial scenario *nbH3*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, it is interesting that the *Poisson CBS* method has the lowest average RMSE (1.7455) as well as the highest proportion of times with correct identification of three segments (.697). Both the *Negative Binomial CBS* method and *DNAcopy* are close in the average RMSE, though a little conservative with higher percentages of times that only one segment was identified.

Next, in scenario *nbF3*, we generate $R = 1000$ samples of size $n = 50$ where the data is simulated from the model

$$X_i \sim \begin{cases} \text{negative binomial}(r = 40, p = .2), i = 1, \ldots, 15 \\ \text{negative binomial}(r = 80, p = .2), i = 16, \ldots, 25 \\ \text{negative binomial}(r = 40, p = .2), i = 26, \ldots, 50 \end{cases} .$$

Some results are shown in Table 4.15.

Method	AveRMSE	1	2	3	4	5	6+
DNAcopy	1.5869	.007	.000	.900	.046	.045	.002
Poisson CBS	1.6273	.000	.000	.780	.014	.176	.030
Negative Binomial CBS	1.4603	.003	.000	.925	.004	.066	.002

Table 4.15: Results for simulation under negative binomial scenario *nbF3*.

The column labeled AveRMSE reports the average of the values of the RMSE for the 1000 simulated data sets. The last six columns labeled 1, 2, 3, 4, 5, and 6+ report the proportion of simulated data sets for which the specified method identified 1, 2, 3, 4, 5, and more than 5 segments. In this setting, the *Negative Binomial CBS* method clearly does both with having the lowest average RMSE (1.4603) and in correctly identifying three segments the highest proportion of times (.925). Notably, the *Poisson CBS* method overestimates the number of segments a large percentage of times (17.6%).

Overall, for the negative binomial scenarios, the *Negative Binomial CBS* method does best in most scenarios and well in each as expected since the distribution is correctly specified. It seems that it might be a bit conservative in identifying splits in some scenarios, and it is interesting that the *Poisson CBS* or *DNAcopy* do better in a few scenarios. The *Poisson CBS* method does incorrectly specify the distribution – ignroing the overdispersion in the negative binomial random variables – but does not do too badly in the scenarios with $p = .2$ where there is not too much overdispersion. However, in the scenarios with $p = .8$ where there is a large amount of overdispersion, the *Poisson CBS* method drastically overestimates the number of segments.

CHAPTER 5
COMPARISON OF METHODS

In the past few decades, due to the major breakthrough of NGS sequencing technologies, there is a great need for appropriate computational and statistical tools for assessing and validating biological models. As a result of that, better computational methods and more efficient software tools are constantly being developed in recent years. Simulated data is crucial for guiding tool development and evaluating tool performance, and therefore it is essential to develop simulation software that can produce next-generation sequencing reads that captures the most vital characteristics of real data (Huang et al. 2012). Hence, computer simulation of genomic data has become more popular, and many simulation softwares for NGS data analysis have been rising rapidly in the bioinformatics field. These tools have very diverse input requirements and functionalities, which make it quite difficult to choose "What is the most appropriate one for the problem" at hand (Escalona et al. 2016). Some currently available software tools for the simulation of genomic NGS data are, ART (Huang et al. 2012), Wgsim from the Samtools package (Li and Durbin 2009), MetaSim (this can be used for metagenomic data too)(Richter et al. 2008), 454Sim (Lysholm et al. 2011), etc. Almost all these programs work well in their domain. Escalona et al. (2016) reviewed 23 currently available software tools that were either recently published or developed, in most cases still maintained and freely available, for the simulation of genomic NGS data (they focused on the simulation of DNA sequences), and discussed their various features, such as the required input, the interaction with the user, the sequencing platforms, the type of reads, the error models, the possibility of introducing coverage bias, the simulation of genomic variants and the output provided.

In this chapter, we will illustrate in detail, the way we simulate Illumina reads from a NGS read simulator package called "MetaSim", which is mentioned above, the output, and the appearance of the simulated Illumina reads. We also present the detailed explanation about the genomic data we used for the data analysis, such as the appearance of the reference genome, the way we created the target genome and its appearance, etc. Then, we will explain briefly, using BWA (a reads alignment tool) and its output. Finally, we will present the data analysis using the methods similar to those present in Chapter 4.

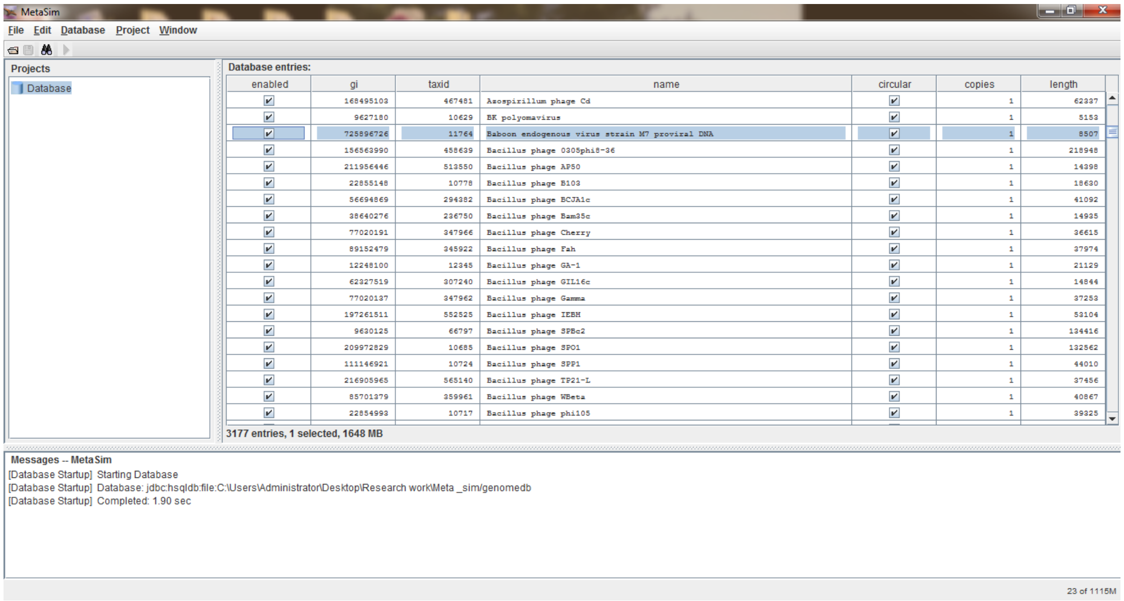
5.1 MetaSim

MetaSim, a sequencing simulator for genomics and metagenomics, was introduced by Daniel H. Huson and Felix Ott, with contributions from Ramona Schmid, Alexander F. Auch and Daniel C. Richter in 2008. The input for MetaSim is a set of known genome sequences (fasta files) and an abundance profile, reflecting (adaptable) error models of current sequencing technologies. MetaSim simulates both Sanger sequencing and Roche's 454 (sequencing-by-synthesis) approach. Additionally, it provides a flexible, empirical error model which is usable to simulate Illumina's short reads, where each read is 36 base pairs long.

MetaSim is written in the programming language Java and it provides versions that run under the Linux, MacOS, Windows and Unix operating systems, and are freely available in their website at: `http://www-ab.informatik.uni-tuebingen.de/software/metasim`. Since MetaSim is written in Java it requires a Java runtime environment version 1.5 or newer, freely available from `www.java.org`. This software package has a user manual, which is easy to understand, and it is freely available online.

5.1.1 Importing Genome Sequences to the Database

Upon first startup of the program, the internal database does not contain any genome sequences. So, the user needs to import the necessary genome sequences into the database. We worked with the virus sequences, and the data was downloaded from the link `ftp://ftp.ncbi.nih.gov/refseq/release/viral/viral1.genomic.fna.gz`, which is given by the MetaSim user manual, and was installed accordingly. Finally, the file `viral1.Genomic.fna.gz` was imported into the database (a screenshot is shown in Figure 5.1) and the baboon endogenous virus strain M7 proviral DNA was chosen as the **Reference Genome**. It consists of 8507 bases. Interestingly, we found that the first 555 bases are repeating at 7953 locus in the reference genome, which implies that, the first 555 bases are same as the last 555 bases in the reference genome. The format of this file is a fasta file, part of which is shown in Figure 5.2.



Database entries:

enabled	gi	taxid	name	circular	copies	length
☑	168495103	467481	Azospirillum phage Cd	☑	1	62337
☑	9627180	10629	BK polyomavirus	☑	1	5153
☑	725896726	11764	Baboon endogenous virus strain M7 proviral DNA	☑	1	8507
☑	156563990	458639	Bacillus phage 0305phi8-36	☑	1	218948
☑	211956446	513550	Bacillus phage AP50	☑	1	14398
☑	22855148	10778	Bacillus phage B103	☑	1	18630
☑	56694869	294382	Bacillus phage BCJA1c	☑	1	41092
☑	38640276	236750	Bacillus phage Bam35c	☑	1	14935
☑	77020191	347966	Bacillus phage Cherry	☑	1	36615
☑	89152479	345922	Bacillus phage Fah	☑	1	37974
☑	12248100	12345	Bacillus phage GA-1	☑	1	21129
☑	62327519	307240	Bacillus phage GIL16c	☑	1	14844
☑	77020137	347962	Bacillus phage Gamma	☑	1	37253
☑	197261511	552525	Bacillus phage IEBH	☑	1	53104
☑	9630125	66797	Bacillus phage SPBc2	☑	1	134416
☑	209972829	10685	Bacillus phage SPO1	☑	1	132562
☑	111146921	10724	Bacillus phage SPP1	☑	1	44010
☑	216905965	565140	Bacillus phage TP21-L	☑	1	37456
☑	85701379	359961	Bacillus phage WBeta	☑	1	40867
☑	22854993	10717	Bacillus phage phi105	☑	1	39325

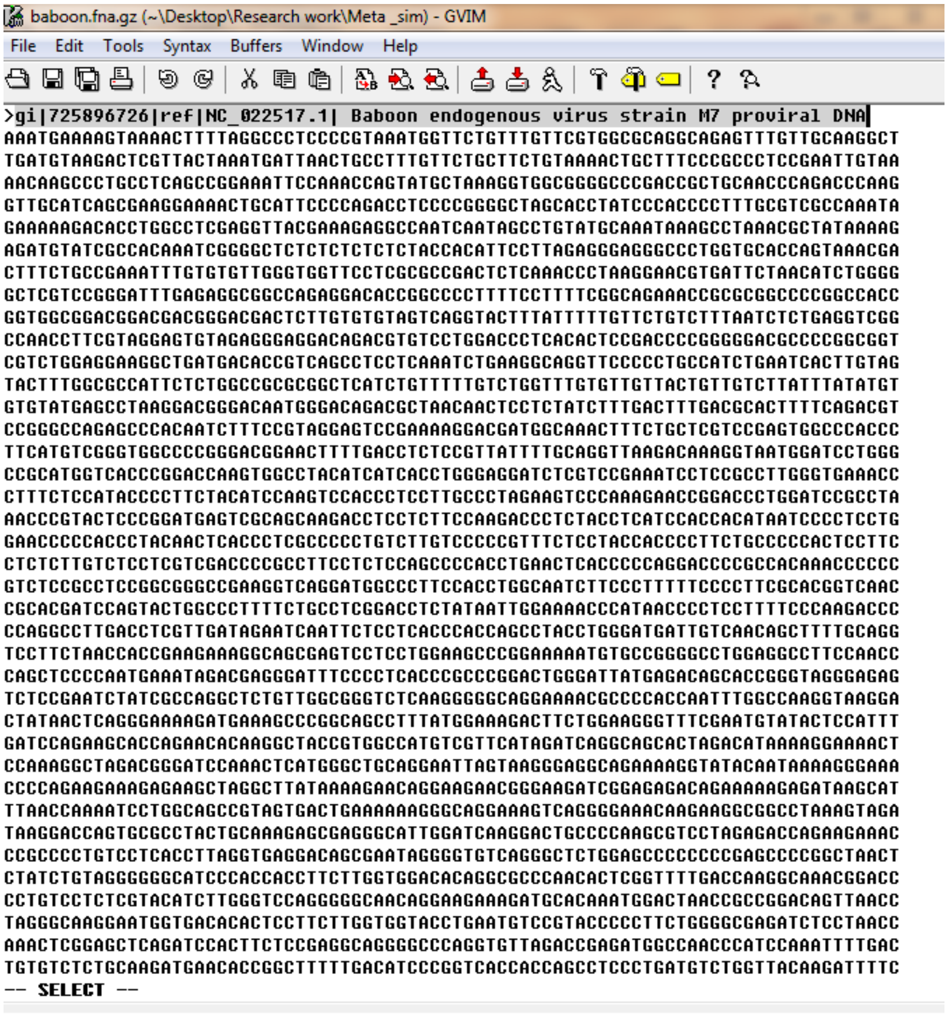
3177 entries, 1 selected, 1648 MB

Messages -- MetaSim

```
[Database Startup]  Starting Database
[Database Startup]  Database: jdbc:hsqldb:file:C:\Users\Administrator\Desktop\Research work\Meta _sim/genomedb
[Database Startup]  Completed: 1.90 sec
```

23 of 1115M

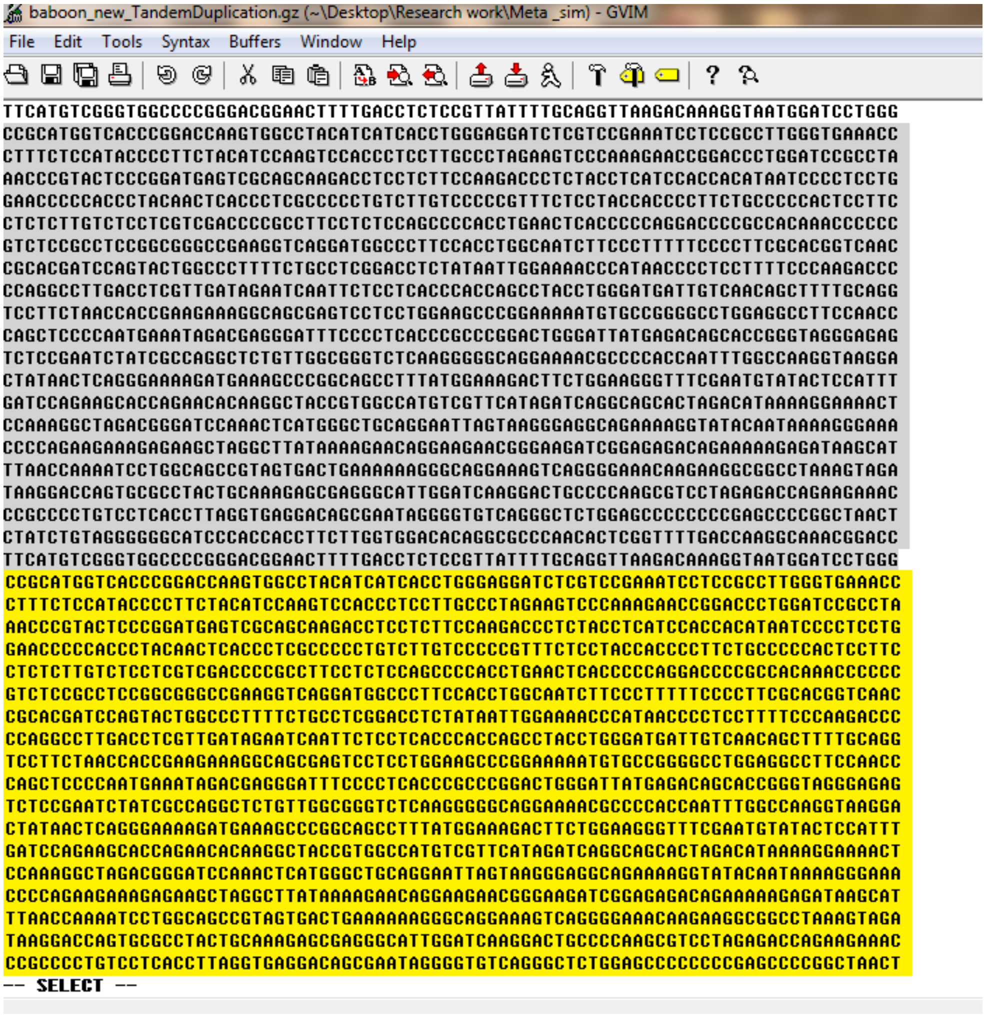
Figure 5.1: The screenshot of baboon endogenous virus in the GUI mode.



```
>gi|725896726|ref|NC_022517.1| Baboon endogenous virus strain M7 proviral DNA
AAATGAAAAGTAAAACTTTTAGGCCCTCCCCGTAAATGGTTCTGTTTGTTCGTGGCGCAGGCAGAGTTTGTTGCAAGGCT
TGATGTAAGACTCGTTACTAAATGATTAACTGCCTTTGTTCTGCTTCTGTAAAACTGCTTTCCCGCCCTCCGAATTGTAA
AACAAGCCCTGCCTCAGCCGGAAATTCCAAACCAGTATGCTAAAGGTGGCGGGGCCCGACCGCTGCAACCCAGACCCAAG
GTTGCATCAGCGAAGGAAAACTGCATTCCCCAGACCTCCCCGGGGCTAGCACCTATCCCACCCCTTTGCGTCGCCAAATA
GAAAAAGACACCTGGCCTCGAGGTTACGAAAGAGGCCAATCAATAGCCTGTATGCAAATAAAGCCTAAACGCTATAAAAG
AGATGTATCGCCACAAATCGGGGCTCTCTCTCTCTACCACATTCCTTAGAGGGAGGGCCCTGGTGCACCAGTAAACGA
CTTTCTGCCGAAATTTGTGTGTTGGGTGGTTCCTCGCGCCGACTCTCAAACCCTAAGGAACGTGATTCTAACATCTGGGG
GCTCGTCCGGGATTTGAGAGGCGGCCAGAGGACACCGGCCCCTTTTCCTTTTCGGCAGAAACCGCGCGGCCCCGGCCACC
GGTGGCGGACGGACGACGGGACGACTCTTGTGTGTAGTCAGGTACTTTATTTTTGTTCTGTCTTTAATCTCTGAGGTCGG
CCAACCTTCGTAGGAGTGTAGAGGGAGGACAGACGTGTCCTGGACCCTCACACTCCGACCCCGGGGGACGCCCCGGCGGT
CGTCTGGAGGAAGGCTGATGACACCGTCAGCCTCCTCAAATCTGAAGGCAGGTTCCCCCTGCCATCTGAATCACTTGTAG
TACTTTGGCGCCATTCTCTGGCCGCGCGGCTCATCTGTTTTTGTCTGGTTTGTGTTGTTACTGTTGTCTTATTTATATGT
GTGTATGAGCCTAAGGACGGGACAATGGGACAGACGCTAACAACTCCTCTATCTTTGACTTTGACGCACTTTTCAGACGT
CCGGGCCAGAGCCCACAATCTTTCCGTAGGAGTCCGAAAAGGACGATGGCAAACTTTCTGCTCGTCCGAGTGGCCCACCC
TTCATGTCGGGTGGCCCCGGGACGGAACTTTTGACCTCTCCGTTATTTTGCAGGTTAAGACAAAGGTAATGGATCCTGGG
CCGCATGGTCACCCGGACCAAGTGGCCTACATCATCACCTGGGAGGATCTCGTCCGAAATCCTCCGCCTTGGGTGAAACC
CTTTCTCCATACCCCTTCTACATCCAAGTCCACCCTCCTTGCCCTAGAAGTCCCAAAGAACCGGACCCTGGATCCGCCTA
AACCCGTACTCCCGGATGAGTCGCAGCAAGACCTCCTCTTCCAAGACCCTCTACCTCATCCACCACATAATCCCCTCCTG
GAACCCCCACCCTACAACTCACCCTCGCCCCCTGTCTTGTCCCCCGTTTCTCCTACCACCCCTTCTGCCCCCACTCCTTC
CTCTCTTGTCTCCTCGTCGACCCCGCCTTCCTCTCCAGCCCCACCTGAACTCACCCCCAGGACCCCGCCACAAACCCCCC
GTCTCCGCCTCCGGCGGGCCGAAGGTCAGGATGGCCCTTCCACCTGGCAATCTTCCCTTTTTCCCCTTCGCACGGTCAAC
CGCACGATCCAGTACTGGCCCTTTTCTGCCTCGGACCTCTATAATTGGAAAACCCATAACCCCTCCTTTTCCCAAGACCC
CCAGGCCTTGACCTCGTTGATAGAATCAATTCTCCTCACCCACCAGCCTACCTGGGATGATTGTCAACAGCTTTTGCAGG
TCCTTCTAACCACCGAAGAAAGGCAGCGAGTCCTCCTGGAAGCCCGGAAAAATGTGCCGGGGCCTGGAGGCCTTCCAACC
CAGCTCCCCAATGAAATAGACGAGGGATTTCCCCTCACCCGCCCGGACTGGGATTATGAGACAGCACCGGGTAGGGAGAG
TCTCCGAATCTATCGCCAGGCTCTGTTGGCGGGTCTCAAGGGGGCAGGAAAACGCCCCACCAATTTGGCCAAGGTAAGGA
CTATAACTCAGGGAAAAGATGAAAGCCCGGCAGCCTTTATGGAAAGACTTCTGGAAGGGTTTCGAATGTATACTCCATTT
GATCCAGAAGCACCAGAACACAAGGCTACCGTGGCCATGTCGTTCATAGATCAGGCAGCACTAGACATAAAAGGAAAACT
CCAAAGGCTAGACGGGATCCAAACTCATGGGCTGCAGGAATTAGTAAGGGAGGCAGAAAAGGTATACAATAAAAGGGAAA
CCCCAGAAGAAAGAGAAGCTAGGCTTATAAAAGAACAGGAAGAACGGGAAGATCGGAGAGACAGAAAAAGAGATAAGCAT
TTAACCAAAATCCTGGCAGCCGTAGTGACTGAAAAAAGGGCAGGAAAGTCAGGGGAAACAAGAAGGCGGCCTAAAGTAGA
TAAGGACCAGTGCGCCTACTGCAAAGAGCGAGGGCATTGGATCAAGGACTGCCCCAAGCGTCCTAGAGACCAGAAGAAAC
CCGCCCCTGTCCTCACCTTAGGTGAGGACAGCGAATAGGGGTGTCAGGGCTCTGGAGCCCCCCCGAGCCCCGGCTAACT
CTATCTGTAGGGGGGCATCCCACCACCTTCTTGGTGGACACAGGCGCCCAACACTCGGTTTTGACCAAGGCAAACGGACC
CCTGTCCTCTCGTACATCTTGGGTCCAGGGGGCAACAGGAAGAAAGATGCACAAATGGACTAACCGCCGGACAGTTAACC
TAGGGCAAGGAATGGTGACACACTCCTTCTTGGTGGTACCTGAATGTCCGTACCCCCTTCTGGGGCGAGATCTCCTAACC
AAACTCGGAGCTCAGATCCACTTCTCCGAGGCAGGGGCCCAGGTGTTAGACCGAGATGGCCAACCCATCCAAATTTTGAC
TGTGTCTCTGCAAGATGAACACCGGCTTTTTGACATCCCGGTCACCACCAGCCTCCCTGATGTCTGGTTACAAGATTTTC
-- SELECT --
```

Figure 5.2: Baboon endogenous virus strain M7 proviral DNA.

Then the target genome was created by adding 20 new sequence lines (1600 bases long) to the reference genome. To do this, we used a software package called Vim text editor. All the bases were copied from the sixteenth sequence line to the thirty-fifth sequence line (which are 1121 – 2720 loci's) and pasted next to it (which are 2721 – 4320 loci's), so that target genome has 10,107 bases, which is partly shown in the Figure 5.3. In this figure, the duplicated region is shown in gray and the new bases that were added are shown in yellow.


baboon_new_TandemDuplication.gz (~\Desktop\Research work\Meta _sim) - GVIM
File
Edit
Tools
Syntax
Buffers
Window
Help

```
TTCATGTCGGGTGGCCCCGGGACGGAACTTTTGACCTCTCCGTTATTTTGCAGGTTAAGACAAAGGTAATGGATCCTGGG
CCGCATGGTCACCCGGACCAAGTGGCCTACATCATCACCTGGGAGGATCTCGTCCGAAATCCTCCGCCTTGGGTGAAACC
CTTTCTCCATACCCCTTCTACATCCAAGTCCACCCTCCTTGCCCTAGAAGTCCCAAAGAACCGGACCCTGGATCCGCCTA
AACCCGTACTCCCGGATGAGTCGCAGCAAGACCTCCTCTTCCAAGACCCTCTACCTCATCCACCACATAATCCCCTCCTG
GAACCCCCACCCTACAACTCACCCTCGCCCCCTGTCTTGTCCCCCGTTTCTCCTACCACCCCTTCTGCCCCCACTCCTTC
CTCTCTTGTCTCCTCGTCGACCCCGCCTTCCTCTCCAGCCCCACCTGAACTCACCCCCAGGACCCCGCCACAAACCCCCC
GTCTCCGCCTCCGGCGGGCCGAAGGTCAGGATGGCCCTTCCACCTGGCAATCTTCCCTTTTTCCCCTTCGCACGGTCAAC
CGCACGATCCAGTACTGGCCCTTTTCTGCCTCGGACCTCTATAATTGGAAAACCCATAACCCCTCCTTTTCCCAAGACCC
CCAGGCCTTGACCTCGTTGATAGAATCAATTCTCCTCACCCACCAGCCTACCTGGGATGATTGTCAACAGCTTTTGCAGG
TCCTTCTAACCACCGAAGAAAGGCAGCGAGTCCTCCTGGAAGCCCGGAAAAATGTGCCGGGGCCTGGAGGCCTTCCAACC
CAGCTCCCCAATGAAATAGACGAGGGATTTCCCCTCACCCGCCCGGACTGGGATTATGAGACAGCACCGGGTAGGGAGAG
TCTCCGAATCTATCGCCAGGCTCTGTTGGCGGGTCTCAAGGGGGCAGGAAAACGCCCCACCAATTTGGCCAAGGTAAGGA
CTATAACTCAGGGAAAAGATGAAAGCCCGGCAGCCTTTATGGAAAGACTTCTGGAAGGGTTTCGAATGTATACTCCATTT
GATCCAGAAGCACCAGAACACAAGGCTACCGTGGCCATGTCGTTCATAGATCAGGCAGCACTAGACATAAAAGGAAAACT
CCAAAGGCTAGACGGGATCCAAACTCATGGGCTGCAGGAATTAGTAAGGGAGGCAGAAAAGGTATACAATAAAAGGGAAA
CCCCAGAAGAAAGAGAAGCTAGGCTTATAAAAGAACAGGAAGAACGGGAAGATCGGAGAGACAGAAAAAGAGATAAGCAT
TTAACCAAAATCCTGGCAGCCGTAGTGACTGAAAAAAGGGCAGGAAAGTCAGGGGAAACAAGAAGGCGGCCTAAAGTAGA
TAAGGACCAGTGCGCCTACTGCAAAGAGCGAGGGCATTGGATCAAGGACTGCCCCAAGCGTCCTAGAGACCAGAAGAAAC
CCGCCCCTGTCCTCACCTTAGGTGAGGACAGCGAATAGGGGTGTCAGGGCTCTGGAGCCCCCCCGAGCCCCGGCTAACT
CTATCTGTAGGGGGGCATCCCACCACCTTCTTGGTGGACACAGGCGCCCAACACTCGGTTTTGACCAAGGCAAACGGACC
TTCATGTCGGGTGGCCCCGGGACGGAACTTTTGACCTCTCCGTTATTTTGCAGGTTAAGACAAAGGTAATGGATCCTGGG
CCGCATGGTCACCCGGACCAAGTGGCCTACATCATCACCTGGGAGGATCTCGTCCGAAATCCTCCGCCTTGGGTGAAACC
CTTTCTCCATACCCCTTCTACATCCAAGTCCACCCTCCTTGCCCTAGAAGTCCCAAAGAACCGGACCCTGGATCCGCCTA
AACCCGTACTCCCGGATGAGTCGCAGCAAGACCTCCTCTTCCAAGACCCTCTACCTCATCCACCACATAATCCCCTCCTG
GAACCCCCACCCTACAACTCACCCTCGCCCCCTGTCTTGTCCCCCGTTTCTCCTACCACCCCTTCTGCCCCCACTCCTTC
CTCTCTTGTCTCCTCGTCGACCCCGCCTTCCTCTCCAGCCCCACCTGAACTCACCCCCAGGACCCCGCCACAAACCCCCC
GTCTCCGCCTCCGGCGGGCCGAAGGTCAGGATGGCCCTTCCACCTGGCAATCTTCCCTTTTTCCCCTTCGCACGGTCAAC
CGCACGATCCAGTACTGGCCCTTTTCTGCCTCGGACCTCTATAATTGGAAAACCCATAACCCCTCCTTTTCCCAAGACCC
CCAGGCCTTGACCTCGTTGATAGAATCAATTCTCCTCACCCACCAGCCTACCTGGGATGATTGTCAACAGCTTTTGCAGG
TCCTTCTAACCACCGAAGAAAGGCAGCGAGTCCTCCTGGAAGCCCGGAAAAATGTGCCGGGGCCTGGAGGCCTTCCAACC
CAGCTCCCCAATGAAATAGACGAGGGATTTCCCCTCACCCGCCCGGACTGGGATTATGAGACAGCACCGGGTAGGGAGAG
TCTCCGAATCTATCGCCAGGCTCTGTTGGCGGGTCTCAAGGGGGCAGGAAAACGCCCCACCAATTTGGCCAAGGTAAGGA
CTATAACTCAGGGAAAAGATGAAAGCCCGGCAGCCTTTATGGAAAGACTTCTGGAAGGGTTTCGAATGTATACTCCATTT
GATCCAGAAGCACCAGAACACAAGGCTACCGTGGCCATGTCGTTCATAGATCAGGCAGCACTAGACATAAAAGGAAAACT
CCAAAGGCTAGACGGGATCCAAACTCATGGGCTGCAGGAATTAGTAAGGGAGGCAGAAAAGGTATACAATAAAAGGGAAA
CCCCAGAAGAAAGAGAAGCTAGGCTTATAAAAGAACAGGAAGAACGGGAAGATCGGAGAGACAGAAAAAGAGATAAGCAT
TTAACCAAAATCCTGGCAGCCGTAGTGACTGAAAAAAGGGCAGGAAAGTCAGGGGAAACAAGAAGGCGGCCTAAAGTAGA
TAAGGACCAGTGCGCCTACTGCAAAGAGCGAGGGCATTGGATCAAGGACTGCCCCAAGCGTCCTAGAGACCAGAAGAAAC
CCGCCCCTGTCCTCACCTTAGGTGAGGACAGCGAATAGGGGTGTCAGGGCTCTGGAGCCCCCCCGAGCCCCGGCTAACT
-- SELECT --
```

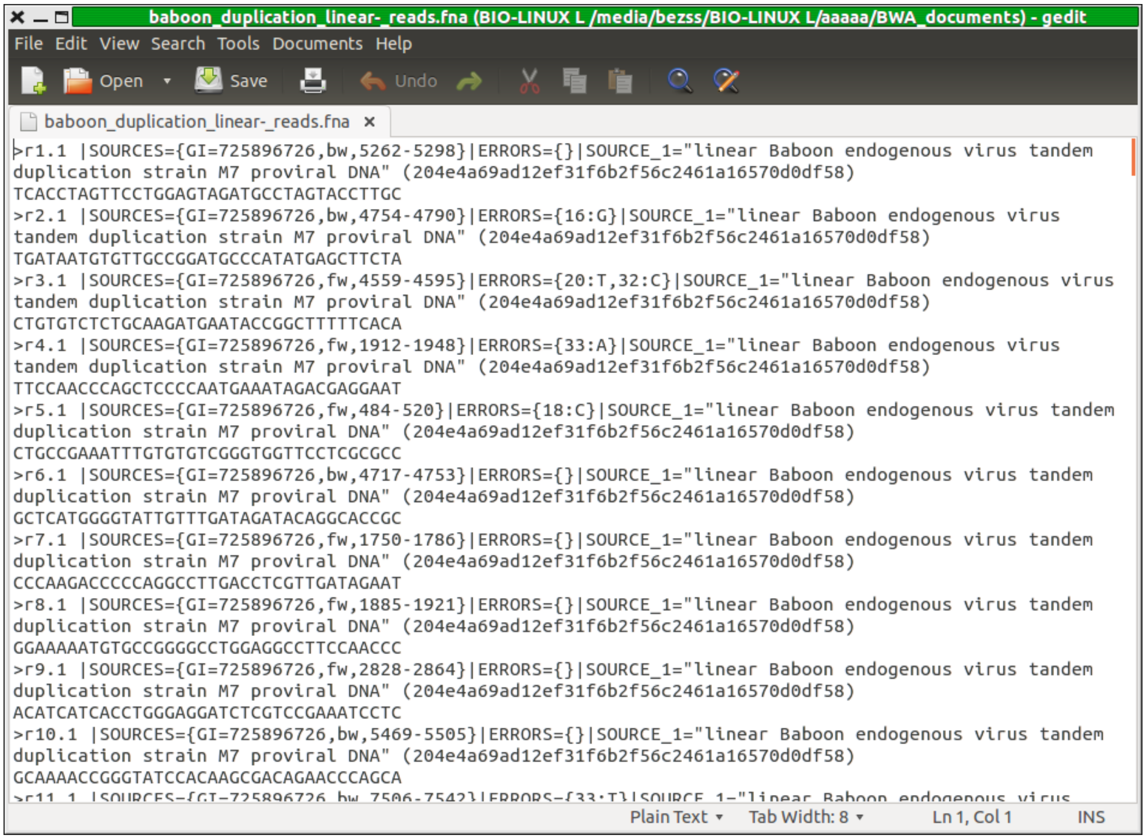
Figure 5.3: Linear baboon endogenous virus tandem duplication strain M7 proviral DNA.

After creating the target genome, it was imported into the database, and 10,000 Illumina short reads were simulated using the empirical error model.

5.1.2 Simulation of Illumina Short Reads

Once sequences are loaded into the database, we need to create a new project, in order to simulate the Illumina short reads. MetaSim seems to consider each input sequence to be circular. So, this may result in MetaSim simulating a read, which has a part (this may either be the first part or the last part) of the 36 bases in the read corresponding to the started bases from the source genome sequence and the remaining part (the last part or the first part) of the read sequence from the end bases in the source genome sequence. Hence, in the alignment process, the alignment software tools might not be able to recognize the best align position of this type of read to the reference genome, and it will fall into the set of unmapped reads. As a consequence of this, we will have data loss, since most alignment software tools (such as Bowtie, BWA, etc.) allow less than 5 mismatches while a few alignment software tools (such as PerM, RMAP, etc.) allow more than 5 mismatches by default for short reads. Therefore, in order to eliminate some data loss, we simulated 10,000 Illumina short reads, which is 36 bases long, by considering the genome sequence to be linear.

MetaSim uses fixed probabilities of sequencing errors (insertions, deletions and substitutions) for the same base in different reads, in a single run (Jia et al. 2013). Figure 5.4 shows a screenshot of part of the fasta file with the short reads produced by MetaSim.


baboon_duplication_linear-_reads.fna (BIO-LINUX L /media/bezss/BIO-LINUX L/aaaaa/BWA_documents) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
baboon_duplication_linear-_reads.fna
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

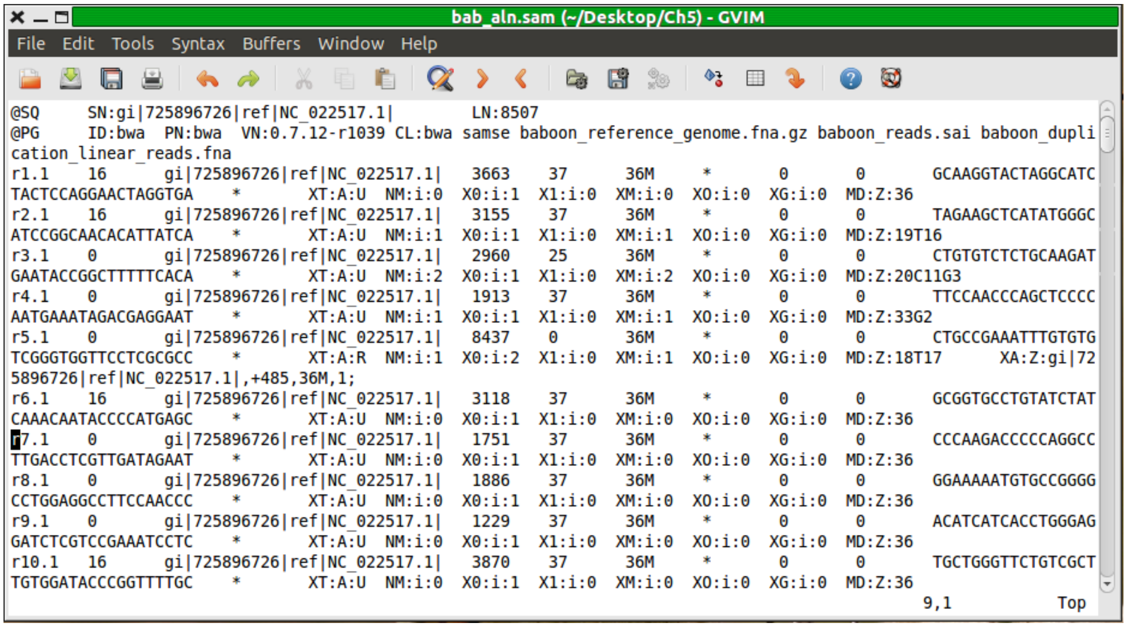
```
>r1.1 |SOURCES={GI=725896726,bw,5262-5298}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
TCACCTAGTTCCTGGAGTAGATGCCTAGTACCTTGC
>r2.1 |SOURCES={GI=725896726,bw,4754-4790}|ERRORS={16:G}|SOURCE_1="linear Baboon endogenous virus
tandem duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
TGATAATGTGTTGCCGGATGCCCATATGAGCTTCTA
>r3.1 |SOURCES={GI=725896726,fw,4559-4595}|ERRORS={20:T,32:C}|SOURCE_1="linear Baboon endogenous virus
tandem duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
CTGTGTCTCTGCAAGATGAATACCGGCTTTTTCACA
>r4.1 |SOURCES={GI=725896726,fw,1912-1948}|ERRORS={33:A}|SOURCE_1="linear Baboon endogenous virus
tandem duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
TTCCAACCCAGCTCCCCAATGAAATAGACGAGGAAT
>r5.1 |SOURCES={GI=725896726,fw,484-520}|ERRORS={18:C}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
CTGCCGAAATTTGTGTGTCGGGTGGTTCCTCGCGCC
>r6.1 |SOURCES={GI=725896726,bw,4717-4753}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
GCTCATGGGGTATTGTTTGATAGATACAGGCACCGC
>r7.1 |SOURCES={GI=725896726,fw,1750-1786}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
CCCAAGACCCCCAGGCCTTGACCTCGTTGATAGAAT
>r8.1 |SOURCES={GI=725896726,fw,1885-1921}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
GGAAAAATGTGCCGGGGCCTGGAGGCCTTCCAACCC
>r9.1 |SOURCES={GI=725896726,fw,2828-2864}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
ACATCATCACCTGGGAGGATCTCGTCCGAAATCCTC
>r10.1 |SOURCES={GI=725896726,bw,5469-5505}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
GCAAAACCGGGTATCCACAAGCGACAGAACCCAGCA
>r11.1 |SOURCES={GI=725896726,bw,7506-7542}|ERRORS={33:T}|SOURCE_1="linear Baboon endogenous virus
```

Figure 5.4: Screenshot of fasta file with short reads.

This simulation generated 5600 substitutions, and no insertions or deletions. The detailed information related to this simulation and its output is in the Appendix. Then using the BWA, we aligned the simulated reads to the reference genome.

5.1.3 Aligning Reads to the Reference Genome

We used BWA-backtrack algorithm to align the reads to the reference genome, because it is designed for the Illumina short reads up to 100bp and used Bio-linux (Field et al. 2006) to run the BWA. The codes we ran and the detailed explanation of the BWA output are shown in the Appendix. BWA output is shown in the Figure 5.5.


bab_aln.sam (~/Desktop/Ch5) - GVIM
File Edit Tools Syntax Buffers Window Help

```
@SQ     SN:gi|725896726|ref|NC_022517.1|        LN:8507
@PG     ID:bwa  PN:bwa  VN:0.7.12-r1039 CL:bwa samse baboon_reference_genome.fna.gz baboon_reads.sai baboon_dupli
cation_linear_reads.fna
r1.1    16      gi|725896726|ref|NC_022517.1|   3663    37      36M     *       0       0       GCAAGGTACTAGGCATC
TACTCCAGGAACTAGGTGA     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r2.1    16      gi|725896726|ref|NC_022517.1|   3155    37      36M     *       0       0       TAGAAGCTCATATGGGC
ATCCGGCAACACATTATCA     *       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:19T16
r3.1    0       gi|725896726|ref|NC_022517.1|   2960    25      36M     *       0       0       CTGTGTCTCTGCAAGAT
GAATACCGGCTTTTTCACA     *       XT:A:U  NM:i:2  X0:i:1  X1:i:0  XM:i:2  XO:i:0  XG:i:0  MD:Z:20C11G3
r4.1    0       gi|725896726|ref|NC_022517.1|   1913    37      36M     *       0       0       TTCCAACCCAGCTCCCC
AATGAAATAGACGAGGAAT     *       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:33G2
r5.1    0       gi|725896726|ref|NC_022517.1|   8437    0       36M     *       0       0       CTGCCGAAATTTGTGTG
TCGGGTGGTTCCTCGCGCC     *       XT:A:R  NM:i:1  X0:i:2  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:18T17       XA:Z:gi|72
5896726|ref|NC_022517.1|,+485,36M,1;
r6.1    16      gi|725896726|ref|NC_022517.1|   3118    37      36M     *       0       0       GCGGTGCCTGTATCTAT
CAAACAATACCCCATGAGC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r7.1    0       gi|725896726|ref|NC_022517.1|   1751    37      36M     *       0       0       CCCAAGACCCCCAGGCC
TTGACCTCGTTGATAGAAT     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r8.1    0       gi|725896726|ref|NC_022517.1|   1886    37      36M     *       0       0       GGAAAAATGTGCCGGGG
CCTGGAGGCCTTCCAACCC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r9.1    0       gi|725896726|ref|NC_022517.1|   1229    37      36M     *       0       0       ACATCATCACCTGGGAG
GATCTCGTCCGAAATCCTC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r10.1   16      gi|725896726|ref|NC_022517.1|   3870    37      36M     *       0       0       TGCTGGGTTCTGTCGCT
TGTGGATACCCGGTTTTGC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
                                                                                                9,1           Top
```

Figure 5.5: Screenshot of sam file with aligned reads.

5.2 Copy Number Analysis of MetaSim Data

The read count data can be obtained from the sam file by extracting the 4th column with the starting position of each read. These values are then tabulated (with 0's exlcuded) to obtain the read counts. Note that we only use the starting position so each read is only counted once in the data set as opposed to "piling up" the reads to count the number of reads that overlap each position as is sometimes done with this type of data. The problem with the pile-up approach is that the assumption of independence is violated.

The number of reads starting at each genomic position are plotted in Figure 5.6. The horizontal axis gives the genomic position. The vertical axis gives the number of reads that start at each position. The points are plotted using R's `jitter` function so that each band corresponds to a single integer value, but a small error is added to the vertical coordinate of each point to make it easier to visualize the number of points corresponding to each of the read count values.

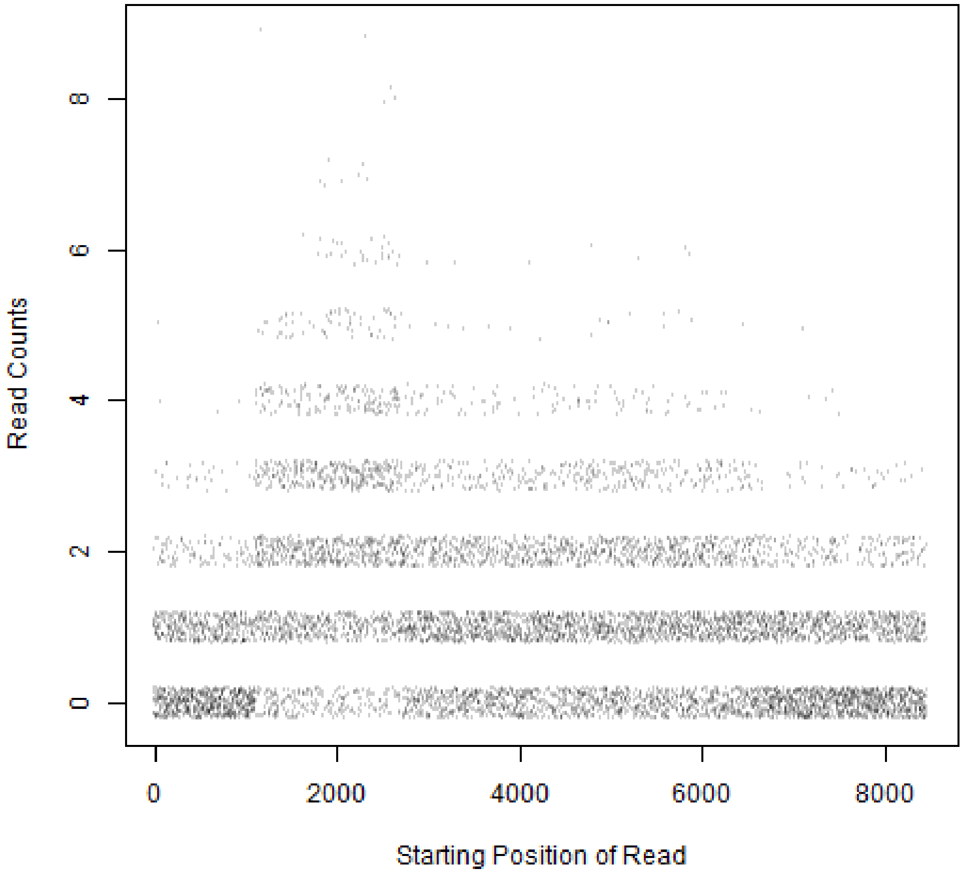


Figure 5.6: Number of reads starting at each genomic position.

First, we analyzed the read count data with the *DNAcopy* method using its defaults except the size `alpha=0.05`. The results are shown in Table 5.1. The first column gives the estimated segment endpoints and the second column gives the mean read count in each interval. This information is provided in the output of the `segment` function of the `DNAcopy` package.

Segment	Mean of read counts in segment
$[1, 1119]$	0.573727
$[1120, 1817]$	1.823782
$[1818, 2688]$	2.328358
$[2689, 2732]$	1.613636
$[2733, 6555]$	1.199581
$[6556, 8472]$	0.626500

Table 5.1: Model estimates for MetaSim simulated reads based on *DNAcopy*.

Here, it appears that the mean read counts at locations on the intervals at each end of the target genome ($[1, 1119]$ and $[6556, 8472]$) is about half of what it is on the largest continuous segment $[2733, 6555]$. There are three segments ($[1120, 1817]$, $[1818, 2688]$, $[2689, 2732]$) in the vicinity of the created duplication, and the mean read count in the middle interval $[1818, 2688]$ is about twice what it is in $[2733, 6555]$ while the mean read counts on the other two intervals $[1120, 1817]$ and $[2689, 2732]$ are about 3/2 that of the interval $[2733, 6555]$; possibly, the mean in $[2689, 2732]$ is further from around 1.8 since it is a shorter interval.

Next, we want to use the methods based on likelihood ratio tests for the Poisson model and the negative binomial model. However, due to the larger number of bases in the reference genome, it is very time consuming to consider all possible pairs of locations as possible endpoints for segments. As an alternative for this example, we propose the following approach. Split the set of all $8507 - 35 = 8472$ possible starting genomic positions in the reference genome into 169 bins where the bin assignment for the ith position is computed using the formula

$$1 + \left\lfloor \frac{169i}{8473} \right\rfloor .$$

Then we first apply the method to the counts in each bin and find the bins which are on the boundaries between two segments. Our method then considers only the positions in these bins as candidates for endpoints of the segments when the method is applied to the read counts at each genomic position.

The results for analyzing the aggregated data with the *Poisson CBS* method are shown in Table 5.2. The first column gives the segment endpoints in terms of the bin numbers and the second column gives the corresponding loci for the genomic positions in the reference genome for the bins. The third column gives the estimated parameter for each group of bins determined by the CBS method. The fourth column gives the mean number of reads in each segment of bins.

Segment	Loci	$\hat{\mu}$	Mean reads per bin in segment
$[1, 22]$	1–1102	29.68421	28.50000
$[23, 36]$	1103–1804	90.13333	90.64286
$[37, 54]$	1805–2707	114.9444	114.9444
$\{55\}$	2708–2757	90.13333	83.00000
$[56, 131]$	2758–6567	59.97368	59.97368
$[132, 134]$	6568–6718	43.33333	43.33333
$[135, 169]$	6719–8472	29.68421	30.42857

Table 5.2: Model estimates for aggregated reads based on *Poisson CBS*.

Now, we create a list of candidates for possible positions where the copy number changes based on the *Poisson CBS* method. The bins on the boundaries based on the aggregated data are 22, 23, 36, 37, 54, 55, 56, 131, 132, 134, and 135 which correspond to loci 1053–1102, 1103–1153, 1755–1804, 1805–1855, 2658–2707, 2708–2757, 2758–2807, 6518–6567, 6568–6617, 6669–6718, and 6719–6768, respectively.

Then the next step applied the modified version of the *Poisson CBS* method in which α and β is restricted to the above loci. Based on this restriction (both in computing the observed likelihood and the bootstraped likelihoods), the results of this CBS procedure are shown in Table 5.3.

Segment	$\hat{\mu}$	Mean of read counts in segment
$[1, 1106]$	0.590956	0.566908
$[1107, 1119]$	1.203560	1.153846
$[1120, 1136]$	2.325112	2.294118
$[1137, 1817]$	1.812041	1.812041
$[1818, 2692]$	2.325112	2.325714
$[2693, 6555]$	1.203560	1.203728
$[6556, 6566]$	0.590956	0.454546
$[6567, 6692]$	0.920635	0.920635
$[6693, 8472]$	0.590956	0.606742

Table 5.3: Model estimates for MetaSim simulated reads based on *Poisson CBS*.

Again, the segments on each end of the target genome ($[1, 1106]$ and $[6556, 6566] \cup [6693, 8472]$) have a mean read count about half of what is seen in the largest continuous segment $[2693, 6555]$, though the loci in these interval differ slightly. Unlike the *DNAcopy* method, the small segment $[1107, 1119]$ is also grouped with the largest continuous segment by *Poisson CBS*. Like *DNAcopy*, there are three segments in the vicinity of the created duplication, but the middle one $[1137, 1817]$ has mean read count about $3/2$ of what it is in $[2693, 6555]$ while the other two intervals $[1120, 1136]$ and $[1818, 2692]$ have mean read counts about twice of what it is in $[2693, 6555]$. There is a small additional segment $[6567, 6692]$ detected to have a slightly larger mean inside $[6556, 8472]$.

Next, we repeat the two-step CBS procedure with the negative binomial model. Table 5.4 shows the results of applying the *Negative Binomial CBS* method to the aggregated data. The first column gives the segment endpoints in terms of the bin numbers and the second column gives the corresponding loci for the genomic positions in the reference genome for the bins. The third and fourth columns give the estimated parameters r and p for each group of bins determined by the CBS method; when these values do not exist, the MLE $\hat{\mu}$ of the Poisson model is given which corresponds to the maximizer of the likelihood function of the negative binomial model. The fifth column gives the mean number of reads in each segment of bins.

Segment	Loci	$\hat{r}$	$\hat{p}$	Mean reads per bin in segment
$[1, 22]$	1–1102	$\hat{\mu} = 29.86207$		28.50000
$[23, 36]$	1103–1804	1572.767	0.054203	90.64286
$[37, 54]$	1805–2707	$\hat{\mu} = 114.9444$		114.9444
$\{55\}$	2708–2757	1572.767	0.054203	83.00000
$[56, 133]$	2758–6668	295.6033	0.167767	59.58974
$[134, 169]$	6669–8472	$\hat{\mu} = 29.86207$		30.69444

Table 5.4: Model estimates for aggregated reads based on *Negative Binomial CBS*.

The estimated segments for the aggregated data using the *Negative Binomial CBS* method is very similar to the results in Table 5.2 based on the *Poisson CBS* method, and only the last three segments are slightly different. Interestingly, most of the parameter estimates for the negative binomial distributions do not exist (i.e., they correspond to $r \to \infty$, thus reverting back to Poisson distribution estimates). The bins on the boundaries based on the aggregated data using the *Negative Binomial CBS* method are 22, 23, 36, 37, 54, 55, 56, 133, and 134 which correspond to loci 1053–1102, 1103–1153, 1755–1804, 1805–1855, 2658–2707, 2708–2757, 2758–2807, 6618–6668, and 6669–6718, respectively.

Then the next step applied the modified version of the *Negative Binomial CBS* method in which α and β is restricted to the above loci. Based on this restriction (both in computing the observed likelihood and the bootstraped likelihoods), the results of this CBS procedure are shown in Table 5.5.

Segment	$\hat{r}$	$\hat{p}$	Mean of read counts in segment
$[1, 1106]$	$\hat{\mu} = 0.600000$		0.566908
$[1107, 1119]$	$\hat{\mu} = 1.197512$		1.153846
$[1120, 1136]$	54.57374	0.040864	2.294118
$[1137, 1817]$	135.3623	0.013210	1.812041
$[1818, 2692]$	54.57374	0.040864	2.325714
$[2693, 6618]$	$\hat{\mu} = 1.197512$		1.197657
$[6619, 8472]$	$\hat{\mu} = 0.600000$		0.619741

Table 5.5: Model estimates for MetaSim simulated reads based on *Negative Binomial CBS*.

Again, the results based on the negative binomial model are very similar to that of the Poisson model except for differences in $[6556, 6693]$. The fitted means based on all three methods are shown in Figure 5.7. All of the statements about the relative sizes of the mean read counts for the segments are similar to those for the Poisson model. Interestingly, again several of the parameter estimates for the negative binomial distributions do not exist and the other estimates of p are close to 0, so it seems natural that the results are very similar.

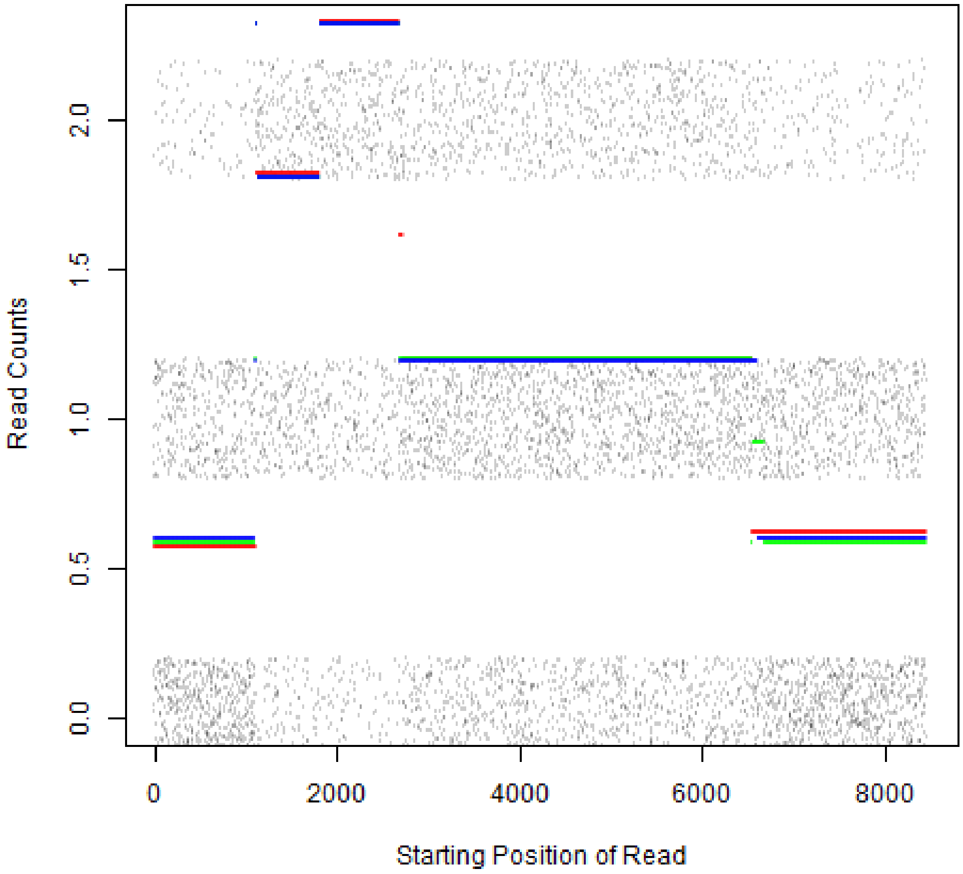


Figure 5.7: Fitted mean values for the *DNAcopy*, *Poisson CBS*, and *Negative Binomial CBS* methods.

CHAPTER 6
CONCLUSIONS

6.1 Conclussion

In the past literature, many authors mentioned that DNA sequencing reads obtained through NGS technologies have slight overdispersion that is not accounted for by the Poisson distribution. Hence, the negative binomial distribution is considered for the read count data. It is known that, a major form of negative binomial distribution, which is a Poisson-gamma mixture, can be considered as an appropriate distribution for genomic data. So, we described the negative binomial distribution as a Poisson-gamma mixture and derived the MLEs for the parameters of negative binomial distribution, r and p. First we derived the MLE for p as a function of r, and then we tried to solve the derivative of profile log likelihood function with respect to r, which is equation (3.6), using the Newton-Raphson method. After some simulation studies, we found that, equation (3.6) has a unique root, when the sample mean ($\bar{x}$) is less than the sample variance ($\hat{\sigma}^2$). On the other hand, if the sample mean is greater than or equal to the sample variance, it showed that the equation (3.6) has no root. We found that it agreed with what was found by some authors in the literature, though there were some conflicting statements about what has been proven. Moreover we proved the limiting behavior of the function $f(r)$ when the 1st condition, which is $\bar{x} < \hat{\sigma}^2$ holds. Simonsen (1976), considered the equation (3.6), and proved that it has no solution, if $k = 1$, or if $k > 2$ and

$m^2 > 2\mathcal{S}$, whereas equation (3.6) has a unique solution, if $k > 2$ and $m^2 < 2\mathcal{S}$, where $k = \max(X_1, X_2, \ldots, X_n)$, m is the sample mean, and $\mathcal{S}$ is defined in Theorem 3.1. We found the relation between $\mathcal{S}$ and the sample variance $\hat{\sigma}^2$ and that led us to prove the relation we found in the simulation studies in Chapter 3, which helps clarify confusing issues in the literature. Hence, if the 1st condition holds, then we proved that unique MLE exists for p and r, otherwise there is no maximizer for equation (3.2). We also proved additional results about the shape of the profile likelihood function for the negative binomial distribution. These results included information about the second derivative of $f(r)$ which allowed us to prove that, if the starting point of the Newton-Raphson is selected appropriately, and the MLE exists, then the Newton-Raphson converges to the true MLE.

Many researchers published articles, considering the Poisson distribution as another appropriate statistical distribution for genomic data for finding CNVs. So, we also considered the Poisson distribution and determined the MLE of its parameter mean. Another important fact that we looked at is, we examined the behavior of negative binomial distribution as the parameter (r) goes to infinity, whereas the probability of success (p) goes to zero. We found that, the probability distribution of negative binomial distribution in this situation approaches the Poisson distribution.

We considered the CNV detection problem by starting with a simple parametric change point model. We estimated the MLEs of the means of each of the continuous segments, and locations where the changes occur. Then we derived the MLEs and found the supremum of the likelihood functions for these simple change point models (with the Poisson and negative binomial distributions). Then to estimate the parameters and the change point locations for the full CNV detection problem, we applied the Circular Binary Segmentation procedure. We proposed the Likelihood Ratio test with parametric bootstrap, and we applied it to find the means and the change point locations in our parametric models for count data. We carried out simulation studies using our new CBS procedures based on count data assuming Poisson and negative binomial models. We compared our method with `DNAcopy` package, which is a well known package that uses CBS method to detect the CNVs. Our method is applied for data, which are simulated from either a Poisson distribution or a negative binomial distribution. In each setting, we varied the parameters to correspond to the true model having one segment or having 3 segments and, for the models that had 3 segments, the mean of the outside segments are equal to each other and the mean of the middle segment is either 1.5 or 2 times that of the outside segments. Overall, the *Negative Binomial CBS* method does well under all scenarios. The *DNAcopy* method also appeared to be reasonable, but in most cases, had a higher average RMSE than the *Negative Binomial CBS*. As expected, the *Poisson CBS* method works well when the count data follows a Poisson distribution, but in some cases, significantly overestimated the number of change points when there is a large amount of overdispersion.

We also analyzed read count data generated using the NGS Illumina read simulator "MetaSim" based on a baboon endogenous virus genome. This genome is considered the reference genome, and then we created a target genome by adding 1600 bases to the reference genome. The reads obtained from MetaSim were aligned using BWA. Then count data was obtained, and also the CBS procedures were applied to analyze this data. All three CBS procedures identified the main change points that we artificially created, but it is interesting to note that each of the procedures also found some other additional segments with different means.

6.2 Discussion: Advantages and Drawbacks

There are some advantages in our newly proposed method. The methods presented here are designed for models using discrete distributions, such as the

Poisson and negative binomial, on count data as opposed to methods designed based on normal data. Furthermore, the circular binary segmentation algorithm is ideally designed to be able to find segments with different parameter values in the middle of the genome, even when the segment is short relative to the length of the entire genome.

The `DNAcopy` package uses the test statistic which is proposed by Venkatraman and Olshen (2007). The test statistic they proposed there is similar to the original CBS, which is explained in the section 4.2, to detect the change point, but modifies the procedure to determine whether the change points are statistically significant. They used the two sample t-statistic, which works better for the data drawn from a normal distribution or data being close to normal. Also, they mentioned that the two sample t-statistics also works, when the data are not normal, *but only if the underlying distribution is not higly skewed* (Venkatraman and Olshen 2007). It is also important to note that, the two sample t-statistic used in the CBS uses the pooled variance, which is the Mean Squared Error(MSE) that they mentioned in the 2007 paper. This indicates that, they assumed the two population variances are to be equal or nearly equal. If this is not the case, then it may not find the true changes in the model. Our model might work well in this situation, since we are only considering the likelihood functions in the corresponding segments. We have implemented the CBS approach using likelihood ratio tests based on discrete distributions that are more appropriate for NGS count data.

Also, in analyzing NGS data, we use only the starting position for each read instead of the pile-up approach where all reads which include the position are counted. This avoids problems for the pile-up approach with its violation of the assumption of independence.

A major drawback of our method is, it takes really large computational time when analyzing a larger data set, especially for the *Negative Binomial CBS* method compared with the refinements made in the `DNAcopy` package. As an alternative for this, we aggregated the data using a bin concept and obtained a list of candidate values for change points and then refined on the next step. Instead of counting the number of whole reads that mapped to each bin (that is what normally people do when considering the bin concept), we were interested in counting the number of starting positions of each read that mapped to each bin. This helps in great deal to avoid the situation, such that a read overlap with two adjacent bins. Then some natural questions that arise in here like, "Do we need to discard that read?" or "Which bin does the read belong to?". If we apply this situation for our case, that is, even though a read overlap with two adjacent bins, it will not have an effect here, since, if we are interested in the starting position, then the read belongs to either one of them. If we find a change, to check where it actually occurs, (that is, to avoid the confusion whether it occurs on the boundary of the bin or anywhere inside the bin) we carried out our method for all the data inside the bin.

Now another question will arise here is, if the change occurs inside the bin, then how can we compute the mean of the read counts in the bin? If the data come from a Poisson distribution then the sum of Poisson variables is Poisson, and then the mean can be computed and method applied accordingly. If the data come from a negative binomial distribution having the same probability of success, say p, then it would be easy to estimate the parameters, which is the parameters of a negative binomial distribution having parameters $r = r_1 + \ldots + r_m$ and p. But the problem now here is, what if we have a different probability of success? How can we properly estimate the parameters and then estimate the mean of each segment? So, we need to improve our method by focusing on these concepts.

Another drawback to keep in mind when using CBS approaches for the multiple change point problem is that the successive tests in the CBS approach are not independent. This is also a general problem for binary segmentation procedures, and an other alternative based on hypothesis testing such as backward search algorithms are even more time consuming and have similar problems with dependence on successive tests.

6.3 Future Work

Even though we developed novel methods to address a CNV detection problem, there are many questions that need to be answered as discussed above, which could not be examined with great attention in this thesis. Therefore, we need to look for further improvement of our method in the future. The following discussion illustrates the different directions we could look for further improvement of our method.

The most critical issue is reducing computation time so future research on these methods should explore approximations for the MLE by not including all possible endpoints, exploring possible update formulas to reduce computation times on successive steps, and/or obtaining the asymptotic distribution of the test statistic or finding the limiting value of the tail probability for the test statistic so we can obtain critical values for the tests (replacing the need for the bootstrap or permutation tests).

Another direction is improving the bin method for larger data sets, such as human chromosomes, by increasing the size of the bin. Then number of read counts mapped to each bin is considerably large. So, we could then approximate the distribution of read counts per bin with the normal distribution using the central limit theorem. Then we can avoid almost all the problems that we discussed in our method. But, still there is a question open to be answered for having a smaller bin size. Additionally, it is important to further study the effective of the bin size in the bin method to find an optimal balance of accuracy and compuation time.

Finally, it might be of interest to study cases where the underlying model exhibits underdispersion and study approaches in alternative models for this case.

REFERENCES

[1] F. J. Anscombe, *Sampling theory of the negative binomial and logarithmic series distributions*, Biometrika **37** (1950), no. 3/4, 358–382.

[2] O. T. Avery, C. M. MacLeod, and M. McCarty, *Studies on the chemical nature of the substance inducing transformation of pneumococcal types: induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III*, The Journal of Experimental Medicine **79** (1944), no. 2, 137.

[3] M. A. Bender and M. A. Kastenbaum, *Statistical analysis of the normal human karyotype*, Am J Human Genetics **21** (1969), 322–351.

[4] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, et al., *Accurate whole human genome sequencing using reversible terminator chemistry*, Nature **456** (2008), 53–59.

[5] N. P. Carter, *Methods and strategies for analyzing copy number variation using DNA microarrays*, Nature Genetics **39** (2007), 16–21.

[6] V. Chaitankar, G. Karakülah, R. Ratnapriya, F. O Giuste, M. J. Brooks, and A. Swaroop, *Next generation sequencing technology and genomewide data analysis: Perspectives for retinal research*, Progress in Retinal and Eye Research **55** (2016), 1–31.

[7] E. Chargaff, *Chemical specificity of nucleic acids and mechanism of their enzymatic degradation*, Experientia **6** (1950), 201–209.

[8] ———, *Some recent studies on the composition and structure of nucleic acids*, Journal of Cellular and Comparative Physiology **38** (1951), no. S1, 41–59.

[9] E. Chargaff, E. Vischer, R. Doniger, C. Green, and F. Misani, *The composition of the desoxypentose nucleic acids of thymus and spleen*, Journal of Biological Chemistry **177** (1949), no. 1, 405–416.

[10] E. Check, *Human genome: patchwork people*, Nature **437** (2005), no. 7062, 1084–1086.

[11] J. Chen and A. K. Gupta, *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*, Springer Science & Business Media, 2011.

[12] K. Chen, J. W. Wallis, M. D. McLellan, D. E. Larson, J. M. Kalicki, C. S. Pohl, S. D. McGrath, M. C. Wendl, Q. Y. Zhang, and D. P. Locke, *BreakDancer: an algorithm for high-resolution mapping of genomic structural variation*, Nature Methods **6** (2009), 677–681.

[13] Y. Chen, L. Zhao, Y. Wang, M. Cao, V. Gelowani, M. Xu, S. A. Agrawal, Y. Li, S. P. Daiger, R. Gibbs, F. Wang, and R. Chen, *SeqCNV: a novel method for identification of copy number variations in targeted next-generation sequencing data*, BMC Bioinformatics **18** (2017), no. 1, 147.

[14] S. Clancy, *Copy number variation*, Nature Education **1** (2008), no. 1, 95.

[15] R. Dahm, *Discovering DNA: Friedrich miescher and the early years of nucleic acid research*, Human Genetics **122** (2008), no. 6, 565–581.

[16] H. Dai, Y. Bao, and M. Bao, *Maximum likelihood estimate for the dispersion parameter of the negative binomial distribution*, Statistics & Probability Letters **83** (2013), 21–27.

[17] K. Dong, H. Zhao, T. Tong, and X. Wan, *NBLDA: negative binomial linear discriminant analysis for RNA-Seq data*, BMC Bioinformatics **17** (2016), 369.

[18] J. Duan, J. G. Zhang, H. W. Deng, and Y. P. Wang, *CNV-TV: A robust method to discover copy number variation from short sequencing reads*, BMC Bioinformatics **14** (2013), no. 1, 1.

[19] ______, *Comparative studies of copy number variation detection methods for next-generation sequencing technologies*, PloS ONE **8** (2013), no. 3, e59128.

[20] C. Erdman and J. W. Emerson, *A fast bayesian change point analysis for the segmentation of microarray data*, Bioinformatics **24** (2008), no. 19, 2143–2148.

[21] M. Escalona, S. Rocha, and D. Posada, *A comparison of tools for the simulation of genomic next-generation sequencing data*, Nature Reviews Genetics **17** (2016), no. 8, 459–469.

[22] D. Field, B. Tiwari, T. Booth, S. Houten, D. Swan, N. Bertrand, and M. Thurston, *Open software for biologists: from famine to feast*, Nature Biotechnology **24** (2006), no. 7, 801.

[23] J. L. Freeman, G. H. Perry, L. Feuk, R. Redon, S. A. McCarroll, D. M. Altshuler, H. Aburatani, K. W. Jones, C. Tyler-Smith, M. E. Hurles, et al., *Copy number variation: new insights in genome diversity*, Genome Research **16** (2006), no. 8, 949–961.

[24] E. Gonzalez, H. Kulkarni, H. Bolivar, A. Mangano, R. Sanchez, G. Catano, R. J. Nibbs, B. I. Freedman, M. P. Quinones, M. J. Bamshad, et al., *The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility*, Science **307** (2005), no. 5714, 1434–1440.

[25] A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, R. C. Lewontin, and W. M. Gelbart, *An introduction to genetic analysis, 7th edition*, New York: W. H. Freeman, 2000.

[26] M. Hattori, A. Fujiyama, T. D. Taylor, H. Watanabe, T. Yada, H. S. Park, A. Toyoda, K. Ishii, Y. Totoki, D. K. Choi, et al., *The DNA sequence of human chromosome 21*, Nature **405** (2000), 311–319.

[27] W. Huang, L. Li, J. R. Myers, and G. T. Marth, *ART: a next-generation sequencing read simulator*, Bioinformatics **28** (2012), no. 4, 593–594.

[28] P. J. Hurd and C. J. Nelson, *Advantages of next-generation sequencing versus the microarray in epigenetic research*, Briefings in Functional Genomics **8** (2009), no. 3, 174.

[29] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee, *Detection of large-scale variation in the human genome*, Nature Genetics **36** (2004), no. 9, 949–951.

[30] S. Ivakhno, T. Royce, A. J. Cox, D. J. Evers, R. K. Cheetham, and S. Tavare, *CNAseg–a novel framework for identification of copy number changes in cancer from second-generation sequencing data*, Bioinformatics **26** (2010), 3051–3058.

[31] T. Ji and J. Chen, *Modeling the next generation sequencing read count data for DNA copy number variant study*, Statistical Applications in Genetics and Molecular Biology **14** (2015), no. 4, 361–374.

[32] B. Jia, L. Xuan, K. Cai, Z. Hu, L. Ma, and C. Wei, *NeSSM: a next-generation sequencing simulator for metagenomics*, PLoS ONE **8** (2013), no. 10, e75448.

[33] M. E. Jones, *Albrecht kossel and a biographical sketch*, The Yale Journal of Biology and Medicine **26** (1953), no. 1, 80–97.

[34] O. P. Kallioniemi, A. Kallioniemi, D. Sudar, D. Rutovitz, J. W. Gray, F. Waldman, and D. Pinkel, *Comparative genomic hybridization: a rapid new method for detecting and mapping DNA amplification in tumors*, Semin. Cancer Bio **4** (1993), 41–46.

[35] P. R. Langer-Safer, M. Levine, and D. C. Ward, *Immunological method for mapping genes on Drosophila polytene chromosomes*, Proceedings of the National Academy of Sciences **79** (1982), no. 4, 4381–4385.

[36] B. Levin and J. Reeds, *Compound multinomial likelihood functions are unimodal: Proof of a conjecture of I. T. Good*, The Annals of Statistics **5** (1977), no. 1, 79–87.

[37] H. Li, *Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM*, arXiv preprint arXiv:1303.3997 (2013).

[38] H. Li and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler transform*, Bioinformatics **25** (2009), no. 14, 1754–1760.

[39] ———, *Fast and accurate long- read alignment with Burrows-Wheeler transform*, Bioinformatics **26** (2010), no. 5, 589–595.

[40] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, et al., *The sequence alignment/map format and SAMtools*, Bioinformatics **25** (2009), no. 16, 2078–2079.

[41] H. Li, J. Vallandingham, and J. Chen, *SeqBBS: A change-point model based algorithm and R package for searching CNV regions via the ratio of sequencing reads*, (2013), 40–43.

[42] X. Li, S. Chen, and W. Xie, *Sensitive and reliable population-scale copy number*

variation detection method based on low coverage sequencing, PLoS ONE **9** (2014), no. 1, e85096.

[43] I. Lobo, *Copy number variation and genetic disease*, Nature Education **1** (2008), no. 1, 65.

[44] F. Lysholm, B. Andersson, and B. Persson, *An efficient simulator of 454 data using configurable statistical models*, BMC Research Notes **4** (2011), no. 1, 449.

[45] C. R. Machado and C. F. M. Menck, *Human DNA repair diseases: From genome instability to cancer*, Brazilian Journal of Genetics **20** (1997), no. 4.

[46] A. Magi, L. Tattini, T. Pippucci, F. Torricelli, and M. Benelli, *Read count approach for DNA copy number variants detection*, Bioinformatics **28** (2012), no. 4, 470–478.

[47] J. H. Mathews, *Numerical methods for mathematics, science, and engineering*, second ed., Prentice Hall, Englewood Cliffs, NJ, 1992.

[48] B. H. Mayall, A. V. Carrano, D. H. II Moore, L. K. Ashworth, D. E. Bennett, and M. L. Mendelsohn, *The DNA-based human karyotype*, Cytometry **5** (1984), 376–385.

[49] M. L. Metzker, *Sequencing technologies- the next generation*, Nature Review Genetics **11** (2010), 31–46.

[50] C. A. Miller, O. Hampton, C. Coarfa, and A. Milosavljevic, *ReadDepth: a parallel R package for detecting copy number alterations from short sequencing reads*, PLoS ONE **6** (2011), no. 1, e16327.

[51] R. Mitra, R. Gill, S. Datta, and S. Datta, *Statistical analyses of next generation sequencing data: An overview*, (2014), 1–24.

[52] A. B. Olshen and E. S. Venkatraman, *Circular binary segmentation for the analysis of array-based DNA copy number data*, Biostatistics **5** (2004), 557–572.

[53] G. Peiró, D. Mayr, P. Hillemanns, U. Löhrs, and J. Diebold, *Analysis of HER-2/neu amplification in endometrial carcinoma by chromogenic in situ hybridization. correlation with fluorescence in situ hybridization, HER-2/neu, p53 and Ki-67 protein expression, and outcome*, Modern Pathology **17** (2004), no. 3, 277–287.

[54] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. L. Kuo, C. Chen, Y. Zhai, et al., *High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays.*, Nature Genetics **20** (1998), no. 2.

[55] D. Pinto, A. T. Pagnamenta, L. Klei, R. Anney, D. Merico, R. Regan, J. Conroy, T. R. Magalhaes, C. Correia, B. S. Abrahams, et al., *Functional impact of global rare copy number variation in autism spectrum disorders*, Nature **466** (2010), no. 7304, 368–372.

[56] J. R. Pollack, C. M. Perou, A. A. Alizadeh, M. B. Eisen, A. Pergamenschikov, C. F. Williams, S. S. Jeffrey, D. Botstein, and P. O. Brown, *Genome-wide analysis of DNA copy-number changes using cDNA microarrays*, Nature Genetics **23** (1999), no. 1, 41–46.

[57] L. Pray, *Discovery of DNA structure and function: Watson and Crick*, Nature Education **1** (2008), no. 1, 565–581.

[58] R. Redon, S. Ishikawa, K. R Fitch, L. Feuk, G. H. Perry, T. D Andrews, H. Fiegler, M. H. Shapero, A. R. Carson, W. Chen, et al., *Global variation in copy number in the human genome*, Nature **444** (2006), no. 7118, 444–454.

[59] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson, *Metasim – a sequencing simulator for genomics and metagenomics*, PloS ONE **3** (2008), no. 10, e3373.

[60] S. C. Schuster, *Next-generation sequencing transforms today's biology.*, Nature Methods **5** (2008), 16–18.

[61] J. Sebat, B. Lakshmi, J. Troge, J. Alexander, J. Young, P. Lundin, S. Månér, H. Massa, M. Walker, M. Chi, et al., *Large-scale copy number polymorphism in the human genome*, Science **305** (2004), no. 5683, 525–528.

[62] A. Sen and M. S. Srivastava, *On tests for detecting change in mean*, The Annals of Statistics **3** (1975), no. 1, 98–108.

[63] V. E. Seshan and A. B. Olshen, *DNAcopy: A package for analyzing DNA copy data*, (2017).

[64] J. Shendure and H. Ji, *Next-generation DNA sequencing*, Nat Biotechnol. **26** (2008), no. 10, 1135–1145.

[65] A. Shlien and D. Malkin, *Copy number variations and cancer*, Genome Medicine **1** (2009), no. 6, 62.

[66] D. Siegmund, *Boundary crossing probabilities and statistical applications*, The Annals of Statistics (1986), 361–404.

[67] W. Simonsen, *On the solution of a maximum-likelihood equation of the negative binomial distribution*, Scand. Actuarial J. **1976** (1976), no. 4, 220–231.

[68] ———, *Correction to "On the solution of a maximum-likelihood equation of the negative binomial distribution"*, Scand. Actuarial J. **1980** (1980), no. 4, 227–228.

[69] A. M. Snijders, N. Nowak, R. Segraves, S. Blackwood, N. Brown, J. Conroy, G. Hamilton, A. K. Hindle, B. Huey, and K. Kimura, *Assembly of microarrays for genome-wide measurement of DNA copy number*, Nat Genet. **29** (2001), 263–264.

[70] D. St. Clair, *Copy number variation and schizophrenia*, Schizophrenia Bulletin **35** (2009), no. 1, 9–12.

[71] Z. Su, Z. Li, T. Chen, Q. Z. Li, H. Fang, D. Ding, W. Ge, B. Ning, H. Hong, R. G. Perkins, et al., *Comparing next-generation sequencing and microarray technologies in a toxicological study of the effects of aristolochic acid on rat kidneys*, Chemical Research in Toxicology **24** (2011), no. 9, 1486–1493.

[72] E. S. Venkatraman and A. B. Olshen, *A faster circular binary segmentation algorithm for the analysis of array CGH data*, Bioinformatics **23** (2007), no. 6, 657–663.

[73] H. Wang, D. Nettleton, and K. Ying, *Copy number variation detection using next generation sequencing read counts*, BMC Bioinformatics **15** (2014), 109.

[74] Y. Wang, *Estimation problems for the two-parameter negative binomial distribution*, Statistics & Probability Letters **26** (1996), 113–114.

[75] J. D. Watson and F. H. C. Crick, *Molecular structure of nucleic acids:a structure for deoxyribose nucleic acid*, Nature **171** (1953), 737–738.

[76] R. Xi, A. G. Hadjipanayis, L. J. Luquette, T. M. Kim, E. Lee, J. Zhang, M. D. Johnson, D. M. Muzny, D. A. Wheeler, R. A. Gibbs, et al., *Copy number variation detection in whole-genome sequencing data using the Bayesian information criterion*, Proceedings of the National Academy of Science **108** (2011), no. 46, e1128–e1136.

[77] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat, *Sensitive and accurate detection of copy number variants using read depth of coverage*, Genome Research **19** (2009), no. 9, 1586–1592.

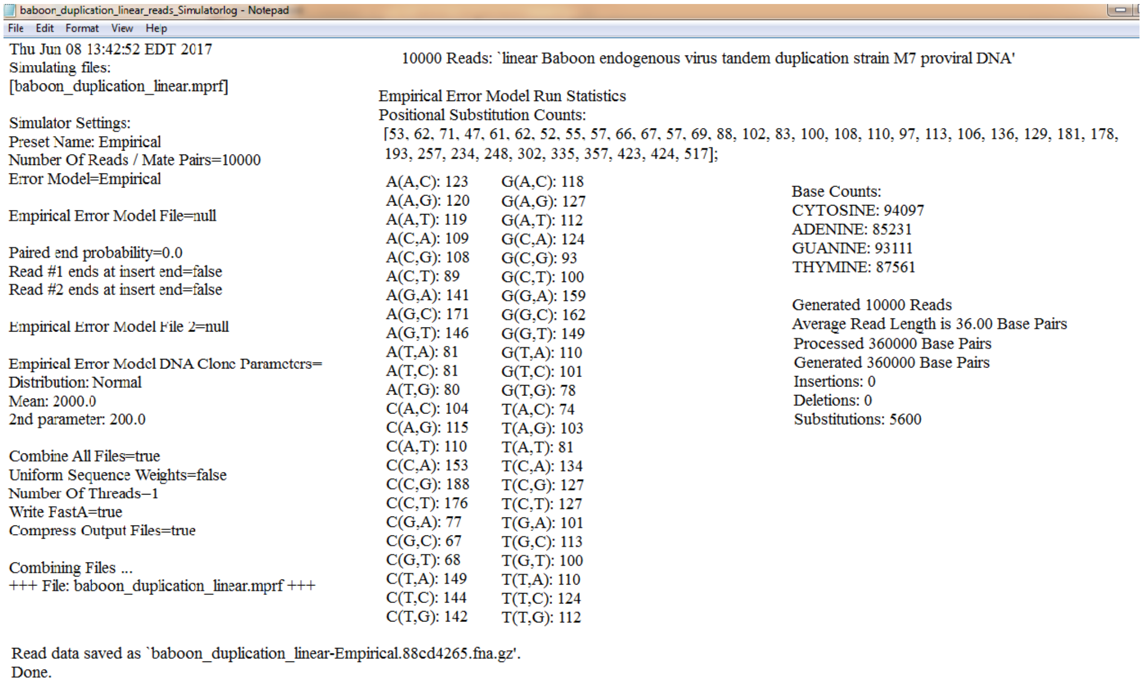
[78] M. Zhao, Q. Wang, Q. Wang, P. Jia, and Z. Zhao, *Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives*, BMC Bioinformatics **14** (2013), no. 11, 1.

APPENDIX

A.1 MetaSim Output for Illumina Short Reads

A.1.1 Simulator Log (.mprf file)

MetaSim generated 10,000 Illumina reads, each is 36 base long, with 5600 substitutions. Figure A.1 is the output of .mprf file, a simulator log, which is shown below.


baboon_duplication_linear_reads_Simulatorlog - Notepad
File Edit Format View Help

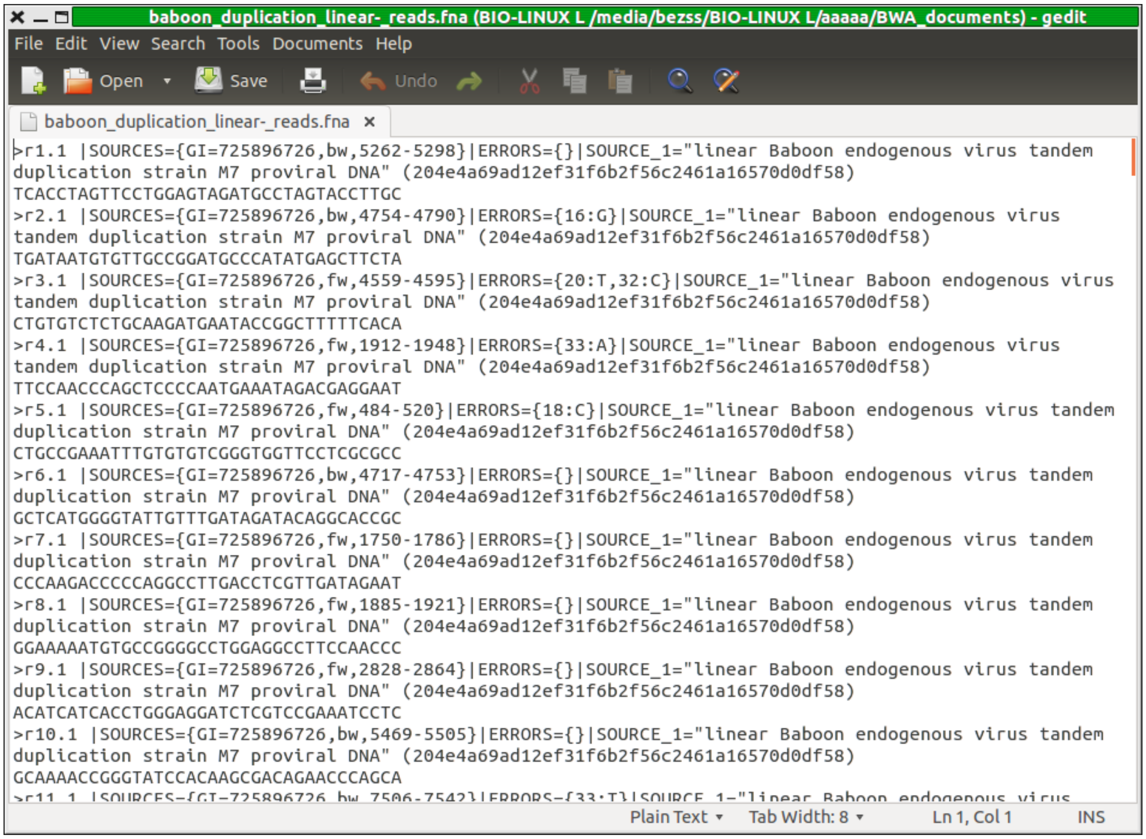
```
Thu Jun 08 13:42:52 EDT 2017
Simulating files:
[baboon_duplication_linear.mprf]

Simulator Settings:
Preset Name: Empirical
Number Of Reads / Mate Pairs=10000
Error Model=Empirical

Empirical Error Model File=null

Paired end probability=0.0
Read #1 ends at insert end=false
Read #2 ends at insert end=false

Empirical Error Model File 2=null

Empirical Error Model DNA Clone Parameters=
Distribution: Normal
Mean: 2000.0
2nd parameter: 200.0

Combine All Files=true
Uniform Sequence Weights=false
Number Of Threads–1
Write FastA=true
Compress Output Files=true

Combining Files ...
+++ File: baboon_duplication_linear.mprf +++

   10000 Reads: `linear Baboon endogenous virus tandem duplication strain M7 proviral DNA'

Empirical Error Model Run Statistics
Positional Substitution Counts:
 [53, 62, 71, 47, 61, 62, 52, 55, 57, 66, 67, 57, 69, 88, 102, 83, 100, 108, 110, 97, 113, 106, 136, 129, 181, 178,
 193, 257, 234, 248, 302, 335, 357, 423, 424, 517];

 A(A,C): 123    G(A,C): 118                          Base Counts:
 A(A,G): 120    G(A,G): 127                          CYTOSINE: 94097
 A(A,T): 119    G(A,T): 112                          ADENINE: 85231
 A(C,A): 109    G(C,A): 124                          GUANINE: 93111
 A(C,G): 108    G(C,G): 93                           THYMINE: 87561
 A(C,T): 89     G(C,T): 100
 A(G,A): 141    G(G,A): 159                          Generated 10000 Reads
 A(G,C): 171    G(G,C): 162                          Average Read Length is 36.00 Base Pairs
 A(G,T): 146    G(G,T): 149                           Processed 360000 Base Pairs
 A(T,A): 81     G(T,A): 110                           Generated 360000 Base Pairs
 A(T,C): 81     G(T,C): 101                           Insertions: 0
 A(T,G): 80     G(T,G): 78                            Deletions: 0
 C(A,C): 104    T(A,C): 74                            Substitutions: 5600
 C(A,G): 115    T(A,G): 103
 C(A,T): 110    T(A,T): 81
 C(C,A): 153    T(C,A): 134
 C(C,G): 188    T(C,G): 127
 C(C,T): 176    T(C,T): 127
 C(G,A): 77     T(G,A): 101
 C(G,C): 67     T(G,C): 113
 C(G,T): 68     T(G,T): 100
 C(T,A): 149    T(T,A): 110
 C(T,C): 144    T(T,C): 124
 C(T,G): 142    T(T,G): 112

Read data saved as `baboon_duplication_linear-Empirical.88cd4265.fna.gz'.
Done.
```

Figure A.1: Reads Simulator Log

Positional Substitution Counts indicate the number of reads that has a substitution for each base. That is, the number of reads in which the nucleotide in the first base is substituted by another nucleotide is 53, the number of reads in which the nucleotide in the second base is substituted by another nucleotide is 62, and so on. So, we have 36 positional substitution counts as shown in the Figure A.1. The entry $A(C,T)$: 89 implies that there are 89 reads that has the substitution error as base T for base C, when the preceding base is A; similarly $A(A,C)$: 123 implies that there are 123 number of reads that has the substitution error as base C for base A, when the preceding base is A; and so on. Likewise, we will have 48 possibilities to create substitution errors, and all are listed in the Figure A.1. Moreover, it is shown in fact that there are total of 94097 Cytosine base counts, 85231 Adenine base counts, 93111 Guanine base counts, and 87561 Thymine base counts in the simulation.

A.1.2 Fasta File

MetaSim output for the Illumina short read simulation is shown in Figure A.2. Information on each of the 10,000 short reads is given in a group of three consecutive lines. The third line gives the 36 base calls for each read, and the first two lines are a comment which includes some important information from MetaSim such as the read number(>ri.1 , where $i = 1, 2, 3, \ldots, 10,000$), the true positions (inclusive interval with 36 bases) of the read from the target genome, the orientation of the read (`fw`, forward, if the bases are read from the 5'-end and `bw`, backward, if the bases are read from the 3'-end), and the bases on the read which are errors (with bases on the read labeled from 0 to 35). For an example, "ERRORS={3:T,28:G}" implies that, there is an error at the 4^{th} base and the error is base T and also, there is an another error at the 29^{th} base, and the error is base G. This commented information on the first two lines corresponding to each read is assumed to be unknown, and only the third line with the base calls for the reads is used in the alignment step.

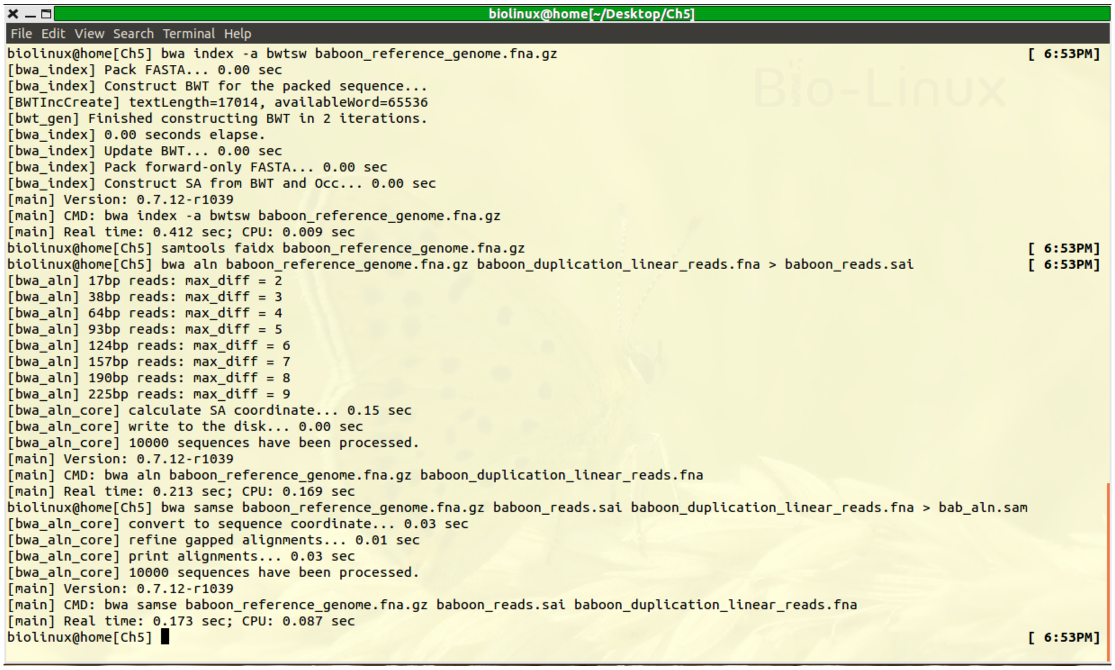

baboon_duplication_linear-_reads.fna (BIO-LINUX L /media/bezss/BIO-LINUX L/aaaaa/BWA_documents) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
baboon_duplication_linear-_reads.fna
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

```
>r1.1 |SOURCES={GI=725896726,bw,5262-5298}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
TCACCTAGTTCCTGGAGTAGATGCCTAGTACCTTGC
>r2.1 |SOURCES={GI=725896726,bw,4754-4790}|ERRORS={16:G}|SOURCE_1="linear Baboon endogenous virus
tandem duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
TGATAATGTGTTGCCGGATGCCCATATGAGCTTCTA
>r3.1 |SOURCES={GI=725896726,fw,4559-4595}|ERRORS={20:T,32:C}|SOURCE_1="linear Baboon endogenous virus
tandem duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
CTGTGTCTCTGCAAGATGAATACCGGCTTTTTCACA
>r4.1 |SOURCES={GI=725896726,fw,1912-1948}|ERRORS={33:A}|SOURCE_1="linear Baboon endogenous virus
tandem duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
TTCCAACCCAGCTCCCCAATGAAATAGACGAGGAAT
>r5.1 |SOURCES={GI=725896726,fw,484-520}|ERRORS={18:C}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
CTGCCGAAATTTGTGTGTCGGGTGGTTCCTCGCGCC
>r6.1 |SOURCES={GI=725896726,bw,4717-4753}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
GCTCATGGGGTATTGTTTGATAGATACAGGCACCGC
>r7.1 |SOURCES={GI=725896726,fw,1750-1786}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
CCCAAGACCCCCAGGCCTTGACCTCGTTGATAGAAT
>r8.1 |SOURCES={GI=725896726,fw,1885-1921}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
GGAAAAATGTGCCGGGGCCTGGAGGCCTTCCAACCC
>r9.1 |SOURCES={GI=725896726,fw,2828-2864}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
ACATCATCACCTGGGAGGATCTCGTCCGAAATCCTC
>r10.1 |SOURCES={GI=725896726,bw,5469-5505}|ERRORS={}|SOURCE_1="linear Baboon endogenous virus tandem
duplication strain M7 proviral DNA" (204e4a69ad12ef31f6b2f56c2461a16570d0df58)
GCAAAACCGGGTATCCACAAGCGACAGAACCCAGCA
>r11.1 |SOURCES={GI=725896726,bw,7506-7542}|ERRORS={33:T}|SOURCE_1="linear Baboon endogenous virus
```

Figure A.2: Screenshot of fasta file with short reads.

A.2 BWA output

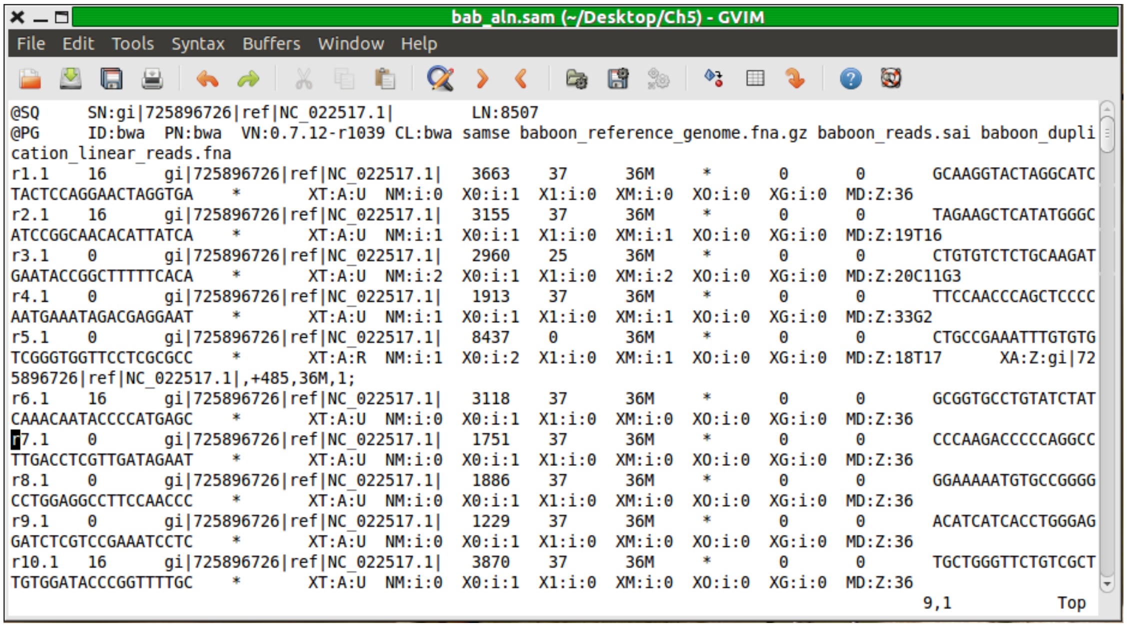
Figure A.3 shows a screenshot of the commands using BWA and samtools software available in the terminal in Bio-linux.


biolinux@home[~/Desktop/Ch5]
File Edit View Search Terminal Help
Bio-Linux

```
biolinux@home[Ch5] bwa index -a bwtsw baboon_reference_genome.fna.gz                                                          [ 6:53PM]
[bwa_index] Pack FASTA... 0.00 sec
[bwa_index] Construct BWT for the packed sequence...
[BWTIncCreate] textLength=17014, availableWord=65536
[bwt_gen] Finished constructing BWT in 2 iterations.
[bwa_index] 0.00 seconds elapse.
[bwa_index] Update BWT... 0.00 sec
[bwa_index] Pack forward-only FASTA... 0.00 sec
[bwa_index] Construct SA from BWT and Occ... 0.00 sec
[main] Version: 0.7.12-r1039
[main] CMD: bwa index -a bwtsw baboon_reference_genome.fna.gz
[main] Real time: 0.412 sec; CPU: 0.009 sec
biolinux@home[Ch5] samtools faidx baboon_reference_genome.fna.gz                                                              [ 6:53PM]
biolinux@home[Ch5] bwa aln baboon_reference_genome.fna.gz baboon_duplication_linear_reads.fna > baboon_reads.sai              [ 6:53PM]
[bwa_aln] 17bp reads: max_diff = 2
[bwa_aln] 38bp reads: max_diff = 3
[bwa_aln] 64bp reads: max_diff = 4
[bwa_aln] 93bp reads: max_diff = 5
[bwa_aln] 124bp reads: max_diff = 6
[bwa_aln] 157bp reads: max_diff = 7
[bwa_aln] 190bp reads: max_diff = 8
[bwa_aln] 225bp reads: max_diff = 9
[bwa_aln_core] calculate SA coordinate... 0.15 sec
[bwa_aln_core] write to the disk... 0.00 sec
[bwa_aln_core] 10000 sequences have been processed.
[main] Version: 0.7.12-r1039
[main] CMD: bwa aln baboon_reference_genome.fna.gz baboon_duplication_linear_reads.fna
[main] Real time: 0.213 sec; CPU: 0.169 sec
biolinux@home[Ch5] bwa samse baboon_reference_genome.fna.gz baboon_reads.sai baboon_duplication_linear_reads.fna > bab_aln.sam
[bwa_aln_core] convert to sequence coordinate... 0.03 sec
[bwa_aln_core] refine gapped alignments... 0.01 sec
[bwa_aln_core] print alignments... 0.03 sec
[bwa_aln_core] 10000 sequences have been processed.
[main] Version: 0.7.12-r1039
[main] CMD: bwa samse baboon_reference_genome.fna.gz baboon_reads.sai baboon_duplication_linear_reads.fna
[main] Real time: 0.173 sec; CPU: 0.087 sec
biolinux@home[Ch5]                                                                                                            [ 6:53PM]
```

Figure A.3: Screenshot of BWA and samtools commands in Bio-linux terminal.

These commands create a sam file `bab_aln.sam` with the alignments of the short reads in the file `baboon_duplication_linear_reads.fna` using the reference genome `baboon_reference_genome.fna.gz`. Figure A.4 shows a screenshot of part of the contents of the sam file.



```
@SQ     SN:gi|725896726|ref|NC_022517.1|        LN:8507
@PG     ID:bwa  PN:bwa  VN:0.7.12-r1039 CL:bwa samse baboon_reference_genome.fna.gz baboon_reads.sai baboon_dupli
cation_linear_reads.fna
r1.1    16      gi|725896726|ref|NC_022517.1|   3663    37      36M     *       0       0       GCAAGGTACTAGGCATC
TACTCCAGGAACTAGGTGA     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r2.1    16      gi|725896726|ref|NC_022517.1|   3155    37      36M     *       0       0       TAGAAGCTCATATGGGC
ATCCGGCAACACATTATCA     *       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:19T16
r3.1    0       gi|725896726|ref|NC_022517.1|   2960    25      36M     *       0       0       CTGTGTCTCTGCAAGAT
GAATACCGGCTTTTTCACA     *       XT:A:U  NM:i:2  X0:i:1  X1:i:0  XM:i:2  XO:i:0  XG:i:0  MD:Z:20C11G3
r4.1    0       gi|725896726|ref|NC_022517.1|   1913    37      36M     *       0       0       TTCCAACCCAGCTCCCC
AATGAAATAGACGAGGAAT     *       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:33G2
r5.1    0       gi|725896726|ref|NC_022517.1|   8437    0       36M     *       0       0       CTGCCGAAATTTGTGTG
TCGGGTGGTTCCTCGCGCC     *       XT:A:R  NM:i:1  X0:i:2  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:18T17      XA:Z:gi|72
5896726|ref|NC_022517.1|,+485,36M,1;
r6.1    16      gi|725896726|ref|NC_022517.1|   3118    37      36M     *       0       0       GCGGTGCCTGTATCTAT
CAAACAATACCCCATGAGC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r7.1    0       gi|725896726|ref|NC_022517.1|   1751    37      36M     *       0       0       CCCAAGACCCCCAGGCC
TTGACCTCGTTGATAGAAT     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r8.1    0       gi|725896726|ref|NC_022517.1|   1886    37      36M     *       0       0       GGAAAAATGTGCCGGGG
CCTGGAGGCCTTCCAACCC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r9.1    0       gi|725896726|ref|NC_022517.1|   1229    37      36M     *       0       0       ACATCATCACCTGGGAG
GATCTCGTCCGAAATCCTC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
r10.1   16      gi|725896726|ref|NC_022517.1|   3870    37      36M     *       0       0       TGCTGGGTTCTGTCGCT
TGTGGATACCCGGTTTTGC     *       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
                                                                                                9,1           Top
```

Figure A.4: Screenshot of sam file with aligned reads.

The first three lines in the sam file indicates the information about the files that BWA used for the aligning (reference genome file and the illumina short reads file). After that, the information on each of the aligned reads is given as a group of two consecutive lines. The three different values 16, 0, and 4 in the first line of the two consecutive lines gives the mapping information of the read. A "0" means that the read matched on the forward strand `fw`, a "4" means that the read did not match, and a "16" means it matched on the reverse strand `bw`. Thus, the values 37, 25, and 0 in the first line of the two consecutive lines gives the mapping quality of each read aligning. A read alignment with a mapping quality 30 or above (in ours it is 37) usually implies:

- The overall base quality of the read is good.
- The read has few or just one "good" hit on the reference. That is current alignment is the best.

Then, a read alignment with mapping quality 25 implies that there are two mismatches in the read, and still it aligns to the best position on the reference. Finally, a mapping quality 0 implies that there a several good hits on the reference, which means that a read can be aligned equally well to multiple positions, so that BWA will randomly pick one position and assign it a mapping quality 0. The digits prior to the mapping quality indicates the true position of each read in the reference genome. BWA generates the following optional fields. Tags starting with X are specific to BWA. (See `http://bio-bwa.sourceforge.net/bwa.shtml` for complete documentation of the fields.)

XT Type: Unique/Repeat/N/Mate-sw

NM Edit distance

X0 Number of best hits

X1 Number of suboptimal hits found by BWA

XM Number of mismatches in the alignment

XO Number of gap opens

XG Number of gap extensions

MD Mismatching positions/bases

Here MD:Z: 36 indicates all bases are mapped to the reference genome. MD:Z:2G16C16 implies that there is an error base (mismatching base) at the 3rd base and the correct base should be G, and also an another error at 20th base and the correct base should be C.

A.3 R Code

This section gives the custom R functions written for implementing the methods in this dissertation. First, we give functions implementing code for the Poisson model.

The function `mle.Pois` finds the maximum likelihood estimator for μ given a random sample of independent Poisson(μ) random variables. This function outputs `mu` (the MLE of μ) and `lik` (the maximum of the likelihood function). The input to this function is `x` (a vector with the observed count data from the random sample).

```
mle.Pois=function(x){
 n=length(x)
 x.bar=mean(x)
 if (x.bar>0)
  lik=n*x.bar*(log(x.bar)-1)
 else
  lik=0
 list(mu=x.bar,lik=lik)
}
```

The function `est.ij.Pois` finds the maximum likelihood estimates based on the model with the likelihood function in equation (4.13). The function outputs `mu.out` (the MLE of μ), `mu.in` (the MLE of $\mu + \delta$), `lik` (the maximum value of the likelihood function), `alpha` (the MLE of α), and `beta` (the MLE of β). The inputs to the function are `x` (a vector with the observed count data from the random sample), `find.mle` (the function used to find the MLE given a random sample of independent Poisson random variables), `ijSet` (set of possible locations for α and β, where the default 0 indicates that all integers from 1 to n are possible), and `print.out` (indicating whether to print out messages with progress as the function executes).

```
est.ij.Pois=function(x,find.mle=mle.Pois,ijSet=0,
print.out=FALSE){
 n=length(x)
 temp=find.mle(x)
 res=list(mu.out=temp$mu,mu.in=temp$mu,lik=temp$lik,
alpha=1,beta=n+1)
 if (n>1){
  if (ijSet[1]==0){
   for (i in 1:(n-1))
    for (j in i:(n-1)){
     w=i:j
     x.in=x[w]
     x.out=x[-w]
     temp.in=find.mle(x.in)
     temp.out=find.mle(x.out)
     temp.lik=temp.in$lik+temp.out$lik
     if (temp.lik>res$lik)
      res=list(mu.out=temp.out$mu,mu.in=temp.in$mu,lik=temp.lik,
alpha=i,beta=j+1)
     if (print.out==TRUE)
      cat("i=",i," j=",j," lik=",temp.lik,"\n")
    }
  }
  else{
   n.ij=length(ijSet)
   for (i in 1:n.ij)
    for (j in i:n.ij){
     if (print.out==TRUE)
      cat("i=",ijSet[i]," j=",ijSet[j],"\n")
     w=ijSet[i]:ijSet[j]
     if (length(w)<n){
      x.in=x[w]
      x.out=x[-w]
      temp.in=find.mle(x.in)
      temp.out=find.mle(x.out)
      temp.lik=temp.in$lik+temp.out$lik
      if (temp.lik>res$lik)
       res=list(mu.out=temp.out$mu,mu.in=temp.in$mu,lik=temp.lik,
alpha=ijSet[i],beta=ijSet[j]+1)
      if (print.out==TRUE)
       cat("lik=",temp.lik,"\n")
     }
    }
  }
  if (print.out==TRUE){
   cat("Inside interval:"," [",res$alpha,",",res$beta-1,"]\n")
  }
 }
 res
}
```

The function `test.split.Pois` performs the likelihood ratio test with the parametric bootstrap described in section 4.2.1 using the Poisson model. The function outputs `reject` (the boolean value decision for whether or not to reject the null hypothesis), `p.val` (the estimated p-value for the test), and `model` (the fitted model output from the function `est.ij.Pois` using the alternative model if the test is rejected or the null model if not). The inputs to the function are `x` (a vector with the observed count data from the random sample), `find.mle` (the function used to find the MLE given a random sample of independent Poisson random variables), `ijSet` (set of possible locations for α and β, where the default 0 indicates that all integers from 1 to n are possible), `alpha` (the nominal size of the test), `B` (the number of bootstrap samples used), and `print.out` (indicating whether to print out messages with progress as the function executes). Note that for computational efficiency, the function does not necessarily use `B` bootstrap samples, but instead proceeds sequentially and stops once the decision for the test based on the specified `alpha` level has been determined.

```
test.split.Pois=function(x,find.mle=mle.Pois,ijSet=0,alpha=.05,
B=1000,print.out=FALSE){
 n=length(x)
 B.accept=ceiling(B*alpha)
 B.reject=ceiling(B*(1-alpha))
 obs.larger=0
 obs.notlarger=0
 null.model=find.mle(x)
 mu.hat=null.model$mu
 obs.lik.null=null.model$lik
 alt.model=est.ij.Pois(x,find.mle,ijSet=ijSet)
 obs.lik.alt=alt.model$lik
 obs.logLambda=obs.lik.alt-obs.lik.null
 if (print.out==TRUE)
  cat("obs.logLambda=",obs.logLambda,"\n")
 while ((obs.notlarger<B.accept)&(obs.larger<B.reject)){
  x.boot=rpois(n,lambda=mean(x))
  boot.lik.null=find.mle(x.boot)$lik
  boot.lik.alt=est.ij.Pois(x.boot,find.mle,ijSet=ijSet)$lik
  boot.logLambda=boot.lik.alt-boot.lik.null
  if (boot.logLambda<obs.logLambda)
   obs.larger=obs.larger+1
  else
   obs.notlarger=obs.notlarger+1
  if (print.out==TRUE)
   cat("boot.logLambda=",boot.logLambda,"Obs Larger= ",
obs.larger,"/",obs.larger+obs.notlarger,"\n")
 }
 if (obs.larger==B.reject){
  if (print.out==TRUE)
   cat("Reject H0\n")
  results=list(reject=TRUE,p.val=obs.notlarger/(obs.larger+
obs.notlarger),model=alt.model)
 }
 if (obs.notlarger==B.accept){
  if (print.out==TRUE)
   cat("Fail to reject H0\n")
  results=list(reject=FALSE,p.val=obs.notlarger/(obs.larger+
obs.notlarger),model=null.model)
 }
 results
}
```

The function `cbs.Pois` performs the *Poisson CBS* method described in Section 4.3. The function outputs a data frame with rows containing the segments determined by the method and columns `ID` (a generic label “ID” for each segment), `chrom` (with generic value 1 for each segment), `loc.start` (the smallest locus in the segment), `loc.end` (the largest locus in the segment), `num.mark` (the number of loci in the segment), and `seg.mean` (the estimated value of μ in the segment). The inputs to the function are `x` (a vector with the observed count data from the random sample), `ijSet` (set of possible locations for α and β, where the default 0 indicates that all integers from 1 to n are possible), `print.out` (indicating whether to print out messages with progress as the function executes), and `ep` (a tolerance threshold used internally to cluster segment estimates).

```
cbs.Pois=function(x,ijSet=0,print.out=FALSE,ep=1e-4){
 n=length(x)
 null.model=mle.Pois(x)
 estimated.segments=cbind(rep(null.model$mu,n))
 unique.segments=unique(estimated.segments)
 n.segments=nrow(unique.segments)
 n.segments.last=0
 while (n.segments>n.segments.last){
  n.segments.last=n.segments
  for (i in 1:n.segments){
   w=which(abs(estimated.segments[,1]-unique.segments[i,1])<ep)
   xw=x[w]
   if (print.out==TRUE)
    cat("Attempting split on indices:",w,"\n")
   try.split=test.split.Pois(xw,ijSet=ijSet,print.out=print.out)
   if (try.split$reject==TRUE){
    new.estimates=cbind(
c(rep(try.split$model$mu.out,try.split$model$alpha-1),
rep(try.split$model$mu.in,
try.split$model$beta-try.split$model$alpha),
rep(try.split$model$mu.out,length(w)+1-try.split$model$beta)))
    estimated.segments[w,]=new.estimates
    if (print.out==TRUE){
     cat("Split is statistically significant\n")
     cat("New groups:\n")
     if (try.split$model$alpha>1)
      cat("Outer segment: ",w[1:(try.split$model$alpha-1)],"\n")
     if (try.split$model$beta<=length(w))
      cat("Outer segment: ",w[try.split$model$beta:length(w)],"\n")
     cat("Inner segment: ",
w[try.split$model$alpha:(try.split$model$beta-1)],"\n")
    }
   }
  }
  unique.segments=unique(estimated.segments)
  n.segments=nrow(unique.segments)
 }
 start=1
 end=NULL
 mark=NULL
 mu=estimated.segments[1,1]
 for (i in 2:n)
  if (abs(estimated.segments[i,1]-estimated.segments[i-1,1])>ep){
   end=c(end,i-1)
   mark=c(mark,max(end)-max(start)+1)
   start=c(start,i)
   mu=c(mu,estimated.segments[i,1])
  }
 end=c(end,n)
 mark=c(mark,n-max(start)+1)
 data.frame(ID="ID",chrom=1,loc.start=start,loc.end=end,
num.mark=mark,seg.mean=mu)
}
```

Next, we give functions implementing code for the negative binomial model.

The function `getN` computes $N_1, \ldots, N_k$ based on observed count data $x_1, \ldots, x_n$ using the formula in Chapter 3. This function outputs `N` (a vector with the observed N's). The input to this function is `x` (a vector with the observed count data from the random sample).

```
getN=function(x){
 k=max(x)
 N=rep(0,k)
 for (i in 1:k){
  N[i]=sum(x>=i)
 }
 return(N)
}
```

The function `mle.nbinom.N` finds the maximum likelihood estimator for r and p given a random sample of independent negative binomial(r, p) random variables. This function outputs `p` (the MLE for p where -1 indicates that it does not exist), `r` (the MLE for r where a negative value indicates that it does not exist), `mu` (the MLE of the mean), `steps` (the number of Newton-Raphson iterations needed for convergence), `score` (the value of $f(r)$ after the last Newton-Raphson iteration), and `lik` (the supremum of the likelihood function). The input to this function is `x` (a vector with the observed count data from the random sample), `ep` (a threshold variable to determine when to stop Newton-Raphson iterations), `count.max` (a threshold of the number of steps allowed for the Newton-Raphson iterations), `print.out` (indicating whether to print out messages with progress as the function executes), and `start` (the starting value of r for the Newton-Raphson method where a default of -999 indicates to use the value suggested by Theorem 3.7).

```
mle.nbinom.N=function(x,ep=.000001,count.max=10000,
print.out=FALSE,start=-999){
 n=length(x)
 Ns=getN(x)
 x.bar=mean(x)
 if (Ns[1]==0){
  p.hat=-1;r.hat=-x.bar/2;count=0;f=-999
  lik=0
 }
 else{
  k=length(Ns)
  sigma2.hat=sum((x-x.bar)^2)/n
  count=0
  if (x.bar>=sigma2.hat){
   if (print.out==TRUE)
    cat("There is no maximizer of the likelihood function\n")
   p.hat=-1;r.hat=-x.bar/2;count=0;f=-999
   lik=sum(x*log(x.bar))-n*x.bar-sum(lfactorial(x))
  }
  else{
   if (start==-999)
    r.temp=Ns[1]^2/(2*length(x)*sum(Ns[-1]))
   else
    r.temp=start
   f=sum(Ns/(r.temp+(1:k)-1))/n-log(1+x.bar/r.temp)
   df=-sum(Ns/(r.temp+(1:k)-1)^2)/n+x.bar/(r.temp*(r.temp+x.bar))
   if (print.out==TRUE)
    cat("count= ",count," r= ",r.temp," f= ",f," df= ",df,"\n")
   while ((abs(f)>ep)&(count<=count.max)){
    r.temp=r.temp-f/df
    f=sum(Ns/(r.temp+(1:k)-1))/n-log(1+x.bar/r.temp)
    df=-sum(Ns/(r.temp+(1:k)-1^2)/n+x.bar/(r.temp*(r.temp+x.bar))
    count=count+1
    if (print.out==TRUE)
     cat("count= ",count," r= ",r.temp," f= ",f," df= ",df,"\n")
   }
   r.hat=r.temp
   p.hat=sum(x)/(n*r.hat+sum(x))
   lgr=lgamma(r.hat)
   lik=log(p.hat)*sum(x)+n*r.hat*log(1-p.hat)+
sum(lgamma(x+r.hat)-lgr)-sum(lfactorial(x))
  }
 }
 list(p=p.hat,r=r.hat,mu=x.bar,steps=count,score=f,lik=lik)
}
```

The function `est.ij.nbinom` finds the maximum likelihood estimates based on the model with the likelihood function in equation (4.6). The function outputs `p.out` (the MLE of p_0 where -1 indicates that it does not exist), `r.out` (the MLE of r_0 where a negative value indicates that it does not exist), `mu.out` (the MLE of the mean in the outside segment), `p.in` (the MLE of p_1 where -1 indicates that it does not exist), `r.in` (the MLE of r_1 where a negative value indicates that it does not exist), `mu.in` (the MLE of the mean in the inside segment), `lik` (the maximum value of the likelihood function), `alpha` (the MLE of α), and `beta` (the MLE of β). The inputs to the function are `x` (a vector with the observed count data from the random sample), `find.mle` (the function used to find the MLE given a random sample of independent negative binomial random variables), `ijSet` (set of possible locations for α and β, where the default 0 indicates that all integers from 1 to n are possible), and `print.out` (indicating whether to print out messages with progress as the function executes).

```
est.ij.nbinom=function(x,find.mle=mle.nbinom.N,ijSet=0,
print.out=FALSE,...){
 n=length(x)
 temp=find.mle(x,...)
 res=list(p.out=temp$p,r.out=temp$r,mu.out=temp$mu,p.in=temp$p,
r.in=temp$r,mu.in=temp$mu,
lik=temp$lik,alpha=1,beta=n+1)
 if (n>3){
  if (ijSet[1]==0){
   for (i in 1:(n-2))
    for (j in (i+1):(n-1))
     if ((i>1)|(j<n-1)){
      w=i:j
      x.in=x[w]
      x.out=x[-w]
      temp.in=find.mle(x.in,...)
      temp.out=find.mle(x.out,...)
      temp.lik=temp.in$lik+temp.out$lik
      if (temp.lik>res$lik)
       res=list(p.out=temp.out$p,r.out=temp.out$r,mu.out=
temp.out$mu,p.in=temp.in$p,r.in=temp.in$r,mu.in=temp.in$mu,
lik=temp.lik,alpha=i,beta=j+1)
      if (print.out==TRUE)
       cat("i=",i," j=",j," lik=",temp.lik,"\n")
     }
  }
  else{
   n.ij=length(ijSet)
   for (i in 1:(n.ij-1))
    for (j in (i+1):n.ij)
     if ((i>1)|(j<n-1)){
      if (print.out==TRUE)
       cat("i=",ijSet[i]," j=",ijSet[j],"\n")
      w=ijSet[i]:ijSet[j]
      if (length(w)<n){
       x.in=x[w]
       x.out=x[-w]
       temp.in=find.mle(x.in,...)
       temp.out=find.mle(x.out,...)
       temp.lik=temp.in$lik+temp.out$lik
       if (temp.lik>res$lik)
        res=list(p.out=temp.out$p,r.out=temp.out$r,mu.out=
temp.out$mu,p.in=temp.in$p,r.in=temp.in$r,mu.in=temp.in$mu,
lik=temp.lik,alpha=ijSet[i],beta=ijSet[j]+1)
       if (print.out==TRUE)
        cat("i=",i," j=",j," lik=",temp.lik,"\n")
      }
     }
  }
  if (print.out==TRUE){
   cat("Inside interval:"," [",res$alpha,",",res$beta-1,"]\n")
  }
 }
 res
}
```

The function `test.split.nbinom` performs the likelihood ratio test with the parametric bootstrap described in section 4.2.1 using the negative binomial model. The function outputs `reject` (the boolean value decision for whether or not to reject the null hypothesis), `p.val` (the estimated p-value for the test), and `model` (the fitted model output from the function `est.ij.nbinom` using the alternative model if the test is rejected or the null model if not). The inputs to the function are `x` (a vector with the observed count data from the random sample), `find.mle` (the function used to find the MLE given a random sample of independent negative binomial random variables), `ijSet` (set of possible locations for α and β, where the default 0 indicates that all integers from 1 to n are possible), `alpha` (the nominal size of the test), `B` (the number of bootstrap samples used), and `print.out` (indicating whether to print out messages with progress as the function executes). Note that for computational efficiency, the function does not necessarily use `B` bootstrap samples, but instead proceeds sequentially and stops once the decision for the test based on the specified `alpha` level has been determined.

```
test.split.nbinom=function(x,find.mle=mle.nbinom.N,ijSet=0,
alpha=.05,B=1000,print.out=FALSE,...){
 n=length(x)
 B.accept=ceiling(B*alpha)
 B.reject=ceiling(B*(1-alpha))
 obs.larger=0
 obs.notlarger=0
 null.model=find.mle(x,...)
 r.hat=null.model$r
 p.hat=null.model$p
 obs.lik.null=null.model$lik
 alt.model=est.ij.nbinom(x,find.mle,ijSet=ijSet,...)
 obs.lik.alt=alt.model$lik
 obs.logLambda=obs.lik.alt-obs.lik.null
 if (print.out==TRUE)
  cat("obs.logLambda=",obs.logLambda,"\n")
 while ((obs.notlarger<B.accept)&(obs.larger<B.reject)){
  if (p.hat>0)
   x.boot=rnbinom(n,prob=p.hat,size=r.hat)
  else
   x.boot=rpois(n,lambda=mean(x))
  boot.lik.null=find.mle(x.boot,...)$lik
  boot.lik.alt=est.ij.nbinom(x.boot,find.mle,ijSet=ijSet,...)$lik
  boot.logLambda=boot.lik.alt-boot.lik.null
  if (boot.logLambda<obs.logLambda)
   obs.larger=obs.larger+1
  else
   obs.notlarger=obs.notlarger+1
  if (print.out==TRUE)
   cat("boot.logLambda=",boot.logLambda,"Obs Larger= ",
obs.larger,"/",obs.larger+obs.notlarger,"\n")
 }
 if (obs.larger==B.reject){
  if (print.out==TRUE)
   cat("Reject H0\n")
  results=list(reject=TRUE,p.val=obs.notlarger/(obs.larger+
obs.notlarger),model=alt.model)
 }
 if (obs.notlarger==B.accept){
  if (print.out==TRUE)
   cat("Fail to reject H0\n")
  results=list(reject=FALSE,p.val=obs.notlarger/(obs.larger+
obs.notlarger),model=null.model)
 }
 results
}
```

The function `cbs.nbinom` performs the *negative binomial CBS* method described in Section 4.3. The function outputs a data frame with rows containing the segments determined by the method and columns `ID` (a generic label "ID" for each segment), `chrom` (with generic value 1 for each segment), `loc.start` (the smallest locus in the segment), `loc.end` (the largest locus in the segment), `num.mark` (the number of loci in the segment), `seg.mean` (the estimated value of μ in the segment), `seg.p` (the estimated value of p in the segment where -1 indicates that it does not exist), and `seg.r` (the estimated value of r where a negative value indicates that it does not exist). The inputs to the function are `x` (a vector with the observed count data from the random sample), `ijSet` (set of possible locations for α and β, where the default 0 indicates that all integers from 1 to n are possible), `print.out` (indicating whether to print out messages with progress as the function executes), and `ep` (a tolerance threshold used internally to cluster segment estimates).

```
cbs.nbinom=function(x,ijSet=0,print.out=FALSE,ep=1e-4,...){
 n=length(x)
 null.model=mle.nbinom.N(x,...)
 estimated.segments=cbind(rep(null.model$p,n),null.model$r)
 unique.segments=unique(estimated.segments)
 n.segments=nrow(unique.segments)
 n.segments.last=0
 while (n.segments>n.segments.last){
  n.segments.last=n.segments
  for (i in 1:n.segments){
   w=which((abs(estimated.segments[,1]-unique.segments[i,1])<ep)&
(abs(estimated.segments[,2]-unique.segments[i,2])<ep))
   xw=x[w]
   if (print.out==TRUE)
    cat("Attempting split on indices:",w,"\n")
   try.split=test.split.nbinom(xw,ijSet=ijSet,print.out=print.out,...)
   if (try.split$reject==TRUE){
    new.estimates=cbind(
c(rep(try.split$model$p.out,try.split$model$alpha-1),
rep(try.split$model$p.in,try.split$model$beta-try.split$model$alpha),
rep(try.split$model$p.out,length(w)+1-try.split$model$beta)),
c(rep(try.split$model$r.out,try.split$model$alpha-1),
rep(try.split$model$r.in,try.split$model$beta-try.split$model$alpha),
rep(try.split$model$r.out,length(w)+1-try.split$model$beta)))
    estimated.segments[w,]=new.estimates
    if (print.out==TRUE){
     cat("Split is statistically significant\n")
     cat("New groups:\n")
     if (try.split$model$alpha>1)
      cat("Outer segment: ",w[1:(try.split$model$alpha-1)],"\n")
     if (try.split$model$beta<=length(w))
      cat("Outer segment: ",w[try.split$model$beta:length(w)],"\n")
     cat("Inner segment: ",
w[try.split$model$alpha:(try.split$model$beta-1)],"\n")
    }
   }
  }
  unique.segments=unique(estimated.segments)
  n.segments=nrow(unique.segments)
 }
 start=1
 end=NULL
 mark=NULL
 p=estimated.segments[1,1]
 r=estimated.segments[1,2]
 for (i in 2:n)
  if ((abs(estimated.segments[i,1]-estimated.segments[i-1,1])>ep)|
(abs(estimated.segments[i,2]-estimated.segments[i-1,2])>ep)){
   end=c(end,i-1)
   mark=c(mark,max(end)-max(start)+1)
   start=c(start,i)
   p=c(p,estimated.segments[i,1])
   r=c(r,estimated.segments[i,2])
  }
 end=c(end,n)
 mark=c(mark,n-max(start)+1)
 data.frame(ID="ID",chrom=1,loc.start=start,loc.end=end,
num.mark=mark,seg.mean=(1-p)*r/p,seg.p=p,seg.r=r)
}
```

CURRICULUM VITAE

UDIKA BANDARA
Department of Mathematics
University of Louisville, Luisville, KY 40292.
email : uiband01@louisville.edu

EDUCATION

University of Louisville, Louisville, KY.
Ph. D., Applied and Industrial Mathematics (Anticipated) August 2017.
M. A., Applied and Industrial Mathematics May 2014.

University of Sri Jayewardenepura, Sri Lanka.
B. Sc., Physical Science May 2006.

PROFESSIONAL EXPERIENCE

Instructor, University of Louisville, Louisville, KY
MATH 111: Contemporary Mathematics Summer 2013 - Summer 2017.
MATH 105: College Algebra.

Graduate Teaching Assistant, University of Louisville, Louisville, KY
MATH 111: College Algebra August 2011 - May 2017.
MATH 105: Contemporary Mathematics.
MATH 109: Elementary Statistics.
MATH 180: Elements of Calculus.

Teaching Assistant, Department of Electrical Engineering, University of Moratuwa, Sri Lanka. May 2008 - August 2009.

Teaching Assistant, Institute of Technology, University of Moratuwa, Sri Lanka, January 2007 - May 2008.

Teaching Assistant, Department of Physics, University of Sri Jayewardenepura, Sri Lanka. May 2006 - December 2007.

PROFESSIONAL QUALIFICATIONS

SOA/CAS Exam P/1 May 2015.

PRESENTATIONS

The 2017 Kentucky Sectional Annual Meeting at Berea College, Berea, KY:

March 2017.

"Likelihood-Based Methods for Analysing Copy Number Variation using Next Generation Sequencing Data"