PRESENCE/ABSENCE MARKER DISCOVERY IN RAD MARKERS FOR MULTIPLEXED SAMPLES IN THE CONTEXT OF NEXT-GENERATION SEQUENCING

A Thesis

by

AMIR NIKOOIENEJAD

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Byung-Jun Yoon
Co-Chair of Committee,	Alan Dabney
Committee Members,	Edward R. Dougherty
	Jean-Francois Chamberland
Head of Department,	Chanan Singh

August 2013

Major Subject: Electrical Engineering

ABSTRACT

Recent improvements in sequencing technologies has caused various interesting problems to arouse. Having millions of read sequences as the final product of sequencing genome at a lower cost compared to micro array era, has encouraged scientists to enhance previous methods in various areas of bioinformatics. Genotyping and generating genetic maps to study inherited genotypes in order to analyse specific traits in a population is one of the fields of bioinformatics that involves generating different genetic markers and identify polymorphisms in different individuals of a population.

Presence/absence markers are the main focus of this thesis. This is one type of Restriction site Associate DNA (RAD) markers which is present in some samples and absent in others and is the sign of variation in the cut site of a restriction enzyme. However, the counts of markers in an experiment are highly correlated and calling true absence and presence is not a straightforward task which means any marker with zero count is not necessarily absent in the sample under study. This is also the case for non-zero count markers which are not necessarily present. A good model that can fit the data is able to make true calls. We propose two different contexts for designing such models as a solution to this problem and investigate their performance.

On the other hand, utilizing features of next generation sequencing technology in an even more efficient way, requires the ability to multiplex high number of samples in a single experiment run. In that case, appropriate barcoding, that is robust to various sources of noise in the machine, becomes paramount. Designing such barcodes in an efficient way is a challenging task which is addressed in detail as another problem of this thesis.

We make two contributions. One, we propose an algorithm for barcoding multi-

plexed RADSeq samples. Two, we propose an algorithm for the statistical selection of presence/absence markers on the basis of RADSeq data on two related individuals. Operating characteristics of our methods are explored using both simulated and real data.

DEDICATION

To my Mom and Dad for their everlasting support

To my beautiful and beloved wife for her endless love

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Dr. Charles D. Johnson, the director of GENO group in Texas A&M AgriLife Research, for his advice and support throughout the course of my research and also providing funding for this thesis and a good research and work environment.

I would also like to thank Prof. Alan Dabney, whose guidance, encouragement and support enabled me to finish my Master's thesis successfully. I thank also Prof. Byung-Jun Yoon as my co-chair and Prof. Edward R. Dougherty and Prof. Chamberland for serving on my committee and all their constructive advice.

My special thanks go to Dr. Scott Schwartz for his consultations and all discussion we had throughout the last two years which made a huge impact on my learning and knowledge of bioinformatics especially in Next-Generation Sequencing. In addition, I would like to thank my dear friend, Dr. Mehdi Maadooliat with whom I had a lot of thoughtful statistical conversations which enlightened me and helped me finish the methodology part of my thesis successfully as well.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience, especially Mrs. Tammy Carda for her efforts and patience in responding to all our administrative concerns.

Last but not least, I would like to thank my parents and sisters for their constant support and encouragement. I would like to express my greatest gratitude to my beloved wife who is also my best friend, for her invaluable love and sacrifice she has made to be on my side as a perfect mate which ultimately made me successful throughout the course of my Master's study.

NOMENCLATURE

CE	Capillary Electrophoresis
CL	Confidence Level
DArT	Diversity Arrays Technology
DNA	Deoxyribonucleic Acid
GWAS	Genom Wide Association Study
MLE	Maximum Likelihood Estimator
NGS	Next Generation Sequencing
PAV	Presence/Absence Variation
RAD	Restriction site Associated DNA
RFLP	Restriction Fragment Length Polymorphism
RNA	Ribonucleic acid
SBS	Sequencing By Synthesis
SNP	Single Nucleotide Polymorphism
SSR	Simple Sequence Repeat
STR	Short Tandem Repeat
VNTR	Variable Number Tandem Repeat

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

With the advent of new technologies in sequencing, many interesting problems are introduced that have not been addressed yet. Although we have not reached the point of $1000 full genome sequencing yet, huge improvements have been taking place in minimizing the cost of sequencing billions of base pairs in a short amount of time.

One of the challenges in this context it to efficiently utilize the features of sequencing machines to even more lessening the cost of sequencing per base. This has led to the concept of multiplexed sampling and ability to sequence tens to hundreds of samples at each run of sequencing machine. This could be accomplished at the cost of jeopardizing quality and reliability of the experiment, if not done with adequate deliberation.

Efficiently designing high quality barcodes with special characteristics in order to minimize the risk of false demultiplexing becomes crucial. One of the contributions of this thesis is to introduce a fast algorithm to produce such barcodes specifically designed for Illumina sequencing machines. It takes into account all potential sources of noise that can lead to false demultiplexing in the design process and makes barcodes robust to such interference.

Another sequencing related task that is considered here is genotyping and marker discovery. Generating dense and high resolution genetic maps requires high number of good quality genetic markers. In recent years, the use of restriction site associated DNA (RAD) markers has been increased due to its novelty and special characteristics in detecting polymorphisms in the genome of species.

RADSeq is the combination of RAD method in generating markers with next

generation sequencing (NGS) technology which can generate millions of RAD markers for further analysis. One of those markers are presence/absence markers which are present in one sample and absent on the other and are informative in this way that show variation in the genome that caused this presence/absence. The counts of such markers are highly correlated between samples of an experiment due to various reasons such as being in the same library preparation procedure and same run of sequencing machine. Hence, calling the true states of presence and absence in those samples is not just based on zero or non-zero counts of markers. It is worth mentioning that one of the important applications of designing high quality barcodes which mentioned earlier, is the ability to perform multiplexed RADSeq sequencing. This will increase the number of produced markers which is very important in genotyping.

Knowing true states of such RAD markers is important as if they are missed and not truly absent, we can impute them or increase the depth of coverage as a try to catch them. Moreover, to decide which marker should be participated in further analysis, it would be helpful if their true presence/absence states are known. There should be a model that can determine true state of a such markers with a certain level of confidence. This problem is addressed as the second contribution of this thesis.

The structure of the thesis is as follows. In Section 2, we briefly review Next Generation Sequencing (NGS) technology, data processing in it and its applications. In Section 3 we introduce a fast algorithm to generate high quality barcodes for Illumina sequencing machines in order to perform multiplexed sequencing [31]. Different types of genetic maps as well as different kinds of genetic markers as key elements of genetic maps and other genotyping analysis, are studied in Section 4. In Section 5 we mainly focus on RAD markers and RADSeq process by investigating their advantages and potential sources of noise and bias which should be considered in

the analysis. Finally in Section 6 we develop two different methods, one suboptimal and the other based on mixture modeling for the problem of calling true states of presence/absence RADSeq markers. The performance of our methods is tested for both simulated and real data. Last section of this thesis is a concluding discussion in which we forecast the follow-up work we plan to do as part of my statistics Ph.D.

2. NEXT GENERATION SEQUENCING (NGS)

In recent years decoding DNA of different species has always been essential for all branches of biological sciences and a major part of bioinformatics. The method of determining the order of four bases, Adenine (A), Guanine (G), Cytosine (C) and Thymine (T) in a strand of DNA is called DNA sequencing. Knowledge of DNA sequences is now one of the basic elements of researches in bioinformatics, biotechnology and biological sciences.

Among all methods of sequencing DNA, capillary electrophoresis (CE)-based Sanger sequencing has become widely adoptable and genetic information of almost any biological system can be demonstrated. There are some limitations to this method such as speed, scalability and the resolution needed to explore essential information researchers need [21]. To overcome those limitations and reduce cost per base while increasing the throughput, other alternative methods have been introduced. 454 Pyrosequencing and Illumina (Solexa) sequencing can be listed as the most common of those alternative methods. Parallelized Pyrosequencing which was developed in 454 Life Sciences and then acquired by Roche Diagnostics can generate about five hundred million bases of raw sequences in just few hours. However, Illumina sequencing can offer even more throughput by generating billions of bases in a single run [32].

These novel methods rely on parallel processing of cyclic examination of sequences from spatially separated clonal amplicons. Amplicons are pieces of DNA or RNA that are sources or products of natural or artificial amplification processes being done on them. This parallelized processing is the key element of high throughput and low cost feature of these methods, although the read length is shorter compared

to Sanger sequencing method [32].

In what follows, we first investigate big picture of NGS on how it works and what are the steps for sequencing a sample DNA. We then introduce one of Illumina sequencing machines, HiSeq 2000, from which the data of this thesis is obtained. Data processing in the context of NGS is also studied and finally some applications of next generation sequencing are investigated at the end of this chapter.

2.1 Overall View of NGS

In principle, the main concept of Illumina sequencing or Next Generation Sequencing (NGS) is similar to CE. It identifies the bases of each fragment by the signals emitted as each fragment is resynthesized from a DNA template strand. This process is being done in massively parallel fashion for millions of fragments instead of being limited to a single or few DNA fragments as in Sanger sequencing method. This is why gigabases of data for a long genome can be generated in single run. The genomic DNA is first fragmented into a library of small fragments that are sequenced in parallel. The produced strings of bases are called reads which are used for further analysis of the species under study.

All Illumina systems consist of flow cells in which there are different lanes. At each lane there exist many tiles that contain millions of clusters in which DNA fragments are being sequenced. Three major steps in NGS using Illumina systems sequencing workflow are library preparation, cluster generation and sequencing.

2.1.1 Library Preparation

DNA is first fragmented and Illumina adapters are ligated to both end of each fragment. These adapters help the DNA strand binds to the clusters on the tile of flow cell. Then adapter ligated fragments are size selected and purified according to the purpose of the analysis.

2.1.2 Cluster Generation

Single molecules are amplified on the flow cell to be prepared for high throughput sequencing. The flow cell has dense oligonucleotides (Oligos) attached to its surface. After DNA molecule binds to oligos on the flow cell by their ligated adapters, they are extended to create copies. Each library fragment is amplified through series of bridge amplification. The result is hundreds of millions of unique clusters on the flow cell in which reverse strands are washed away. The library is ready to be sequenced after sequencing primers are attached to DNA templates and cluster is generated.

2.1.3 Sequencing

Sequencing is done base by base for all clusters simultaneously and in parallel using four fluorescently labeled nucleotides. Those four nucleotides compete to bind to the DNA template. This competence ensures high accuracy of sequencing. After each cycle of sequencing clusters are excited by a laser, emitting a color that identifies the last added base. This chemistry reads one base at each cycle. This is the way how sequencing step is done in Illumina sequencing machines.

2.2 HiSeq 2000 Systems

Illumina has different sequencing systems such as GAII and HiSeq. The data for this thesis are generated by HiSeq 2000 system. HiSeq Sequencing Systems uses widely adopted reversible terminator-based sequencing by synthesis (SBS) chemistry. HiSeq 2000 sequencing system produces data at the fastest generation rate. It has also lowered the cost of whole-human genome sequencing to unmatched levels [22]. It uses dual flow cell instrument to maximize experimental flexibility. Therefore it can run applications that require different read lengths simultaneously. In this system images clusters are on both surfaces of flow cell which results to highest output and

fastest data rate.

In the mentioned system, library preparation can be performed using llluminas simplified TruSeq sample prep kits. There is an automated cluster generation system named cBot which does the task in just less than 10 minutes hand on time. In HiSeq 2000 when using dual flow cell, up to 3 billion clusters passing filter and up to 6 billion paired-end reads are produced. The throughput of this system in case of 2×100 bp run is up to 55 billion bases per day [22].

2.3 Data Processing

The output of sequencing machine after sequencing step is finished, is just raw sequencing files of intensities at clusters, not interpretable by any biologist or bioinformatician. To change them to something biologically meaningful, data analysis step should be accomplished. HiSeq has control software which offers real-time analysis processing that automatically produces image intensities and quality scored base calls on the computer. Base Calling is the process of transforming image intensities at each cycle of clusters to one of A, G, C or T bases. Quality score for each base is a measure of certainty in calling that base for that specific read in that specific cycle.

In HiSeq system, in combination with Consensus Assessment of Sequence and Variation (CASAVA) software, GenomeStudio data analysis software provides intuitive graphical analysis [22]. The output of data processing steps are fastq files which contains reads along with their base calling quality scores. It also contains headers for each read sequence which carries information about the cluster that the read has come from and its position on that cluster, etc.

These fastq files are then used in further analysis such as being mapped to a reference genome to find Single Nucleotide Polymorphism (SNP), De Novo assembly

analysis when no reference genome exist, RNA Seq analysis when we sequence RNA instead of DNA fragments in sequencing machine and, etc.

2.4 Applications of NGS

So far we know that the output reads of NGS, from 35 bp in Illumina to 300 to bp in 454 technology, is much shorter than reads generated by Sanger sequencing, 700 to 900 bp. But in NGS we have much larger number of reads which can be used in specific applications that utilize specific data format and are not feasible by Sanger sequencing output reads.

One class of such applications and analysis involve using a reference genome. For instance, all existing species in the sample can be distinguished by mapping all the reads to candidate species reference genomes. The number of reads that map to specific genome are proportional to the presence of that species in the sample. Another application in this class is finding SNPs in the sample by mapping it back to its reference genome. Those single nucleotide differences can be determined by different algorithms using high throughput data, the main characteristic of NGS, by comparing it to the reference genome.

Another class of applications are those where no reference genome exist. Most of bioinformatics analyses are in this group as genome of most of species have not been completely sequenced yet. De novo sequence assembly and making contigs out of sequence reads can be named as the most important application of this class. These made contigs can then be used in further analysis such as making a reference genome for different species. Short output of NGS is a deficiency for this class. There are many software which utilize different algorithms to do the task but in all of algorithms, the longer sequences are, the better a consensus contig can be generated. To overcome this drawback, paired end sequencing is used in which both

ends of a fragment of defined size are sequenced.

Although many classes of applications which benefit from NGS data can be listed here, explaining all of them would be out of the scope of this thesis. Among those classes the one that makes use of RADSeq data is the basis of the research that has been done in this thesis. The details of RADSeq data and the applications and analysis based on this type of NGS data are all discussed in detail in following chapters.

3. FAST DNA BARCODE GENERATING ALGORITHM USING RADIX CODING METHOD*

This chapter introduces a fast algorithm for generating high quality barcodes [31]*, which enables us to perform a high throughput sequencing experiment for multiplex samples.

High multiplexing of samples is critical for Next Generation Sequencing (NGS) where within one lane 10s or 100s of samples can be sequenced simultaneously. Often the limited factor is the number of unique DNA sequences that can be used as molecular tags, i.e. barcodes. We propose the creation of a new algorithm to generate these sequences for sample barcoding. Devising algorithms that make the designing possible in a reasonable run-time and high quality output has been always challenging.

In this chapter we introduce a new search method which utilizes proposed coding of features named Radix Coding. This method facilitates searching between multi-feature objects where all the features need to be considered in the search process. This method can also be generalized and applied to similar applications and outperforms common and time consuming search algorithms as it reduces the run-time by a significant factor.

---

*Reprinted, with permission, from "Fast DNA barcode generating algorithm using Radix Coding method" by A. Nikooienejad, R. Metz, B. J. Yoon and C. D. Johnson, 2012. *IEEE International Workshop on Genomic Signal Processing and Statistics, (GENSIPS)*, 26-30, Copyright © 2012 by IEEE.

3.1 Introduction

With the advent of new high throughput NGS technologies, the ability to combine, sequence and identify the reads from multiple samples simultaneously in an accurate, cost effective manner is paramount. Current barcoding schemes involve placing a molecular barcode at a known location within the insert to be sequenced (5' or 3' end) or by performing a separate read (or reads) using an additional primer or primers targeting a region of the library outside of the insert, Fig. 3.1. Both of these strategies have pros and cons relating to time and cost of library preparation, maximization of data, sequencing recipe and data handling and/or processing.

As new methods are being developed to harness the power of NGS, custom library preparation is becoming more common. When developing a novel library preparation protocol, it is important to consider the most efficient barcoding scheme that will maximize data acquisition, allow unambiguous sample identification, and have minimal adverse effects on sample performance throughout the library prep process and during sequencing.

For the purposes of this discussion, we will focus on Illumina-based sequencing as it currently affords the highest throughput [29]. The standard Illumina TruSeq paradigm uses a separate indexing read targeting the i7 barcode, Fig. 3.1. This is important as the metrics of an Illumina sequencing run are determined by the first few cycles and are therefore dependent upon the random nature of insert sequences to supply diversity [20]. For custom procedures such as amplicon sequencing or restriction enzyme associated DNA sequencing, it is often more convenient to introduce barcodes at the beginning of the insert [12]. When this is done, careful consideration to create balance as well as diversity is critical for accurate down stream signal processing. In addition, the decision to use certain barcodes should consider adjacent

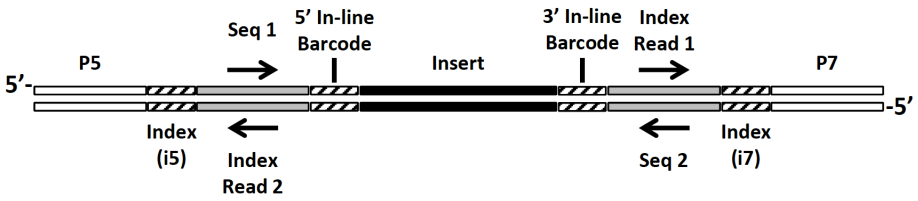

Figure 3.1: Illumina-compatible library structure with potential indexing sites

sequences such as the primer used for amplification (amplicon sequencing) or the restriction enzyme used to reduce the possibility of artifacts during amplification.

In what follows we introduce an algorithm to design high quality barcodes in a fast and efficient manner. This algorithm employs a novel method for searching multi feature objects in order to minimize redundancy and decrease processing time resulting in a faster and more efficient method than brute-force searching.

3.2 Problem Statement

Today's NGS technologies allow high throughput sequencing on an unprecedented scale. This has made it possible to multiplex large number of samples at a time, which necessitates a way to distinguish between samples in an accurate and efficient manner through molecular identification, or barcoding.

The factors contributing to generating appropriate barcodes are determined by the properties of the cluster detection and base calling methods used by Illumina. Using these constraints, we identified parameters and applied their properties as constraints in generating a set of molecular barcodes for use with high throughput NGS experiments.

3.2.1 Having Short Length

Shorter barcodes tend to be superior because the possibility of sequence errors increases with barcode length. If the barcode is in-line with the target insert sequence, the shorter the barcode the more sequence cycles can be devoted to sample data.

3.2.2 Balance in All Base Positions

When barcodes are located at the beginning of an Illumina-based sequencing library, they must be balanced since the first 4 cycles are used to set the cluster and base calling metrics of the entire run. If all the barcodes sequences started with the same nucleotide, the instrument cannot distinguish individual clusters. Resulting in poor data quality and possibly complete run failure.

Ideally the base ratio across the flow cell should be equal for all bases. Since we have only 4 bases: A,G,C,T therefore we can have exact 4 balanced barcodes at each base position. This means the barcode scheme will work best when used as groups of four balanced for base at each position.

3.2.3 Base Diversity in Each Barcode

In addition to balancing the reads during the metrics- setting first few cycles, sequence diversity helps establish the clusters from which data will be obtained. If a barcode is at the 5' end of the first sequence read, the clusters will be called more efficiently if there are more unique barcodes. For instance, if there are two clusters close together, and they have the same barcode, the machine might call them as one cluster.

As the sequencing processes continues there will be two different signals coming from that 'cluster', because in fact it contains a mix of two fragments and it will be dropped during the error correction process. If the two clusters close together have

different barcodes, they will give different signals initially, and will be called different clusters.

3.2.4 GC Content of Each Barcode

In general, for all molecular biological procedures balanced GC and AT content is considered the best situation. GC base pairs form three hydrogen bonds and require more energy to which may result in PCR bias and inaccurate sequencing. This is especially a problem when there are long stretch of GC rich sequences.

On the surface, consideration of GC content in barcodes seems unimportant since the sequence is small. However, depending on the process used, a barcode may contribute to GC content of adjacent sequences in a significant way. This is particularly relevant to Restriction site Associated DNA Sequencing (RAD-Seq) procedures where the barcode is adjacent to a restriction enzyme site, which may be GC rich (e.g. FseI site is GGCCGGCC).

3.2.5 Hamming Distance Between Barcodes

If barcodes are too similar, an error during sequencing might make one barcode look like another, resulting in a read being attributed to the wrong sample. This can be addressed by setting the hamming distance. Hamming distance is defined as number of differences in base-by-base comparison of two barcodes. A hamming distance of 2 requires two errors to get a barcode to look like another, which is much less likely to occur. For our process, we also allow one mismatch in base calling step and thus set the minimum hamming distance between each pair of barcodes to be 3. The resulting list of constraints for generating our list of barcodes is as follows:

1. Barcodes of each table of four have to be balanced at each base position.

2. Barcodes must have diversity in their bases across the tables. This is set by

user by the maximum allowance of repeats for each base at each barcode.

3. The GC content of each barcode has to be less than a parameter as percentage of barcode length which is set by user.

4. The hamming distance between each pair of the barcodes in the final list should be equal or greater than 3.

According to these characteristics, the main advantages of these barcodes compared to TruSeq ones, can be summarized in following points:

- The length of TruSeq barcodes is fixed at 7, whereas in our method it can be set to any number by user.

- The hamming distance parameter is not controllable in TruSeq barcodes, while in this method it is an important parameter that can be set by the user.

- The number of produced barcodes is very limited in TruSeq which is a huge drawback, whereas in this method we get the maximum number of possible barcodes with the given constraints.

3.3 Method & Algorithm

Implementing those constraints discussed in section 3.2 in a full search manner has a high computational cost. We can convert it to a problem of filtering the list of all possible barcodes of length N. The number all possible barcodes is 4^N as we have 4 possible bases at each position.

Our algorithm to accomplish that filtering can be divided into three steps:

Step 1: Generating Initial possible tables of four barcodes

Step 2: Filtering generated balanced tables of barcodes based on all the constraints except hamming distance

Step 3: Filtering the output list of the last step with respect to hamming distance constraint

3.3.1 Generating All Possible Tables of 4 Barcodes

Because the barcodes should be balanced in each base position in a table, each table should contain 4 barcodes (4 different bases at each column).

We first make all possible tables although it contains redundant tables with different permutations of barcodes. Then we filter tables instead of each barcode. In this way we can make sure that the balance constraint is always preserved.

The number of all possible tables of 4 barcodes of length N is computed as follows: Each table has 4 rows and N columns. Having balanced barcodes at each position means having unique and non redundant bases at each column of the table. Since we have 4 different bases: A,C,G,T, thus we have $4! = 24$ different columns. Therefore, the number of all possible redundant tables is then equal to 24^N.

Bases A,C,G,T are coded to numbers 0 to 3 respectively in the entire algorithm and also programming steps. We decode them at the end while printing the results. We also avoid using many number of 'for' loops. For instance, recursive method is used to generate all different 24 combinations of columns and then we index those by numbers 0 to 23. Each row in resulted matrix contains column indices and represents a table of 4 balanced barcodes. This matrix, which we name it **M1**, therefore contains all possible balanced tables.

In the following subsection we start the second step of those steps described previously, which is filtering matrix **M1** according to constraints we have except for hamming distance.

3.3.2 Filtering Generated Balanced Barcodes

The filtering step can be divided into two phases. In the first phase we apply GC content and Diversity constraints. In the second phase, we try to remove redundant tables.

3.3.2.1 First Phase of Filtering

Matrix **M1** has 24^N rows and N columns representing all possible balanced tables. To delete those tables whose at least one barcode doesn't satisfy diversity constraint, we first decode that matrix by replacing column indices with the original column itself. By this, the number of rows of the new matrix changes to 4×24^N. Then we inspect each row to find number of repetitive bases in that row. If the number of repeats of each base in a row is more than user parameter we delete that row's table. We apply the G-C content constraint simultaneously here the same way and if a row (barcode in a balanced table) violates the user set parameter, we delete the whole table containing that row to preserve balance. Since we have numbers instead of alphabets, checking these two constraints is very faster than the usual case.

We again encode the rows to their column indices to have less number of rows. The output of this step is purified encoded matrix in which all the tables contain balanced, diverse and required G-C content. We store the output of this step in matrix **M2**.

3.3.2.2 Second Phase of Filtering

Although matrix **M2** is filtered in terms of the first three constraints discussed in section 3.2, it contains many redundant tables in which just the permutation of barcodes are different. The total number of barcodes in the initial list, matrix **M1** before encoding into indices is 4×24^N while the total number of possible barcodes

is 4^N. It shows how much redundancy we have in that list. We now define the main tool helping us through out resolving this issue and also the heart of this method. Final step of the method which is applying hamming distance filter is also utilized with this tool.

Radix Coding Method: One of the keys to the barcode development process is a fast and efficient algorithm.

When there is a vector of numbers between 0 and $n-1$, It can be presumed that it is a number in base n and therefore has a decimal value. Since every decimal number has a unique representation in base n and each number in base n has a unique decimal value, this transform is one to one and onto and therefore two spaces have bijection and therefore can be used interchangeably. Each digit in the number in base n could be considered as a feature of a multi feature object. When we convert a combination of numbers in base n to its decimal value, we call it *Radix Encoding* and when we find the base n representation of a decimal number we call it *n-Radix Decoding*.

To find which rows are producing the same list of 4 barcodes in matrix **M2**, since its rows' elements are numbers between 0 and 23, we can apply 24-Radix Encoding to that matrix and therefore change it to a vector of correspondent decimal numbers. Each combination of columns has equivalent set of combinations which result to same tables of barcodes but with different orders. Because four barcodes of each table can be rearranged in 4! ways, there exists 24 different equivalent tables for each table. Each row of **M2** represents a table, thus there exists 24 different equivalent rows of matrix **M2** to the row under investigation.

These equivalent tables (rows) can be found easily by using our designed algorithm which produces all equivalent combinations given starting column, using the dictionary we have generated for different combinations of columns. Each row can

be presented by their unique decimal index after applying Radix Encoding. For each row of our matrix, we find 24 different decimal numbers which are Radix Encoded numbers of equivalent tables for that row. Those numbers are flagged in the matrix and the table under investigation is kept. One of the benefits of flagging those equivalent tables is that we don't investigate this process for all the rows. These steps are done for those rows that have not been out flagged by previous rows. This can make a huge difference in terms of the run-time of the algorithm which reduces the run-time from $O(n^2)$ to $O(m)$ where $m < n$, [10].

Here we see the major benefit of Radix Coding and how it facilitates searching for equivalent rows by searching through a vector of numbers instead of a 2D matrix. The output of this step would be a vector of decimal numbers, representing tables that are not equivalent in any pair and also satisfying all previous constraints. The only thing remains to be filtered is the hamming distance constraint. We store the output of this step in matrix **M3**.

3.3.3 Applying Hamming Distance Filter

We define two balanced tables, t_i and t_j which consist of barcodes $b_{i1}, ..., b_{i4}$ and $b_{j1}, ..., b_{j4}$ respectively, consistent in terms of hamming distance if they satisfy the following condition:

$$\text{Consistency of Tables } t_i, t_j : \begin{cases} d(t_i, t_j) = 0 & \text{if } i = j \\ d(t_i, t_j) \geq hdist & \text{if } i \neq j \end{cases} \tag{3.1}$$

where 'hdist' is the user set parameter for hamming distance and $d(t_i, t_j)$ is defined as follows,

$$d(t_i, t_j) = \min(H(b_{ik}, b_{jl})),\ \forall k, l \in \{1, ..., 4\}$$

$H(b_1, b_2)$ is the hamming distance between two barcodes b_1 and b_2.

In this section, we use matrix **M3** to find as much consistent tables as required and if we couldn't, we find the largest set of consitent tables.

3.3.3.1 Creating Binary Comparison Table

First of all we need a comparison table to show which barcodes in **M3** are consistent with which barcode, with respect to user set parameter for hamming distance, 'hdist'. It can be used as a look up table for further analysis.

This table is actually a n by n binary symmetric matrix **B** whose rows and columns represent barcodes of **M3** and its elements are just zeros and ones. If they have a hamming distance of less than parameter 'hdist', that element of **B** is set to 1 and it is 0 otherwise. To construct **B** with the lowest computational cost, there are two points that need to be considered.

First, even for filtered matrix **M3** there exist redundant barcodes each belong to different balanced tables. Since, **B** acts as a look up table it should contain unique barcodes present at **M3**. Therefore, we need another redundancy remover acting on **M3** and hence, we can employ Radix Coding method for another time.

After decoding rows of **M3** and replacing them with actual columns, each decoded column at each table consists of numbers 0 to 3 representing bases A,C,G and T respectively. Thus, we can apply 4-Radix Encoding to assign a decimal value to each row of expanded **M3**. We name this Radix Encoded version of expanded **M3** as vector **DM3**. Now we use just one member of each set of equal elements in **DM3** as unique barcodes. It is notable again how Radix Coding method accelerated searching redundant barcodes of length N in matrix **M3**.

Second, we need to compare those unique barcodes one by one and check their hamming distance. To avoid using 'for' loops in implementation of the algorithm,

we reshape the matrix of unique barcodes from 2-Dimensional to 3-Dimensional and again to 2-Dimensional appropriately to slide each barcode over all others and produce one by one comparison result which yields to required binary matrix **B**.

3.3.3.2 Finding Required Set of Tables

As mentioned in previous part, **DM3** is a vector whose elements are Radix Code of expanded **M3**'s rows. On the other hand we index each column of **B** with its equivalent Radix Code so that we can extract hamming distance information of each element of **DM3**, using **B**. By placing each four consecutive elements of **DM3** in a same row, we reshape **DM3** to a new matrix **M4** whose number of rows is one quarter of length of **DM3**.

Since each four row of expanded **M3** was a table, each row of matrix **M4** actually represents a table and therefore we can do the filtering easier by removing or adding a row to our final largest set. We mainly do that because we need to add or delete a table instead of a barcode just to preserve balance and all other constraints. If any barcode of a table conflicts with any other barcode of another table according to matrix B, we name those two tables as conflicting tables. For each table i, or row i in **M4**, the list of all its conflicting tables can be obtained by the union of all other tables that contain conflicting barcodes with at least one of the barcodes of table i.

By means of matrix **B**, row number of tables that conflict with table (row) i of **M4** in i^{th} element of a can be stored in cell array **C**. This cell array can then be used to filter the tables according to hamming distance. By this, we are upgrading our look up table of conflicting barcodes to a look up table of conflicting tables.

The important point here that is worth mentioning is the way we construct and grow the sets. The general flow chart of proposed algorithm is described in Fig. 3.2. According to that flow chart, at each time we add a table to a growing set that is

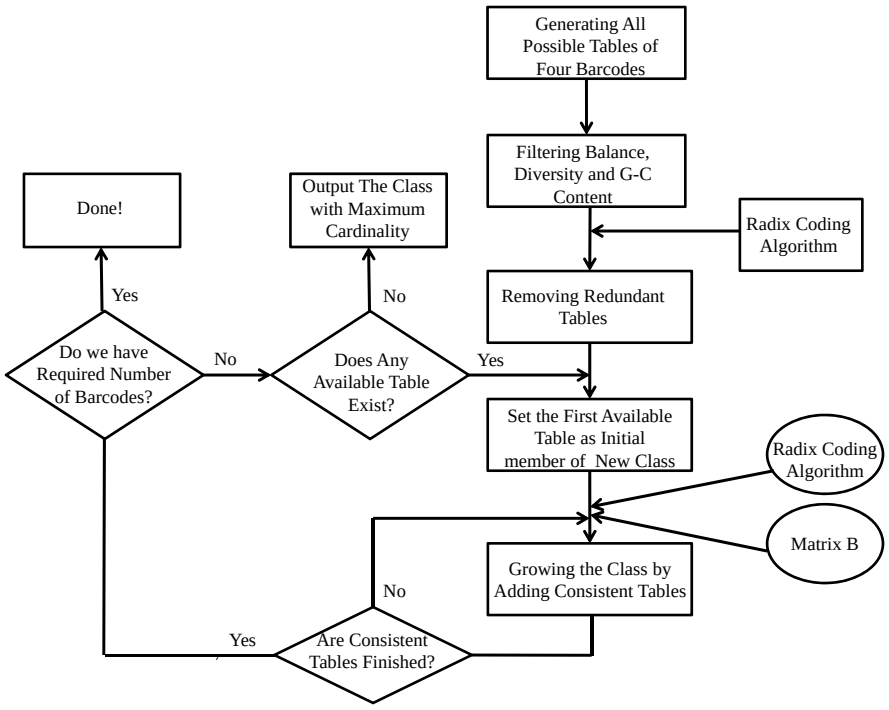

Figure 3.2: Flow chart of the proposed algorithm

consistent in terms of hamming distance with *all* of the previously added tables of that set. Consistency relation is defined at the beginning of this subsection in (3.1). Moreover, if a table is assigned to a set, it is removed from available list of tables and can not be used any more.

Thus, if number of required barcodes is ignored as the stopping condition of the algorithm, that growing set actually stops growing if there is no other table available to be added to the set. If still another tables available that are not still assigned to any set, another growing set is initiated until no more tables can not be added to that set. The whole algorithm stops whenever no unassigned table is remained. This is how we construct our sets of consistent tables.

Based on what is just described, following holds for all balanced tables of barcodes:

$$\begin{aligned} &1.\ \forall t \in T \ \Rightarrow \exists i; t \in C_i \\ &2.\ \forall i, j \ \Rightarrow C_i \cap C_j = \emptyset \end{aligned} \tag{3.2}$$

where C_i is each growing set, t is each balanced table and T is the set of all purified balanced tables.

We can deduce from (3.2) that the way we construct the sets of tables is partitioning the total set of tables and we can also call those sets as classes. Therefore, no matter by which member we start to grow that class, we always get the same class.

By this, the problem is now changed to find the class whose cardinality is as required or if there is not any, find the one with largest cardinality. This is what presented in Fig. 3.2.

To accomplish what we want, we start from the first available table and put it as the only member of the first class. Now to grow that class, we search through all other tables. Using cell array **C**, if a candidate table doesn't conflict with all previous members of the class, it is added to the class and is not available any more. Otherwise it will be rejected. This is how a class is created. For each class, this has to be repeated for all available tables in **M4**. Those who has not joined any class yet.

The algorithm stops whenever the cardinality of the undergoing class is at least equal to what the user wants. Otherwise we have to finish partitioning to find the class with largest cardinality. This mainly depends on how many barcodes user wants with respect to the length and other constraints such as hamming distance. Number of required tables to set in the algorithm can be easily calculated by required number of barcodes as each table consists of 4 balanced barcodes.

3.4 Results

We tested proposed algorithm for barcodes with two lengths of 5 and 6 bases. Parameter 'hdist' which represents hamming distance set to be 2 and 3. Hamming distance of at least 3 is more of our interest since it allows two mismatches for sequencing error and one while base calling. Diversity parameter is set not to allow more than 2 repeats for each base at each barcode. The G-C content parameter is %60 and %50 for length 5 and 6 respectively.

Considering all eminent factors such as short length, reasonable GC content and appropriate diversity, these constraints make barcodes with just highest quality compared to other existing methods, e.g. TruSeq, in a fast and simple manner using Radix Coding method. This algorithm is implemented in MATLAB® on a computer with 16 GB of RAM and a 3.4 GHz intel Core i7 processor with 4 Cores.

The number of maximum possible barcodes with mentioned constraints, which is 4 times cardinality of the largest possible consistency equivalence class for tables, along with correspondent runtime to find such barcodes are provided in Table. 3.1.

There exists no such application for generating barcode which takes into account all constraints, thus it is difficult to compare the run-time or other properties. However, in very similar applications generating barcodes of length 6 or higher with hamming distance of at least 3 seems impossible in terms of run-time and without Radix Coding, even the process of generating barcodes of length 5 and hamming distance of at least 2 would have been around 500 times slower on a computer with similar capabilities. Accordingly, Radix Coding completed the task %99.998 faster than common programming techniques. The run-time could even be faster in a Unix based programming language.

3.5 Conclusion

In this paper, we introduced a new search method utilizing coding features by proposed method named Radix Coding. This method is beneficial where objects have more than one feature and all of the features have to be considered. We employed this search method to generate barcodes of special properties for NGS high throughput experiments in a very fast and easy manner.

The constraints used in designing the barcodes make them more advantageous over TruSeq ones by offering control of diversity of bases, GC content and hamming distance between barcodes. These parameters can be set each time according to the expected quality of the output.

One important point to be noted here is maximum number of possible barcodes with hamming distances of at least 2 and 3 reported in Table. 3.1. For instance it is impossible to have more than 52 barcodes with hamming distance of at least 3 and length 6 and GC content of %50. Without exploiting proposed algorithm and actually Radix Coding Method, producing such results seemed impossible and very time consuming. Applying and adapting this method as a powerful tool to other multi-feature object searches especially in bioinformatics processes can be considered as future works.

Length	hdist	Max. Possible Barcodes	Runtime (hr)
5	2	128	2.73
	3	28	4.01
6	2	360	8.26
	3	52	12.21

Table 3.1: Maximum number of possible barcodes and correspondent run-time for different lengths

4. A REVIEW ON DIFFERENT TYPES OF GENETIC MARKERS AND THEIR APPLICATION IN GENOTYPING

Determining and finding responsible genes for specific characteristics of species is one of the important applications of genetics. Finding specific region of a maize DNA that makes it having high heat tolerance is one example of such applications, or seeking answer to the question of what are those genetic variants that make specific race of quarter horse be more resistant to some diseases than other races. In order to investigate how these can be done in detail and prior to exploring different kinds of genetic markers, we first discuss the basics and terminologies regarding those.

Observable characteristics in species are called *phenotype*. Those characteristics could be biochemical and physiological properties or even behaviors. For instance eye color is considered as a phenotype in humans. Different variants of a phenotype is called trait which in our example blue, green or brown are different traits of eye color phenotype. Different colors in rose flower or the tail length in cats could be other examples of phenotypes. Phenotypes are originated from either expression of organism's genes or even in some cases environmental and developmental factors.

On the other hand, all expression instructions inherited in terms of codes in organism's genome is called genotype. As mentioned earlier, one of the sources of having different phenotypes is genotype along with environmental factors. Thus, same phenotype between two organisms does not necessarily show similarity between their genotype and vice versa, because of environmental and developmental factors that might change phenotypes. The distinction of genotype and phenotype was first proposed by a Danish botanist named Wilhelm Johannsen.

In following sections we first investigate different types of genetic maps, one of

the most important part of genotyping in which markers are highly used and then we review various common types of genetic markers.

4.1 Genetic Maps

In order to better do genotype analysis in an organism and investigate genes behind a specific phenotype, it is better to know functionality of different genes in DNA, their relative location on chromosomes and how they are inherited during meiosis. Genetic map is generally a graphical representation of location and arrangement of genes on each chromosome of an organism. If the reference genome for an organism is unknown, no genetic maps exist and it is improved and completed when the whole genome of the organism is sequenced.

According to [34], gene mapping can link diseases to their correspondent genes that they come from which can help better understanding the disease as well as curing it. It, on the other direction, can unravel the functionality of newly found DNA clones by investigating the correlation with previously found and described variant phenotypes. It can also distinguish heritable and non-heritable components of complex traits and the mechanism by which they interact.

Different variants for a specific gene are called its alleles. Monomorphic genes are those that have only one variant or one allele. Similarly bi-allelic ones are those that have two different variants and so on. Today, despite pre-recombinant DNA era when only those genes that had known phenotype were called loci of DNA, geneticist use the term *locus* to describe any distinguishable DNA segment by any forms of genetic analysis, whether the functionality for that segment is unknown or known [34]. Those loci whose functionality is still unknown are called anonymous loci.

Now having this definition in mind, genetic map is simply the distribution of a set of loci within the genome of an organism. The loci under investigation could

be mutually independent or could be relevant to each other based on some factors such as functional or structural homologies. Mapping of those loci can be done in different levels of resolution. In the lowest level just the chromosomes where each locus resides in are identified, in one step above that loci are assigned to sub chromosomal regions. In a higher level of resolution, relative order and approximate distances between individual loci in a linked set can be determined. The ultimate resolution is attained when loci are mapped onto the DNA sequence itself [34].

There are three types of genetic maps, linkage, chromosome and physical map which are explained in following subsections.

4.1.1 Linkage Maps

This is also referred to as recombination map and is for those genes that has at least two alleles. Therefore monomorphic loci cannot be mapped in this type of map. They are generated by counting the number of offspring that inherit either parental or recombinant alleles. This type of analysis allows determining whether loci are linked to each other as well as their relative order and distances if they are linked.

That would be worthy to define recombination and explore its properties here. At an early stage of meiosis, the two chromosomes each from one of the parents lie side by side. Then each chromosome is duplicated into two sister chromatids connected at centromere. Therefore there are four chromatids beside each other which are called tetrad. At the next step known as cross over, two non-sister chromatids adhere to each other in a semi-random fashion at regions called chiasmata. Chiasmata represent points where crossing over between those non-sister chromatids, each from one parent, can occur [40].

Chiasmata do not occur completely at random as they are more likely to farther away from centromere and it is rare to find two chiasmata very close to each other

[40]. Each gamete inherits one of those four resulted chromatids after cross over step. If no cross overs occur, the chromosome in the gamete would be the exact replicate of the chromosome in one of the parents. On the other hand, since it is possible that more than one cross over occur in meiosis, it is likely that the haploid chromosome in the gamete consists of loci from the two parents. Number of such loci depends on the number of cross overs occurred during meiosis.

For instance, suppose parent 1 has allele 'A' for locus A and allele 'B' for locus B and parent 2 has allele 'a' and 'b' for those two loci respectively. In case of having no cross over, the resultant chromosome has 'AB' or 'ab' as its allele for those two loci. In the case of having one cross over occurs between those two loci, we have new combinations 'Ab' or 'aB' in resultant chromosome. These two new combinations are called recombinant types. The important point here is that if the number of occurred cross overs between those two loci is even, the result is not distinguishable with 'AB' and 'ab' as they get back to their original case where no recombinant alleles have been made. However, if the number of cross overs is odd we have recombinant type alleles in resultant chromosomes.

Linkage between two loci means that it is less likely that a cross over occurs between them which in another words means separating them. The more two loci are linked, less likely to be separated. Suppose the proportion of recombinant alleles in gametes is r which is also called recombination fraction.

4.1.2 Chromosome Maps

Another name for this map is cytogenetic map. Cytogenetic deals with study the structure of chromosomes along with their functionality. It is based on karyotype of the species' genome. Karyotyping is the analysis of metaphase chromosomes. First, chromosomes are defined according to their sizes and banding pattern (karyotyping).

In these maps, all chromosomal assignments are made by cytogenetic analysis or by linkage to a previously assigned locus. There are different ways of generating chromosome maps with different levels of resolution.

In a recent method named in situ hybridization which is now feasible by availability of a locus-specific DNA probe, one can directly visualize the location of the corresponding sequence within a particular chromosomal band. Since no assumptions exist in this method and is based on no correlations, it is considered as the most direct method of chromosome mapping that exist. Nevertheless, the resolution is not as high compared to other methods [34].

4.1.3 Physical Maps

All physical maps are obtained by direct analysis of DNA. Physical distances between loci are measured in bp (base pairs), kbp (kilo base pairs) and mbp (mega base pairs). Physical maps are divided into two groups of short range and long range. In short range context, distances are around 30kbp which usually is average size of genes in genomes. Cloned regions of this size can be easily mapped with high resolution using appropriate restriction enzymes. With advances in sequencing technology, sequencing regions of interest in this size is becoming more common [34].

Long range physical mapping can be done over mega base size of sequences which are obtained by using rare-cut restriction enzymes along with various methods of electrophoresis referred to as pulsed field gel electrophoresis (PFGE). This method allows for sizing DNA sequences of length 6 mb or more. The details on how restriction enzymes work and how cut sites are generated are discussed in the following chapter [34].

Physical maps consisting of overlapping clones will potentially cover the whole chromosome and short range restriction maps of high resolution will be merged to-

gether along each chromosomal length will provide the highest resolution possible in order to have whole chromosome DNA sequence [34].

Different maps that discussed above are somehow connected together. For example they all provide the same information regarding chromosomal assignment. However, the relative distances within each map could be different. An accurate description of actual length of DNA that separates loci from each other can be provided only by physical maps. It does not mean that the measurements of two other maps are incorrect; they provide a version equivalent to that in physical map that has been modulated and adapted for a specific purpose. In chromosome maps, cytogenetic distances are modulated by relative packing of the DNA into different chromosomal regions. In linkage maps, distances are modulated by variable inclination of different regions of DNA to take part in recombination events.

4.2 Genetic Markers

In the previous section we discussed different types of genetic maps and briefly explained how they are generated. In this section we investigate genetic markers, a critical part of genetic maps.

To illustrate, if we consider genetic maps as road maps we use in our road trips, genetic markers are land marks or places of interest on those maps. Genetic markers are also called *marker* for short. The term marker is used very broadly to describe any observable variation in the genome that could be resulted from mutation, alteration or recombination between two parent genes. DNA markers are used for generating genetic maps especially when there are predictable mutations that occur by recombination during meiosis and can lead to, in long term, to high variability in that species . Those markers can be within genes that code for a known phenotype too such as eye color or even a kind of disease.

Since markers are resulted from mutations and gene variation, they can be used to study relationship between inherited diseases and finding its correspondent genetic cause. Or in another application, they can be used to enhance good phenotypes in off springs. For example to enhance high yield characteristic of wheat by finding its relevant gene that is done by means of genetic markers. Genetic genealogy and determining genetic distance between individuals of a population is another purpose of using markers, in which Y chromosomal or mitochondrial DNA are tested for maternal and paternal lineages and other autosomal markers are used for all ancestry. Autosomes are all non-sex chromosomes. In this literature sex chromosomes are called allosome and non-sex chromosomes are called autosome.

Markers can be categorized into different groups based on their objective and the way they are produced. In the following we investigate some of the most common of them and will explain the last one, RAD markers, in detail in the following chapter.

4.2.1 Restriction Fragment Length Polymorphism (RFLP)

This is in fact the name of a technique which is based on variations in homologous DNA molecules [6]. It investigates the differences between samples of homologous DNA that are obtained from different cut sites of exploited restriction enzyme. These samples which are called restriction fragments then are separated according to their lengths by gel electrophoresis method. By rising inexpensive high throughput sequencing methods, this method is obsolete but it was the first DNA profiling technique that was employed in many applications, especially genetic mapping. Markers captured by this technique are also called RFLP markers.

In this method, as explained above, DNA is fragmented by means of restriction enzymes and not the common method of shot gun sequencing. By doing some known procedures which is out of the scope of this context, length of fragments related to

a labeled segment of the DNA under study is determined. An RFLP occurs when the length of a detected fragment varies between individuals. In fact in this type of marker the difference in length of fragments is a sign of mutation in the genome and therefore is called an allele and is used in further genetic analysis.

But how mutation in genes results to different lengths of fragments, directly relates to how these markers are generated and restriction enzyme cut sites. Suppose that a mutation occurs in the region which restriction enzyme had to cut which means that site will not be cut and the length of fragment between two consecutive cut sites will be more than the case there was not any mutation there. This is how RFLPs show mutations in the genome.

4.2.2 Variable Number Tandem Repeat (VNTR)

For this part it needs to define a tandem repeat first. Tandem repeat occurs when a sequence of nucleotides is repeated and repeats are adjacent to each other. For example ATCGTTATCGTTATCGTT is a tandem repeat which consists of 3 repetition of ATCGTT together. VNTR is a location in a genome where tandem repeat occurs. They are on chromosomes and the variation between individuals happens in the length of those tandem repeats.

By recombination or replication, number of repeats changes from parents to offspring which results to different alleles. VNTRs can be analyzed by RFLP technique as the strand attached to tandem repeats are non-repetitive regions that can be found using restriction enzymes. By recent improvements in high throughput sequencing this type of marker is highly used for forensic crime investigation purposes.

It is mainly because of its especial characteristic that when tested in a group of independent VNTR markers, the likelihood of two unrelated individual having the same VNTR pattern is almost impossible. After fishing VNTRs and determining

their length by gel electrophoresis methods, they form a pattern which is unique for each individual. This marker can also be used in studying breeding patterns in populations of animals.

4.2.3 Microsatellites

Microsatellites [25] involves tandem repeats too as they are repeating sequences of 2 to 6 base pairs. Alternative names for this type of marker are STR (Short Tandem Repeats) or SSR (Simple Sequence Repeats). It is actually a subset of VNTRs. As mentioned, the repeats are often simple consisting of two, three or four nucleotides at each block which are called di- tri- and tetra-nucleotide repeat respectively. One common example of such marker is repeat blocks of (CA)n where n is number of blocks and varies between alleles. It is the only marker that provides clues about which alleles are more closely related [16].

The more the number of repeat block, the more alleles will be resulted due to higher potential of slipped strand mispairing known as slippage which is a kind of mutation occurs during DNA replication. This issue also arises during amplification of the microsatellites. For example during if slippage occurs during PCR, microsatellites of incorrect length could be generated.

Since this type of markers is kind of VNTR marker, its application and usage is almost the same as VNTRs in forensics, genetic fingerprinting , population genetic studies and recombination mapping.

4.2.4 Single Nucleotide Polymorphism (SNP)

This marker is actually a variation in one nucleotide in a DNA sequence. For example there is one SNP in ATGGC and ATCGC that happens in the third nucleotide position. In this type of marker different alleles are determined by the nucleotide that is different between sequences.

In above example there are two alleles of ATGGC and ATCGC. For SNP marker there is usually two different alleles. They can happen between members of a biological species or chromosome pairs in human, etc. They usually occur in non-coding parts of DNA more than coding parts or more generally, in the parts that natural selection fixates the allele of the most favorable genetic adaptation [4].

If SNPs happen in coding part of DNA, the resulted amino acid would not necessarily be change because of degeneracy of the genetic code. Based on this fact, SNPs of coding regions are divided into two groups of synonymous and non-synonymous, in which the resulted amino acid would be the same in the former and different in the latter. Even those that are not in the coding region could cause different gene splicing, transcription factor binding and messenger RNA degradation. These all together can develop different gene functionality and different amino acid production.

The main use of SNP markers is in genome wide association studies (GWAS) which is examination of many genetic variants in different individuals to see if any variant is associated with a trait. It is used as a high resolution marker in gene mapping in GWAS. The study of SNPs can help better understand how body reacts to specific drugs as well as Mendelian diseases. For complex diseases, they do not act individually but they work in coordination with other SNPs to initialize a state condition as in Osteoporosis [35] A wide range of human diseases such as cancer, infectious diseases, Sickle-cell anemia and autoimmune might be resulted from SNPs in human genome [23][9][17]

Based on that fact, they could be target for some drugs to cure diseases and therefore are used in drug therapy [14]. Even SNPs without impact on a specific phenotype are still good targets for marker studies because of their quantity and stable inheritance over generations. As of now almost two hundred million SNPs have been identified on human Genome.

There are several ways to identify and detect SNPs which is discussed in the context of SNP genotyping. By becoming the cost of sequencing less expensive year by year, the use of this marker has been increased compared to methods based on micro array data which were originally introduced to overcome the cost of SNP identification. DART which is discussed in the following is one of those methods that introduced in those days that sequencing was considered as an expensive laborious task.

4.2.5 Diversity Arrays Technology (DArT)

This is actually a method for DNA polymorphism analysis using microarrays, which can provide comprehensive genome coverage even for species that there is no information regarding their DNA sequence [24].

To make a diversity panel, large number of DNA fragments that are prepared from representations which are derived from a selected group of genotypes are cloned and individually arrayed. First, DNA is cut with a chosen restriction enzyme to reduce its complexity. Then fragments from each representation are amplified by PCR method. Individual DNA fragments are isolated by cloning. Those inserts are then amplified and arrayed on a solid support. This is how diversity panels are made [24].

DArT is essentially assaying for the presence or amount of a specific DNA fragment in a representation that is resulted from the total genomic DNA of an organism or a population.[24]. In [37] it has shown that this method could be applied to genetic mapping and diversity analysis of barley. They also have validated the Mendelian behavior of DArT markers by constructing a genetic map for a cross between two cultivars. They have also provided a comparison between RFLP based framework map and DArT based map in which they confirm the quality of the DArT was equivalent, if not superior, to that of the framework map.

Although today due to improvement in sequencing techniques, the throughput of SNP analysis has increased and the cost has decreased correspondingly, yet for complex polyploidy genomes such as most of plants, SNP analysis is difficult and DArT could be the better alternative. SNP assay is easier to be used in diploid organisms such as humans and a limited number of model organisms [37].

5. RESTRICTION SITE ASSOCIATED DNA (RAD) MARKERS AND RADSeq ANALYSIS

This is the type of marker that the main work of this thesis is based on and therefore it is explored more than other markers with investigating it from many aspects. First section is dedicated to review the method of generating this type of marker. Then we investigate features as well as potential issues of combining RAD method in generating markers and Next Generation Sequencing (NGS), which is called RADSeq. The last section addresses the most common tool for RADSeq analysis as well as designed post processing steps in order to retrieve RADSeq data.

5.1 Restriction Site Associated DNA (RAD) Markers

This method was first introduced in [30] based on micro array data. Similar to RFLP method which mentioned earlier, it uses restriction enzyme to reduce the complexity of DNA and only considers sequences flanking the restriction enzyme. Thus to perform an unbiased comparison, it should be compared to other restriction enzyme based marker discovery approaches. As described in [5], [36] and [38], SNPs are the most abundant type of genetic marker and their high density makes them suitable for studying inheritance of genotypes and generating genetic maps.

The major goal of RAD technique described in [30] is to make polymorphism identification in various genomes fast and cost effective, however the drawback of microarray based RAD techniques is that accomplishing a reliable and widely applicable kind of such technique can only assay a fraction of polymorphisms. Moreover, that technique suffers from making specific species arrays for each organism and for every comparison new array hybridization should be performed [3]. To overcome this limitation [3] enhanced RAD genotyping platform by utilizing next-generation

sequencing which makes the use of extensive polymorphism data feasible.

RAD tags are the output fragments obtained from RAD method. These sequenced RAD tags have many features for genetic mapping purposes. First, as in all restriction enzyme based methods, it reduces the representation of the genome and allows for over sequencing specific sits close to restriction enzyme and detection of SNPs. Second, an appropriate number of markers for a specific application can be selected by a good choice of restriction enzyme. Moreover, the number of markers can be always increased by adding additional enzymes to the experiment. Third, in this method genotyping bulk segregant analysis, pooled population type of experiments and multiplexed genotyping of individuals, all could be accomplished [3].

Polymorphism can occur either due to a disruption in a cut-site which leads to differential isolation or due to presence of SNPs within the sequence of tags that are present in individuals [3]. To increase the resolution of genotype mapping, it is better to have more number of SNPs. This can be achieved by a good choice of restriction enzyme for the genome under study which is able to cut in more places. The feature of adding barcodes in this technique also is a good advantage which allows parallel processing of samples by enabling all samples to be combined in all steps of RAD library preparation and sequencing. Multiplexed sequencing of a large population also enables rapid mapping of multiple traits [3]

In summary, the major advantages of combining next generation sequencing and RAD markers is increase in number of generated markers, decrease in cost and effort as well as increase in speed of analysis. By this rate of decreasing cost and increasing produced data in sequencing technology, sequencing whole genome of most of the species might be viable in the future, but produced SNPs in such abundance would be wasteful. Therefore, even with the improvement of sequencing technology, RAD markers would be still useful as they provide targeted SNPs in the sequences flanking

the end of restriction enzyme [3].

5.2 RADSeq

In [12], authors have introduced a new terminology, RADSeq, for combining two simple molecular biology techniques with Illumina sequencing which is RAD method and the use of molecular identifiers (MID) to associate sequence reads to particular individuals. After shearing the DNA with an appropriate restriction enzyme, for resulted flanking sequences to be sequenced on an Illumina machine, RADSeq uses modified Illumina adapters that enables binding to the surface such as P1 and then a MID which will uniquely identify each individual and attach them to the end of the sequence. RADSeq can be applied for population study purposes even to genomes that have no or limited sequence data. Its difference compared to other restriction enzyme based marker detecting methods such as RFLP is that it can identify, verify and score markers simultaneously, whereas in other similar methods, this requires extensive development process [12].

One of the important facts regarding RADSeq is that it is also able to detect presence-absence polymorphisms [12]. This is an important type of marker on which one of the main researches of this thesis is relied. Presence-absence marker is a marker that is present in one set of individuals but absent in another which indicates a variation in the restriction enzyme cut site that causes no cut to be occurred in one of the samples [12].

In RADSeq method, to find other types of markers such as SNPs and insertions or deletions (indels) of DNA, if reference genome exits, we can easily map RAD tags to the reference genome using BWA [28] or other mapping software and identify SNPs, insertions and deletions. Those mapping applications automatically correct for the low level of sequencing error in the reads. In case of having no reference genome,

RAD tags can be analyzed de novo. In that case contigs are made and by comparing candidate contigs we can find SNP and indels between alleles at the same locus [12].

As mentioned earlier, RADSeq is able to produce two kinds of markers: presence-absence marker which is a result of polymorphism in the restriction enzyme cut site and also SNP or indel markers. Furthermore, paired-end sequencing can also be accomplished from RADSeq libraries. Since fragments are randomly sheared, second pairs of each RAD tag can begin in different positions downstream of the restriction site. These, after assembly, can produce extended contigs of 200-300 base pairs. Produced contigs are useful for development of PCR-based assays as well as identification of SNPs and indels [12].

Now in this part we start to compare RADSeq with other similar methods. First, there is not any variant discovery and assay design step and also no hybridization optimization issues. Moreover, full sequences of the RAD tag or its paired contig in case of paired end sequencing can be analyzed for SNPs and indels. Another advantage is that obtaining additional sets of markers for increasing density just requires using different restriction enzymes, rather than extensive design process. Initial analysis of RAD tag sequences is also simpler compared to other methods [12].

To improve accuracy of RADSeq mapping for identification of the loci attached to traits of interest, like other genetic association experiments, three major factors have impacts: number of independent markers assayed, their levels of variability and the number of crossovers that have occurred in the mapping population. Therefore by adding restriction enzymes to get more markers and increasing the number of individual cross progeny or population accuracy of RADSeq mapping could be increased [12].

RADSeq is a very versatile method and is expected to work with any restriction

enzyme. Using new Illuminas high throughput sequencing machines such as HiSeq 2000, mentioned earlier, it can be utilized for assaying hundreds of individuals at a substantial depth. This is useful for identifying markers for large scale population genotyping purposes, creating complete linkage maps and generating dense linkage maps for scaffolding of newly sequenced genome [12]. It can be exploited for various projects. It is suitable for fine scale linkage mapping [2], population genetics, phylogenetic and phylogeography, genome scaffolding and generating large SNP data sets for many species. Moreover, RADSeq produces more robust markers compared to other related methods and therefore is more appropriate for de novo analyses of wild populations for which there is lack of information about the genome and consequently imputation of missing data is very difficult [13].

5.2.1 Biases and Noises

In this subsection we briefly investigate potential bias that might be in the process RADSeq which can affect the final results and should be considered in order to get accurate assay of data for further analysis.

Sequencing-by-synthesis process which is the foundation technique of Illumina sequencing machines, introduces noise which make the path from raw reads to biological information very far from simple [13]. Although library preparation methods can eliminate those noises in some sense [1], but each experiment requires its own library preparation which can produce novel sources of noise. Filtering high quality markers and separate them from noise is not an easy task and since RADSeq is mostly used for populations with no reference genome, validating the performance of automatic common tools for RADSeq analysis is a difficult task [13]. In the following we investigate two of those sources of biases.

As shown in [13], there is a high correlation of RAD locus read depth of coverage

and the logarithm of the length of restriction fragment. This bias is caused by shearing step during library preparation where sonicators do the job with varying effieciency.

Another anticipated source of bias is PCR GC bias as there is a PCR step during RADSeq library preparation. GC content could always be a source of noise in most of bioinformatics applications, as G pairs with C by three hydrogen bonds and therefore is harder to be broken apart compared to the bond between A and T which are paired with two hydrogen bonds [11]. Moreover, PCR GC bias is a known source of noise in sequencing-by-synthesis data [13]. The interesting results regarding the impact of this bias on read depth in RADSeq analysis is that the RAD loci with high GC content are sequenced at a higher depth compared to those with lower GC content as number of PCR cycles increases. However, at 14 to 16 PCR cycles, high GC content RAD loci are under-sequenced compared to low GC content RAD loci [13].

RAD markers and more specifically RADSeq markers which is the combination of RAD method and sequencing by synthesis used in Illumina sequencers are known to be a powerful tool for genotyping and SNP identification. They can develop markers in such extent that can be utilized in high resolution mappings. Being versatile in many applications as well as flexibility in increasing number of markers just by adding different restriction enzymes, are its other important advantages over other related methods. However, as discussed earlier, the read depth of RAD loci are highly variable which could be the result of different sources of noise and bias [13]. We just mentioned two of those sources: Different lengths of restriction fragments and PCR GC content bias. However, the effect of those biases could be diminished by filtering reads and picking reliable ones in order to increase the confidence of downstream genotyping analysis.

5.3 Retrieving RADSeq Data Using Stacks, a Software for RADSeq Analysis

In this section we describe our pipeline for producing RAD markers and their correspondent counts. We first explain algorithms used in Stacks [8], a useful tool for RADSeq analysis and genotyping, as the main step of the pipeline and then describe post processing steps for generating marker counts.

Stacks is an open source software system for RADSeq analysis which uses sequenced fragments of reduced representation data obtained by RAD method. It can be used with or without reference genome, which in latter it uses Velvet [41] for its de novo analysis. Marker analysis of a cross experiment between two samples of a population with their progenies as well as analyzing individuals of a population can be done with Stacks. It can generate markers for linkage maps, examination of population phylogeography and help in reference genome assembly in a computationally efficient way [8].

This software is written in C++ and Perl and its core algorithms are parallelized by utilizing OpenMP libraries. It has a web interface to see the results of how RAD tags are arranged SNPs for each one and final genotyping. It stores and retrieves data in a MySQL database.

The pipeline of Stacks is as follows which is performed for all the samples before catalog generation begins. For each sample it first groups read fragments together based on allowable user set mismatches and make stacks of them. This is why they name the software, Stacks [8]. Usually, allowable mismatch for this part is set to zero in order to capture all perfectly similar read fragments flanking from the same restriction enzyme cut site, although sometimes read fragments from different cut sites can be grouped together if they are exactly the same. Therefore at this step, different stacks are constructed. The minimum number of read fragments to comprise

a stack is also set by user [8].

At the second step, it uses kmer search algorithm to merge stacks together in order to generate loci. Options regarding distance of stacks to be merged are also set by user. This step is done in an iterative fashion on number of mismatches. It first finds those stacks that are one, then two and then three nucleotides apart from each other consequently and then merges them. At this step it has generated loci and is time to infer alleles and haplotypes at each locus. For this, Stacks examines each putative locus one nucleotide position at a time and uses the maximum likelihood framework in [18]. Haplotypes are comprised of SNPs. Two SNPs at two different positions can introduce haplotypes of length two at that locus. Therefore Stacks examines the matrix of reads column wise to detect SNPs and then it detects it row wise to identify haplotypes [8].

The final step is to construct catalog from previously constructed loci for each sample. This step is for merging different samples together based on user defined options and examining the behavior of polymorphisms and haplotypes between the samples. One of the samples initializes the catalog, i.e. the female parent in a parent progeny experiment. Then loci of each individual are matched against the catalog containing those samples are already there, using kmer search algorithm and merges a sample to an entry of catalog only when their SNPs are matched, otherwise merging is failed [8].

In the catalog RAD tags are made as entries of the catalog and each is assigned a Catalog ID. At each RAD tag, it is possible to see which samples are present, what are their alleles and what are present haplotypes made with those alleles across all samples present at that RAD tag. Moreover, the counts and read depth of each allele is also reported. This is how we get the counts of each marker or its depth and use them in further analysis. Stacks can call mappable markers based on alleles

in progenies and the parents. It is also capable of doing automated corrections for previously called markers [8].

Retrieving Data: After Stacks does the analysis, we can use its output files to do further post processing steps. We need to get the counts of each allele of each sample for each RAD tag. We accomplished this task by writing codes in Perl language which gets output files of Stacks as an input and by processing them it can generate all the files with mentioned details for counts.

The final generated file in the post processing step has following columns where each row shows each entry of the table. It has the Catalog ID as the first column along with its haplotype. If a catalog ID has more than one haplotype, they are considered as different entries and come in next rows although having the same catalog ID. The catalog ID and its haplotype are separated by an underscore. For instance, we have 116_AG and 116_TG as two different entries of our file which show two alleles of ID 116. We have two parents and all other progenies as next columns in which we put its count of that specific catalog ID_haplotype.

In other column of the file we have put the adaptively changed consensus sequence for that entry. To make that, we use the consensus made by Stacks for that RAD tag and then we change correspondent bases of the consensus based on the position of SNPs in the reported haplotype. Number of present parents as well as present progenies for that RAD tag is also reported as other columns. As mentioned earlier, it is a detailed file providing all kinds of counts for each RAD tag and is the result of post processing parts of our pipeline for RADSeq analysis.

For our problem, we are interested in PAV markers in two parents of a cross experiment in one of the populations of wheat. Therefore, we extract those two columns of our generated file which contains the counts of each RAD tag in parents samples. This gives us many two dimensional points each represents a RAD tag. It

means that we can plot them in a two dimensional graph and do the further analysis which is described in detail, while revisiting the problem, in the next chapter.

6. METHODS ON CALLING TRUE STATES OF PRESENCE/ABSENCE RADSeq MARKERS

In this chapter, we focus on the the main problem of the research, regarding one of RADSeq marker types named presence absence markers and its solution. The first section of this chapter defines the problem in detail, while following sections discuss the solution to the problem in two different contexts.

6.1 Problem Statement

In our research we focus on specific type of RAD markers which is presence-absence markers and is also known by presence-absence variation (PAV). As described previously, this occurs when a RAD tag is present in one of the samples and not present in the other one or in terms of marker counts, it has a zero count in one while non-zero count in the other one. PAVs actually show variation in the cut site which caused the restriction enzyme not to cut that region. One of important considerations here is that when a nonzero count is seen in a sample, is it really an absent marker or it could be a miss. This fact can be inferred by investigating the count of that specific RAD tag in the other sample under study.

As explained in previous chapter, retrieving markers and their counts after samples being sequenced out of machine is obtained by Stacks [8] and some post processing steps which is our pipeline for RADSeq analysis.

The goal of this chapter is to investigate PAV markers between two parent samples in a cross of a specific wheat population. As described earlier, presence-absence markers are those that are present in one of the sample and absent in the other. They could be informative in the sense that they show variation in the cut site of restriction enzyme as it cuts for one of the samples and does not do the same for the

other one.

Because of many sources of bias and noise in the experiment it is not straightforward to call a zero count necessarily as absence and similarly a non zero one, necessarily as presence. They were in the same library preparation procedure, in the same sequencing experiment and even in the same lane of sequencer machine. These all could be considered as sources of noise which can affect both samples. Therefore counts of those two samples can not be considered as independent. However, there should be a model to interpret this behavior and be used in order to perform further confidence measurement.

Putting counts of two samples in a pair, they can be considered as points in a two dimensional space. For instance, $(0, 5)$ shows zero count for sample 1 and count 5 for sample 2. Independence between the counts can be interpreted in this way that those counts affect each other. For example the zero count in $(0, 5)$ might not be true absence or even 5 might not be true present. This is the result of dependence which itself is caused by mentioned sources of bias.

Our goal is to ultimately find a threshold on the count of each sample so that for the counts of more than that threshold, we reach a required confidence level of calling true absence for zero count sample and and true presence for non-zero count one. By intuition, we have more confidence in calling true absent and true presence in $(0, 20)$ comparing to $(0, 5)$. If counts were totally independent, we could even call counts in $(0, 5)$ as true absence and true presence but that dependence causes counts to affect each other and hence requires more investigation to find confidence level in calling true states of the counts. A sample count data for two parent samples is shown in Fig. 6.1.

The importance of making true absence or missed is that based on [33], we can fill out missed calls by either imputation or redoing the experiment by increasing the

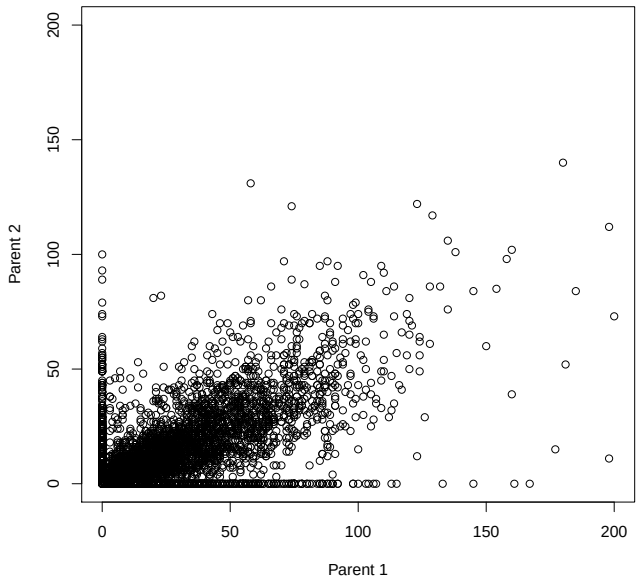

Figure 6.1: Counts of all RAD markers in Parent 1 vs. Parent 2 for a wheat breeding study

depth coverage. Thus, it would be better if we have information about zero count RAD tags whether they are absent or missed with a high level of confidence.

6.2 Solution 1: Estimating the Joint Distribution

Since the counts of two samples are not independent, if the correlation between them can be estimated, it can lead us to find the threshold for confidence level of true absence or true presence for a given count point. In this context, we call that 'confidence threshold'. One important point here is that we do not know a point on one of the axis which has one of its components as zero, has come form which source. Are the counts of that point happened independently or each of them has occurred by the impact of the other one.

To find out the answer to above problem, we need to estimate the impact of

each sample on the other one by means of the data points we have. It is worth to mention again in what type of data points we are interested. Since we need to make true absence-presence call, we only care about the points on both axes for which one of the components is zero. Other data points on the first quadrant is just used for analysing the correlation and estimating the model.

One reasonable solution to this problem could be fitting a mixture model to the data which can capture the impact of all potential existent distribution in the data. This solution is discussed in detail in the following subsection, Fitting a Mixture Model to Data. Another solution is to measure the impact of the count of each sample on the other one by estimating the joint distribution between two samples.

6.2.1 General Algorithm

Because the status of data points on either of the axis is not certain that whether they belong to the joint distribution or occurred independently, the impact of other sample on the sample under study should be measured as the indicator of in what probability that data point occurred independently or from the joint distribution.

In this context, the confidence of calling a point on each axis as true presence-absence is defined as the complement of the impact of zero count sample on the count of non-zero one for any value higher than that point. It means the smaller the impact gets, the more confident we are in calling true presence-absence for that point and its higher value points and visa versa.

On the X-axis, the second sample is always zero and on the Y-axis, the first one. Hence, for a specific point x^* on the X-axis, the impact of zero count sample on the non-zero values higher than that can be defined as the conditional probability of X being more than or equal to x^*, given $y = 0$ which is: $Pr(X > x^*|y = 0)$. Similarly for the points on the Y-axis which that changes to: $Pr(Y > y^*|x = 0)$. As

we defined, confidence is complement of that, Hence:

confidence threshold for the point $X = x^*$ on the X-axis: $1 - Pr(X > x^*|y = 0)$

confidence threshold for the point $Y = y^*$ on the Y-axis: $1 - Pr(Y > y^*|x = 0)$

By playing with those expressions we obtain:

$$\begin{aligned} &\text{confidence threshold of } (x^*, 0): \ 1 - (1 - Pr(X \leq x^*|y = 0)) = F(x^*|y = 0) \\ &\text{confidence threshold of } (0, y^*): \ 1 - (1 - Pr(Y \leq y^*|x = 0)) = F(y^*|x = 0) \end{aligned} \tag{6.1}$$

where F is the CDF of conditional distribution function.

In fact, that conditional distribution resulting from the joint distribution is the key element toward measuring the confidence threshold. It has the correlation essence in it and changes the value of the confidence threshold. That conditional distribution indicates the impact of one of the sample having zero count on the non-zero count sample. This is why we are less confident about $(0, 5)$ to have true presence-absence compared to $(0, 20)$, unless both counts are independent which is not the case for our data. This is the algorithm by which the confidence threshold of calling true presence-absence for any given point and any points higher than that is obtained as a number between zero and one or as a percent. The only requirement is estimating the joint distribution.

6.2.2 Estimating Joint Distribution

To achieve that goal, first we need to find a distribution that better fits the data. In order to make counts of two samples comparable, we need to cancel out library preparation bias for each one. If we have twice total number of reads for parent 1 compared to total number of reads for parent 2, this affects the final count of RAD

tags for each sample. Thus, the number of counts for both samples are normalized according to proportion of reads produced in library preparation step. After this normalization, the data become continuous and therefore we can apply a continuous distribution to data. Since they are defined in first quadrant of two dimensional space, a lower truncated bivariate normal distribution would be a good candidate for to be fit to bivariate data. The lower truncated bivariate normal distribution [26], with parameters $\mu_1, \mu_2, \sigma_1, \sigma_2$ and ρ is given by:

$$f(x', y') = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}L(h,k)} \exp\left[-\frac{1}{2(1-\rho^2)} \left\{ \frac{(x'-\mu_1)^2}{\sigma_1^2} - \frac{2\rho(x'-\mu_1)(y'-\mu_2)}{\sigma_1\sigma_2} + \frac{(y'-\mu_2)^2}{\sigma_2^2} \right\} \right], \quad h \leq x' \leq \infty,\ k \leq y' \leq \infty \tag{6.2}$$

where $L(h,k)$ is the total probability for the truncated distribution.

If we are able to estimate the joint distribution, it is informative of all parameters we require for confidence measurement, especially the correlation between the counts. To estimate the joint distribution we choose Maximum Likelihood Estimation method which requires data points to estimate the distribution. To simplify (6.2) to be used in ML estimator, change of variables should be done and finally we have [26]:

$$f(x, y; \boldsymbol{\theta}) = c^{-1} \exp\{-\frac{1}{2}(h_1 x^2 + 2h_3 xy + h_2 y^2 + 2\eta_1 x + 2\eta_2 y)\}, \quad 0 \leq x \leq \infty, 0 \leq y \leq \infty$$
$$c = \int_0^\infty \int_0^\infty \exp\{-\frac{1}{2}(h_1 x^2 + 2h_3 xy + h_2 y^2 + 2\eta_1 x + 2\eta_2 y)\} \tag{6.3}$$

where $\boldsymbol{\theta}$ is the vector of parameters and,

$$x = x' - h, \quad y = y' - k, \quad \xi_1 = \frac{h - \mu_1}{\sigma_1}, \quad \xi_2 = \frac{k - \mu_2}{\sigma_2}, \quad h_1 = \frac{1}{\sigma_1^2(1-\rho^2)},$$
$$h_2 = \frac{1}{\sigma_2^2(1-\rho^2)}, \quad h_3 = \frac{-\rho}{\sigma_1\sigma_2(1-\rho^2)}, \quad \eta_1 = \frac{\xi_1 - \rho\xi_2}{\sigma_1(1-\rho^2)}, \quad \eta_2 = \frac{\xi_2 - \rho\xi_1}{\sigma_2(1-\rho^2)}$$

Therefore, the ML estimator of the parameters using n data points is going to be:

$$(\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2, \hat{\rho}) = \operatorname{argmax}\Big(\sum_{i=1}^{n} \log f(y_{i1}, y_{i2}; \boldsymbol{\theta})\Big) \tag{6.4}$$

where f is given in (6.3).

For the data points to be used in ML estimator, as mentioned previously, the state of the points on the axis is not determined whether they are from the joint distribution or occurred independently, especially when they get farther from origin. Therefore, we can not count all of them in estimating the joint distribution. It is not straight forward to find what proportion of them to pick either. In fact, in that case we knew the answer to our problem already. This is why considering a mixture model that examines all the cases and finds the best fitted one based on some criteria, seems an optimal solution.

Based on above explanation, in our algorithm we exclude all the points on the both axes from data points participating in MLE. This makes the algorithm sub-optimal at the first glance, but when we ran it in simulation for a true model, it could estimate the parameters precisely and in a reasonable confidence interval. For generating random numbers from the true truncated bivariate normal model, we used 'tmvtnorm' package in R [39]. The average estimates of parameters as well as standard deviation of estimation over the square root of the number of simulations which is 1000 times are given in Table. 6.1. Those information can be used in

calculating different confidence intervals.

Interior point algorithm [7] is used for optimizing the non-linear constraint optimization problem in (6.4), using one of built-in functions in MATLAB®. The initial values are also set to empirical values obtained from the data. By applying this estimator to the real data, we estimate the joint distribution and proceed with the algorithm explained earlier. Fig. 6.2 shows how this estimator can cover count data points for two parent samples. The data points in Fig. 6.2 are normalized in order to remove library preparation bias in total number of reads.

	μ_1	μ_2	σ_1	σ_2	ρ
True Value	20	10	45	32	0.6
Avg. Est.	19.9721	9.9852	45.0145	32.0023	0.6002
$\frac{\sigma}{\sqrt{n}}$	0.0245	0.0187	0.0114	0.0088	0.0002

Table 6.1: Results of 1000 times simulation for estimating parameters of truncated bivariate normal distribution

As presented in Table. 6.1, the confidence interval of the estimator for true model is small which verifies performance of the estimator.

As described in [19], the conditional distribution of a multivariate truncated normal distribution is a truncated bivariate normal itself. Based on definition of conditional distribution and (6.3), the conditional distribution of bivariate truncated normal is given by:

$$f(x|Y=y^*)=\frac{f(x,y)}{f(y^*)}=\frac{\exp\{-\frac{1}{2}(h_1x^2+2h_3xy^*+h_2y^{*2}+2\eta_1x+2\eta_2y^*)\}}{\int_0^\infty \exp\{-\frac{1}{2}(h_1x^2+2h_3xy^*+h_2y^{*2}+2\eta_1x+2\eta_2y^*)\}dx} \tag{6.5}$$

Similarly, it can be re-written for $f(y|X=x^*)$.

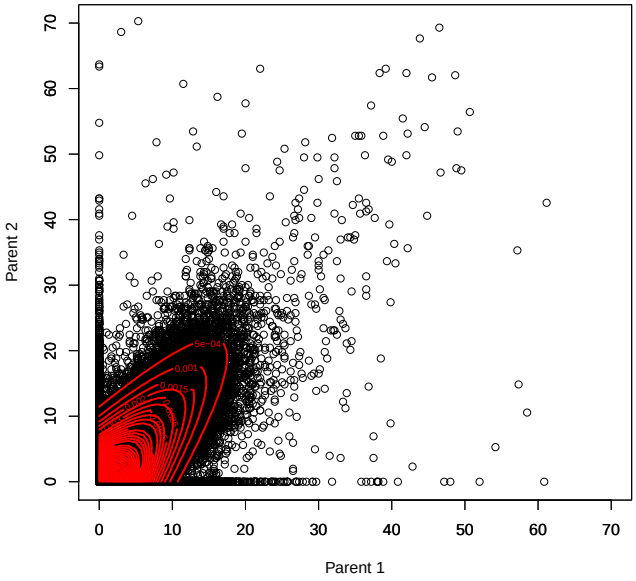

Figure 6.2: Contours of estimated joint distribution on normalized count data

6.2.3 Results and Discussion

In our problem, y^* and x^* in (6.5) should be set to zero for X-axis and Y-axis respectively. Now by integrating the conditional distribution in (6.5), we can obtain conditional CDF and therefore the confidence threshold. This means the confidence threshold in (6.1) is changed to:

$$\text{confidence threshold of } (x^*, 0): \quad \int_0^{x^*} f(x|y=0)$$
$$\text{confidence threshold of } (0, y^*): \quad \int_0^{y^*} f(y|x=0)$$

where each conditional distribution is obtained from (6.5). The final results of confidence thresholds for different values on X and Y axes are brought in Tables 6.2 and 6.3.

x^*	1	2	3	4	5	6	7	8	9	10
CT (%)	21.27	40.65	57.26	70.66	80.83	88.10	92.99	96.09	97.93	98.96

Table 6.2: Confidence threshold for different values on X-Axis

y^*	1	2	3	4	5	6	7	8	9	10
CT (%)	23.17	42.53	58.17	70.39	79.63	86.38	91.16	94.43	96.56	97.98

Table 6.3: Confidence threshold for different values on Y-Axis

As it is presented, by getting farther from the origin, the confidence threshold on calling true presence-absence is raised, which is reasonable as the impact of zero count sample becomes weaker on the other sample. The correlation between two samples is estimated by the joint distribution and the impact of samples on each other on each axis which is resulted from that correlation, is exhibited by the conditional distribution. Finally, that conditional distribution is our main tool to measure the level of confidence in calling true presence-absence RAD biomarkers.

6.3 Solution 2: Fitting a Mixture Model to Data

Another solution for this problem is to model it with mixture of distributions and measure the likelihood of true absence or true presence. According to Fig. 6.1, the model constitutes of three different distributions, two univariate distributions on each axis and one bivariate distribution for the points on first quadrant.

If a point on one of the axis comes from the correspondent univariate distributions on that axis, it means the zero count is always zero with probably one. Hence, we have true absence for those points. On the other hand those points that comes from the bivariate one, we can not make any calls. Another reason to use a mixture model is that, it is not certainly known that each point on axis has come from which distribution, the univariate one on the axis or from the bivariate distribution. We

need to estimate parameters of proposed mixture model and then measure likelihoods based on that.

6.3.1 General Idea of Mixture Modelling

the mixture model can be defined as follows:

$$f = \sum_{i=1}^{3} p_i f_i \tag{6.6}$$

where, $\sum_{i=1}^{3} p_i = 1$ and

- f_1: univariate distribution representing presence-absence markers for parent 1.
- f_2: univariate distribution representing presence-absence markers for parent 2.
- f_3: bivariate distribution representing always present markers for both samples.

One important point that should be considered here is that the mixture model, in general, should be a distribution itself. This means that first, $\int f(\mathbf{w})d\mathbf{w} = 1$ in the continuous case and $\sum f(\mathbf{w}) = 1$ in the discrete case and second, we should define the problem in a way that the rules of measurable spaces are satisfied. To satisfy the second condition,since we are mixing two univariate distributions and one bivariate one, our space that the probabilities are defined on should be two dimensional. In this case, the univariate probabilities of the points on both axes are also defined in two dimensional space in which one of its components is always zero. That is why when a point comes from one of those univariate distributions, its zero component means true absence. Now, by this definition, the components of the model for both

cases of discrete and continuous is given by:

$$g_1(x,y;\boldsymbol{\theta_1}) = \begin{cases} 0 & \text{if } y \neq 0 \\ f_1(x;\boldsymbol{\theta_1}) & \text{if } y = 0 \end{cases} \qquad g_2(x,y;\boldsymbol{\theta_1}) = \begin{cases} 0 & \text{if } x \neq 0 \\ f_2(y;\boldsymbol{\theta_2}) & \text{if } x = 0 \end{cases} \tag{6.7}$$

$$g_3(x,y;\boldsymbol{\theta_3}) = f_3(x,y;\boldsymbol{\theta_3})$$

where, $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$ and $\boldsymbol{\theta_3}$ are vector of parameters for distributions f_1, f_2 and f_3 respectively.

Having this model, the potential issue regarding measurable spaces is resolved and the mixture model in (6.6) is changed to:

$$g(x,y;\boldsymbol{\theta}) = \sum_{i=1}^{3} p_i g_i(x,y;\boldsymbol{\theta_i}) \tag{6.8}$$

Second criteria to make $g(x,y;\boldsymbol{\theta})$ a valid mixture distribution is to make sure that the total summation of pdf or pmf is 1. This is where the behavior of discrete and continuous cases differ.

To estimate the parameters of the model using MLE, we should be able to partition g_3 into three components, one for each axis and one for the first quadrant in order to consider its contribution to other elements of the mixture model on each axis. For instance, for a point on the X-Axis $(x_1, 0)$, it has come from g_1 with probability p_1 and has come from the correspondent partition of g_3 with the probability p_3. This is shown in the mixture model as, $p_1 g_1(x_1, 0) + p_3 g_3(x_1, 0)$. Similarly for the Y-Axis. This is how g_3 is contributed on each axis.

In the context of continuous distributions, there is no way to contribute the impact of g_3 on both axis and still satisfy $\int g(\mathbf{w}) d\mathbf{w} = 1$. This is because slicing out just X-axis and Y-axis from the two dimensional surface, $g_3(x,y)$, does not

produce three partitions for $g_3(x, y)$ which are supposed to be considered separately for each part of the mixture model including both axes and being added to g_1 and g_2. In another words, when computing the total integral of mixture model, having partitions described in the last paragraph, we need to integrate those partitions as well but they are not valid and therefore the total probability will not be 1. This fact is described in the following:

$$\iint g_3(x, y) \neq \int g_3(x, 0) + \int g_3(0, y) + \iint_{x,y\neq 0} g_3(x, y)$$

This means the second condition, which was $\int g(\mathbf{w})d\mathbf{w} = 1$ is not satisfied for continuous distributions. and therefore, it is not a valid mixture model. One might suggest to consider the contribution of g_3 by its marginal distribution on each axis, but in that case since the marginal distribution itself is a distribution with total integral equal to 1, the overall integral for the whole mixture, $\int g(\mathbf{w})d\mathbf{w}$ becomes more than 1.

On the other hand, in the context of discrete distributions, that partitioning is valid and it is valid to partition whole $g_3(x, y)$ into three parts of X-axis, Y-axis and first quadrant. This is mainly because in discrete distributions, we have probability mass function (pmf) for discrete points that create an integer grid for the space, in which the value of each point is its probability where the summation of all these points equals to one.

Corollary 1. *It is not probabilistically valid to define a mixture model as (?? for continuous distributions. All distributions in (6.8) should be discrete and therefore each f_i is actually a probability mass function.*

6.3.2 Mixture Model in the Context of Discrete Distribution

Having known that the distributions should be discrete, we can define the model in (6.8) as follows:

$$\begin{aligned}(Y_1, Y_2) \sim (f_1(\boldsymbol{\theta_1}), 0) &\text{ with probability } p_1 \\ (0, f_2(\boldsymbol{\theta_2})) &\text{ with probability } p_2 \\ \text{bivariate } f_3(\boldsymbol{\theta_3}) &\text{ with probability } p_3\end{aligned} \tag{6.9}$$

Since the data is NGS count data, it is common to use either Poisson or Negative Binomial probability mass functions. In case of Poisson, (6.9) is changed to:

$$\begin{aligned}(Y_1, Y_2) \sim (\text{Poisson}(\lambda_1), 0) &\text{ with probability } p_1 \\ (0, \text{Poisson}(\lambda_2)) &\text{ with probability } p_2 \\ \text{bivariate Poisson}(\lambda_{10}, \lambda_{20}, \lambda_{00}) &\text{ with probability } p_3\end{aligned} \tag{6.10}$$

According to [27], the bivariate Poisson with parameters $(\lambda_{10}, \lambda_{20}, \lambda_{00})$ has the following form:

$$P(Y_1 = y_1, Y_2 = y_2) = \sum_{j=0}^{\min(y_1, y_2)} \lambda_{10}^{y_1 - j} \lambda_{20}^{y_2 - j} \lambda_{00}^{j} \times \exp(-\lambda) / [(y_1 - j)!(y_2 - j)! j!] \tag{6.11}$$

where $\lambda = \lambda_{00} + \lambda_{10} + \lambda_{20}$ and $y_1, y_2 \geq 0$

By comparing (6.10) and (6.8), we infer that f_1 is Poisson(λ_1), f_2 is Poisson(λ_2) and f_3 is a bivariate Poisson$(\lambda_{10}, \lambda_{20}, \lambda_{00})$. After partitioning the bivariate Poisson given in (6.11), to contribute it to f_1 and f_2, the probability distribution of the

mixture model is given by:

$$
\begin{aligned}
&P(Y_1 = 0, Y_2 = 0) = p_1 \exp(-\lambda_1) + p_2 \exp(-\lambda_2) + p_3 \exp(-\lambda) \\
&P(Y_1 = y_1, Y_2 = 0) = [p_1 \lambda_1^{y_1} \exp(-\lambda_1) + p_3 \lambda_{10}^{y_1} \exp(-\lambda)]/y_1! \\
&P(Y_1 = 0, Y_2 = y_2) = [p_2 \lambda_2^{y_2} \exp(-\lambda_2) + p_3 \lambda_{20}^{y_2} \exp(-\lambda)]/y_2! \\
&P(Y_1 = y_1, Y_2 = y_2) = p_3 \sum_{j=0}^{\min(y_1, y_2)} \lambda_{10}^{y_1 - j} \lambda_{20}^{y_2 - j} \lambda_{00}^{j} \times \exp(-\lambda)/[(y_1 - j)!(y_2 - j)!j!]
\end{aligned}
\tag{6.12}
$$

where $\lambda = \lambda_{00} + \lambda_{10} + \lambda_{20}$ and $y_1, y_2 \neq 0$. Another important point here is that based on [27], to reduce the number of parameters by 2 and deal with less parameters, it is assumed that $\lambda_1 = \lambda_{10} + \lambda_{00}$ and $\lambda_2 = \lambda_{20} + \lambda_{00}$. For more information regarding the structure of bivariate Poisson distribution one can read [27]. We can estimate the parameters of the mixture model using Maximum Likelihood given by:

$$
\begin{aligned}
&(\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{\lambda}_{00}, \hat{\lambda}_{10}, \hat{\lambda}_{20}) = \\
&= \operatorname{argmax} \sum_{i=1}^{n} \log P(Y_{1i}, Y_{2i}\,; \lambda_{00}, \lambda_{10}, \lambda_{20}, p_1, p_2, p_3)
\end{aligned}
\tag{6.13}
$$

where P is defined in (6.12).

To solve the non-linear constrained optimization problem in (6.13), we use interior point algorithm again, similar to previous subsection. For our data, the optimization algorithm converges and can find the maximum of likelihood function for different initial values. The estimator can also be verified by simulation. Now the last step of the solution is to interpret confidence level in our model and investigate how to make the final call for each point on the axis whether its zero count is true absence and its non-zero count is true presence.

Having estimates of the mixture model's parameter, the confidence level for each

point on either of the axis is defined as follows:

$$\begin{aligned} &\text{For point } (y_1, 0) \text{ on the X-axis: } c = \frac{p_1 g_1(y_1, 0)}{p_3 g_3(y_1, 0) + p_1 g_1(y_1, 0)} \\ &\text{For point } (0, y_2) \text{ on the Y-axis: } c = \frac{p_2 g_2(0, y_2)}{p_3 g_3(0, y_2) + p_2 g_2(0, y_2)} \end{aligned} \tag{6.14}$$

where, as described in details of (6.10), f_1, f_2 are univariate Poisson distribution with parameters λ_1 and λ_2 respectively and f_3 is a bivariate Poisson distribution with parameters $(\lambda_{10}, \lambda_{20}, \lambda_{00})$. Since they are discrete distributions, the values of f_i at each point is actually the probability of that point.

Points regarding (6.14) that should be noted is that, the contribution of f_2 for points on the X-axis is zero as well as the contribution of f_1 for the points on the Y-axis. The reason is in the definition of the model in (6.10) in which the points on each axis are either coming from the univariate distribution of that axis or the bivariate distribution. Another point is that the measured confidence level is between 0 and 1 and it increases as the probability of that point coming from bivariate distribution decays, which in another words means it is high probable that it has come from univariate distribution which has the zero count component as true absence as described in the model.

$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{\lambda}_{00}$	$\hat{\lambda}_{10}$	$\hat{\lambda}_{20}$
0.4616	0.1485	0.3899	7.3897	13.8546	6.8511

Table 6.4: Estimated parameters of mixture model with Poisson distribution

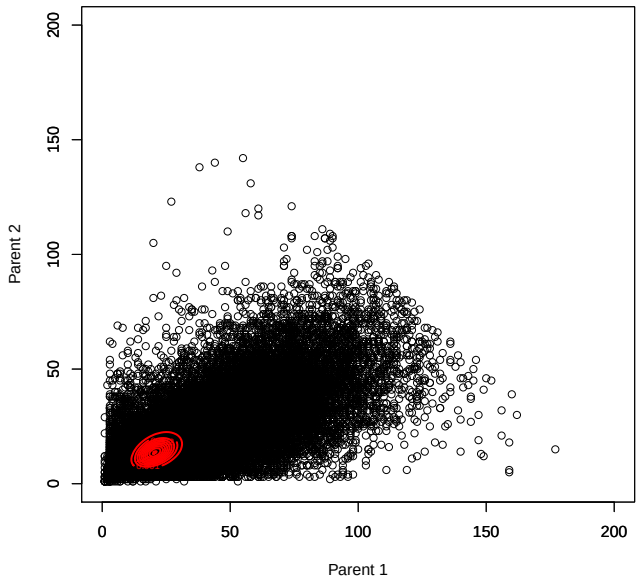

Figure 6.3: Contours of estimated mixture model of Poisson distribution on the data

6.3.3 Results and Discussion

The estimated values for our data using described mixture model with Poisson distribution is in Table. 6.4. Although the algorithm converges and we can find the maximum of the likelihood function and our estimator can be verified by simulation, but there are some facts that confirm Poisson distribution is not a good distribution for our data. Fig. 6.3 shows how Poisson distribution contour can not capture all of the bivariate part of the data. The reason of this is that in the Poisson distribution, mean and variance is the same and therefore it decays soon specially when mean does not a have a high value. In fact, the estimator is estimating the parameters precisely but the distribution is not a right distribution for this data.

Moreover, since the likelihood function has factorials in denominator for the bivariate part, for counts more than 172, it is equal to infinity according to [15] and

therefore causes likelihood function to become zero. This is another reason that Poisson distribution can not be a good distribution for the mixture model.

Considering negative binomial distribution and investigating if it can capture the data is an interesting topic of research and is a good opportunity for future work.

7. CONCLUSION

In this thesis at the first part we investigated and implemented an algorithm to produce high quality barcodes in order to do multiplexed sampling type of run in Illumina machines. Those high quality barcodes help us utilize high throughput feature of Illumina HiSeq machine even more efficiently while not compromising the demultiplexing which is a crucial step toward separating barcoded samples.

To accomplish this task, as we proposed, the barcodes should have 4 major characteristics such as balance in all positions, diversity, proper hamming distance and low GC content to minimize errors that may occur during the procedure in the sequencing machine. Designing a fast algorithm to produce such barcodes is another milestone for this task which has been done in this thesis. The algorithm is able to report the maximum number of barcodes that can be produced with given values for mentioned features which are set by user.

As a second contribution of this thesis, we studied different common biomarkers along with their applications in detail. We investigated different genetic maps as well in making of which biomarkers play an important rule. The main focus of our work was on RADSeq markers which are RAD markers that are produced using NGS techniques. Properties of such markers were studied as well.

The targeted problem of the second part of the thesis was to call true states of presence/absence markers which are one type of RADSeq biomarkers. Because of high correlation between the samples' counts, their presence/absence state can not be determined merely based on counts. To solve this problem, two statistical approaches are studied, a heuristic and a model based approach. For the model based approach we first proved that continuous distributions can not be used for such problem and

among two candidates of discrete distributions, Poisson and Negative Binomial, we investigated Poisson and we left Negative Binomial as a future work in my Ph.D. The results for estimated parameters in both approaches are reported in different tables in correspondent chapters.

REFERENCES

[1] Daniel Aird, Michael G Ross, Wei-Sheng Chen, Maxwell Danielsson, Timothy Fennell, Carsten Russ, David B Jaffe, Chad Nusbaum, and Andreas Gnirke. Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol*, 12(2):R18, 2011.

[2] Angel Amores, Julian Catchen, Allyse Ferrara, Quenton Fontenot, and John H Postlethwait. Genome evolution and meiotic maps by massively parallel DNA sequencing: spotted gar, an outgroup for the teleost genome duplication. *Genetics*, 188(4):799–808, 2011.

[3] Nathan A Baird, Paul D Etter, Tressa S Atwood, Mark C Currey, Anthony L Shiver, Zachary A Lewis, Eric U Selker, William A Cresko, and Eric A Johnson. Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PloS ONE*, 3(10):e3376, 2008.

[4] Luis B Barreiro, Guillaume Laval, Hélene Quach, Etienne Patin, and Lluís Quintana-Murci. Natural selection has driven population differentiation in modern humans. *Nature Genetics*, 40(3):340–345, 2008.

[5] Jürg Berger, Takashi Suzuki, Kirsten-André Senti, Janine Stubbs, Gotthold Schaffner, and Barry J Dickson. Genetic mapping with SNP markers in *Drosophila*. *Nature Genetics*, 29(4):475–481, 2001.

[6] David Botstein, Raymond L White, Mark Skolnick, and Ronald W Davis. Construction of a genetic linkage map in man using restriction fragment length polymorphisms. *American Journal of Human Genetics*, 32(3):314, 1980.

[7] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[8] Julian M Catchen, Angel Amores, Paul Hohenlohe, William Cresko, and John H Postlethwait. Stacks: building and genotyping loci de novo from short-read sequences. *G3: Genes, Genomes, Genetics*, 1(3):171–182, 2011.

[9] Judy C Chang and Yuet Wai Kan. beta 0 thalassemia, a nonsense mutation in man. *Proceedings of the National Academy of Sciences*, 76(6):2886–2889, 1979.

[10] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, Cambridge, MA, 2001.

[11] Aniruddha Datta and Edward R Dougherty. *Introduction to genomic signal processing with control*. CRC Press, Boca Raton, FL, 2007.

[12] John W Davey and Mark L Blaxter. RADSeq: next-generation population genetics. *Briefings in Functional Genomics*, 9(5-6):416–423, 2010.

[13] John W Davey, Timothée Cezard, Pablo Fuentes-Utrilla, Cathlene Eland, Karim Gharbi, and Mark L Blaxter. Special features of RAD sequencing data: implications for genotyping. *Molecular Ecology*, 22(11):3151–3164, 2013.

[14] Mohd Fareed and Mohammad Afzal. Single nucleotide polymorphism in genome-wide association of human population: A tool for broad spectrum service. *Egyptian Journal of Medical Human Genetics*, 14(2):123–134, 2013.

[15] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48, 1991.

[16] David B Goldstein, A Ruiz Linares, Luigi Luca Cavalli-Sforza, and Marcus W Feldman. An evaluation of genetic distances for use with microsatellite loci. *Genetics*, 139(1):463–471, 1995.

[17] Ada Hamosh, Terri M King, Beryl J Rosenstein, Mary Corey, Henry Levison, Peter Durie, Lap-Chee Tsui, Iain McIntosh, Marion Keston, David JH Brock, et al. Cystic fibrosis patients bearing both the common missense mutation, gly asp at codon 551 and the δf508 mutation are clinically indistinguishable from δf508 homozygotes, except for decreased risk of meconium ileus. *American Journal of Human Genetics*, 51(2):245, 1992.

[18] Paul A Hohenlohe, Susan Bassham, Paul D Etter, Nicholas Stiffler, Eric A Johnson, and William A Cresko. Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. *PLoS Genetics*, 6(2):e1000862, 2010.

[19] William C Horrace. Some results on the multivariate truncated normal distribution. *Journal of Multivariate Analysis*, 94(1):209–221, 2005.

[20] http://res.illumina.com/documents/products/brochures/brochure_hiseq_systems.pdf.

[21] http://res.illumina.com/documents/products/illumina_sequencing_introduction.pdf.

[22] http://res.illumina.com/documents/systems/hiseq/datasheet_hiseq_systems.pdf.

[23] V. M. Ingram and A New. A specific chemical difference between the globins of normal human and sickle-cell anemia hemoglobin. *Nature*, 178, 1956.

[24] Damian Jaccoud, Kaiman Peng, David Feinstein, and Andrzej Kilian. Diversity arrays: a solid state technology for sequence information independent genotyp-

ing. *Nucleic Acids Research*, 29(4):e25–e25, 2001.

[25] Philippe Jarne and Pierre JL Lagoda. Microsatellites, from molecules to populations and back. *Trends in Ecology & Evolution*, 11(10):424–429, 1996.

[26] CG Khatri and MC Jaiswal. Estimation of parameters of a truncated bivariate normal distribution. *Journal of the American Statistical Association*, 58(302):519–526, 1963.

[27] Chin-Shang Li, Jye-Chyi Lu, Jinho Park, Kyungmoo Kim, Paul A Brinkley, and John P Peterson. Multivariate zero-inflated Poisson models and their applications. *Technometrics*, 41(1):29–38, 1999.

[28] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[29] Michael L Metzker. Sequencing technologies: the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2009.

[30] Michael R Miller, Joseph P Dunham, Angel Amores, William A Cresko, and Eric A Johnson. Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. *Genome Research*, 17(2):240–248, 2007.

[31] A. Nikooienejad, R. Metz, B. J. Yoon, and C. D. Johnson. Fast DNA barcode generating algorithm using Radix Coding method. In *Genomic Signal Processing and Statistics, (GENSIPS), 2012 IEEE International Workshop on*, pages 26–30, 2012.

[32] Erik Pettersson, Joakim Lundeberg, and Afshin Ahmadian. Generations of sequencing technologies. *Genomics*, 93(2):105–111, 2009.

[33] Jesse A Poland and Trevor W Rife. Genotyping-by-sequencing for plant breeding and genetics. *The Plant Genome*, 5(3):92–102, 2012.

[34] Lee M Silver. *Mouse genetics: concepts and applications.* Oxford University Press, New York, NY, 1995.

[35] Monica Singh, Puneetpal Singh, Pawan Kumar Juneja, Surinder Singh, and Taranpal Kaur. SNP–SNP interactions within APOE gene influence plasma lipids in postmenopausal osteoporosis. *Rheumatology International*, 31(3):421–423, 2011.

[36] Heather L Stickney, Jeremy Schmutz, Ian G Woods, Caleb C Holtzer, Mark C Dickson, Peter D Kelly, Richard M Myers, and William S Talbot. Rapid mapping of zebrafish mutations with SNPs and oligonucleotide microarrays. *Genome Research*, 12(12):1929–1934, 2002.

[37] Peter Wenzl, Jason Carling, David Kudrna, Damian Jaccoud, Eric Huttner, Andris Kleinhofs, and Andrzej Kilian. Diversity arrays technology (DArT) for whole-genome profiling of barley. *Proceedings of the National Academy of Sciences*, 101(26):9915–9920, 2004.

[38] Stephen R Wicks, Raymond T Yeh, Warren R Gish, Robert H Waterston, and Ronald HA Plasterk. Rapid gene mapping in *Caenorhabditis* elegans using a high density polymorphism map. *Nature Genetics*, 28(2):160–164, 2001.

[39] Stefan Wilhelm and BG Manjunath BG. tmvtnorm: A package for the truncated multivariate normal distribution. *Sigma*, 2:2, 2010.

[40] Rongling Wu, Changxing Ma, and George Casella. *Statistical genetics of quantitative traits: Linkage, maps and QTL.* Springer, New York, NY, 2010.

[41] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18(5):821–829, 2008.