# IDENTIFYING PERSON THROUGH DNA FINGERPRINTING ON MIXED SAMPLE USING NEXT-GENERATION SEQUENCING DATA

BY

JOSHUA CHAN MUN WEI

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2017

# DECLARATION OF ORIGINALITY

I declare that this report entitled "**IDENTIFYING PERSON THROUGH DNA FINGERPRINTING ON MIXED SAMPLE USING NEXT-GENERATION SEQUENCING DATA**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________________

Name : ____________________________

Date : ____________________________

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Ng Yen Kaow for giving me this opportunity to take up this project as it is a challenging, yet interesting subject. It is his diligence towards research and enthusiasm in regard to teaching that persuaded me in taking up the project. Without his guidance, patience and confidence, this project would not have been possible.

Not to forget ZiCheng, Zhao from City University of Hong Kong who has been giving me insights about the project, thanks for being so helpful and always available. Finally, I would like to thank my parents and family for their love and support.

# ABSTRACT

DNA fingerprinting has long been used to identify a person by characteristics of their DNA. In forensic science, DNA analysis has been useful in criminal investigations to find out whether a suspect was in the crime scene and also used as evidence in court cases. However, forensic samples sometimes contain DNA from two or more individuals and are difficult to interpret. Traditionally, researchers were able to interpret mixture by the signal peak height and area produced by first generation sequencing. Various methods that can be categorised as Frequentist approaches and Bayesian approaches are designed to evaluate DNA mixtures.

With the emergence of NGS technologies, the sequencing of billions of DNA molecules can be parallelised; greatly increasing the throughput and reducing the associated costs. Alleles that have similar lengths that were indiscernible using first generation sequencing techniques are now easily distinguished. In this project, we proposed a new mathematical model and design a likelihood ratio method that handles NGS data to interpret DNA mixtures. A software toolkit is also developed to test and verify the method.

We have applied the method to 4480 simulated DNA mixtures of various mixture proportions using 8 unrelated individuals in an unpublished dataset from Beijing Genomics Institute. The results confirms the feasibility of utilizing NGS data in DNA mixture interpretations. Among the positively labelled results, the mean likelihood ratio for two-person mixtures is as high as $\log_{10} 285978$. Using our method, all 224 identity tests for two-person mixtures and three-person mixtures were correctly identified. This project serves as a basis to implementing likelihood ratio analysis of DNA mixture using NGS data.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| *DNA* | Deoxyribonucleic Acid |
| *FBI* | Federal Bureau of Investigation |
| *MCMC* | Markov chain Monte Carlo |
| *NGS* | Next-generation Sequencing |
| *PCR* | Polymerase Chain Reaction |
| *STR* | Short Tandem Repeat |
| *BGI* | Beijing Genomics Institute |

# Chapter 1: Introduction

## 1.1 Motivation and Problem Statement

With regards to the analysis of DNA mixture for forensic purposes, the procedures currently in use are still highly variable despite improvements in protocols and interpretation guidelines. Other than that, there seems to be subjectivity and bias in complex mixture interpretation. In an experiment conducted by Dror and Hampikian (2011), 17 independent DNA expert analysts gave different conclusions when examining the same DNA profiles.

There are grave consequences when a mixed DNA sample is misinterpreted. It is possible to have false inclusions and exclusions in cases where the DNA profile of an innocent person is misinterpreted as a match with the suspect or when the true contributor is said to have a mismatch with the sample given a low DNA template. Besides, results from DNA analysis of complex mixtures are often inconclusive when factors such as sample degradation, low template concentration, allele drop-in, drop-out and stutters that occurs prevalently are not considered (Haned, *et al*., 2015).

Over the last decade, many new techniques have been developed for complex mixture interpretation. Mathematical models that take into account the challenges above were also formulated. Nevertheless, some of the methods often yield inconclusive results or are computationally expensive.

Traditionally, DNA sequencing is known to have low throughput, high cost and operation difficulties. With the emergence of Next-generation Sequencing (NGS) technologies, the sequencing of billions of DNA molecules can be parallelised; greatly increasing the throughput and reducing the associated costs. Alleles that have similar lengths that were indiscernible using first generation sequencing techniques are now easily distinguished (Yang, *et al*., 2014). These advantages make STR analysis possible in low DNA templates and give fast and accurate allele determination.

## 1.2 Project Scope

The scope of the project includes the development of a likelihood ratio method that uses NGS data in a probability model to do DNA testing on mixed samples. The probability model should take into account aspects such as mixture scenario, sequencing errors, drop-in, drop-out, and stutters.

## 1.3 Project Objectives

The objectives of the project are:

i. To formulate a likelihood ratio method that utilizes NGS data to interpret DNA mixtures.

ii. To consider the aspects such as mixture scenario, sequencing errors, drop-in, drop-out, and stutters in the probability model.

iii. To develop a software toolkit that uses the formulated likelihood ratio method to handle DNA mixture using NGS data.

## 1.4 Proposed Approach

The proposed approach to tackle the problems stated is to develop a likelihood ratio method that utilizes NGS data to analyse DNA mixtures. The mathematical model should consider sequencing errors and can handle various mixture scenarios. The devised method allows DNA mixture interpretation to be carried out in a standardised and reliable way. To prove that the proposed method works, a software toolkit that uses the formulated likelihood ratio method will be developed to test and verify its correctness. Experiments are carried out to find out cases where the devised method is applicable as well as its limitations.

# Chapter 1: Introduction

## 1.5 Background information

Deoxyribonucleic acid (DNA) is a molecule that contains the genetic code for an organism. DNA, usually present in the nucleus of a cell, is responsible for the way proteins are made. The part of the DNA that carries the codes for a protein is called a *gene*. Different individuals can have different variants of the same gene due to minor mutation, these variants are known as *alleles*. Generally, each person has two alleles at each gene locus inherited from their parents.

DNA fingerprinting, also known as DNA profiling has been used to identify a person by the characteristics of their DNA. In forensic science, DNA analysis has been useful in criminal investigations to find out whether a suspect was at the crime scene; such findings have been used as evidence in court cases. Despite the fact that a high fraction of human DNA sequences are the same in every person, there are differences in certain regions (Norrgard, 2008). It was found out that there are recurrent patterns of DNA units that change in length among individuals; one of such repeating sequences is the so-called short tandem repeat (STR). By comparing the STR loci between DNA samples, a probability of match can be calculated for identity testing. Using the 13 core STR loci which were identified by the Federal Bureau of Investigation (FBI), a study shows that the likelihood of a complete match between profiles of two unrelated Caucasians is approximately 1 in 575 trillion (Reilly, 2001).

Assuming that the DNA from a crime scene evidence is in good quality and a single person is the sole contributor to the sample, identity testing using DNA analysis is accurate and reliable. However, forensic evidentiary samples often contain DNA from multiple contributors. In such cases, the interpretation of DNA evidence is relatively complex and subjective. It is important to note that not the entire genome is compared during DNA testing; instead, only a small subset of the loci that have high variability is used.

## 1.6 Report Organisation

This report consists of 6 chapters in total. The first chapter introduces the project, the motivations for working on the project and also the project objectives. In the second chapter, we study and review past researches on the topic of DNA mixture analysis.

Chapter 3 explains our new likelihood ratio method in detail. The definitions and equations of the mathematical model are in this chapter. Chapter 4 describes the methodology and tools used in the project in addition to the timeline of the project. Chapter 5 details experimental design and results of the devised method in the project.

The final chapter concludes the project and explains on possible future improvements on the finished system.

# Chapter 2: Literature Review

## 2.1 Introduction

Over the course of the years, a number of methods have been developed for DNA mixture interpretation. They can be characterised into Frequentist approaches and Bayesian approaches. Frequentist approaches interpret the mixture data by taking account of observed genotype and population allele frequency to calculate a probability of exclusion (which will be explained in Section 2.2). Bayesian approaches are usually associated with calculating likelihood ratios and may or may not utilize quantitative information.

By taking advantage of techniques such as automated fluorescent labelling in the Sanger method (first generation sequencing), researchers were able to interpret a mixture by its signal peak height and area. The progenitor of employing quantitative data in mixtures, now termed the *binary model*, is the paper by Evett, *et al*. in 1991. However, the model is not only contingent on the resolution of two-person mixtures, but also relies on the experience of experts upon the observed height and shape of the sequencing signal.

Although further improvements have been made on the binary model, such as the use of linear mixture analysis through minimizing the sum of squared deviation across all loci (Perlin & Szabady, 2001), the reliability of the model is still questionable. Recent studies adopt the use of Markov chain Monte Carlo (MCMC) methods to formulate a continuous model, as demonstrated in the paper by Buckleton, *et al*. published in 2013.

## 2.2 Frequentist Approach – Probability of Exclusion

The Frequentist approach of solving complex mixture interpretation problem depends on the calculation of an exclusion probability. The probability of exclusion is defined as “the probability that a random individual would be excluded as a contributor to a DNA mixture”. In the case where a suspect’s DNA profile matches the mixture, then the probability of exclusion would be important. Practically, a high probability of exclusion would mean that the suspect is more likely to be the only valid contributor to the mixture. Conversely, a low probability of exclusion means that the mixture contains DNA sequences that are commonly found in a population – the suspect might just be falsely included.

Given a mixture of DNA that consists of alleles $A_1, \dots, A_n$, we want the probability of exclusion at locus $l, PE_l$. First, we consider the possible genotypes that can be formed entirely within the locus; which is the sum of all heterozygotes and homozygotes:

$\sum_{i \neq j} p_i p_j + \sum_{i=1}^{n} p_i^2$, where $p_i$ is the allele frequency $A_i$.

The probability of exclusion is then:

$$PE_l = 1 - \sum_{i \neq j} p_i p_j + \sum_{i=1}^{n} p_i^2$$

$$= 1 - \left( \sum_{i=1}^{n} p(A_i) \right)^2$$

Considering all loci, the total probability of exclusion is:

$$PE = 1 - \prod_{l} (1 - PE_l)$$

Using this approach has its benefits, such as its simplicity, and that it does not require us to know the number of contributors to the DNA sample beforehand. Still, it is commented that there is a substantial waste of information as the genotype of suspect is not taken into account (Buckleton, *et al.*, 2005).

## 2.3 Bayesian Approaches

### 2.3.1 Qualitative Method

The Bayesian approach for DNA mixture analysis typically involves the calculation of a likelihood ratio. To form the likelihood ratio, the primary step is usually the formulation of the null hypothesis and the alternative hypothesis. This step is non-trivial as it is dependent on the scenario of the casework and requires understanding of the related problems.

To demonstrate, consider a case where a woman who has been assaulted claims to have scratched her attacker. Suppose the fingernails of the woman contain the mixed DNA profiles of two individuals. In such case, it is obvious to see that the profile of the donor herself is present in the DNA sample. Hence, the most reasonable hypotheses would most likely be as follow:

$H_p$: The sample is composed of the DNA of the donor and the suspect.
$H_d$: The sample is composed of the DNA of the donor and an unknown person.

Note that the inclusion of the DNA of the donor in both hypotheses makes the deduction of the other contributor's DNA profile rather straightforward – the alleles that does not belong to the donor belongs to the other contributor. This property applies to mixture that is made up of more than 2 contributors as well, though there will be an increase in complexity.

Denoting the DNA mixture as $M$ and the genotypes of the donor and suspect as $G_d$ and $G_s$ respectively, the likelihood ratio is calculated as:

$$LR = \frac{\Pr(M \mid G_d, G_s, H_p)}{\Pr(M \mid G_d, H_d)}$$

The likelihood ratio essentially tells us how much more probable that the mixture data would be if the suspect is included as a contributor than if the suspect is excluded.

### 2.3.2 Binary Model

Originally, the qualitative method require analysts to consider the possible genotype combinations under each hypothesis to calculate the likelihood ratio. Effectively, the probability of 1 or 0 is assigned to each genotype; however, this is undesirable since all possible genotypes should be weighted between 0 and 1. Thus, the binary model extends from the initial qualitative method by incorporating quantitative information such as allele peak height and peak area.

The binary model makes several assumptions:
(1) Across all loci, the mixture proportion is approximately constant.
(2) The peak area depends on the amount of DNA.
(3) The area of common peaks is the sum of the areas of the contributors.

**Heterozygous Balance**

Heterozygous balance is defined as the ratio of two chosen allele peaks:

$$Hb = \frac{\phi_1}{\phi_2}$$

where $\phi_1$and $\phi_2$ are the peak area of allele 1 and allele 2 respectively. The order of the alleles can be selected arbitrarily and 2 alleles are said to be consistent with the proposition that they came from the same person if Hb falls within the range:

$$0.6 \ \leq Hb \leq 1.66$$

**Mixture Proportion**

It was demonstrated that the mixture proportion in a complex DNA sample is approximately the same throughout all the loci compared. Consider that two contributors to a mixed sample have the allele pairs ($a$, $b$) and ($c$, $d$). From the peak areas ($\phi$), the mixture proportions for the contributors would be estimated as: $\widehat{M}_x = \frac{\phi_a+\phi_b}{\phi_a+\phi_b+\phi_c+\phi_d}$ and $1 - \widehat{M}_x$. Typically, the mixture proportions are allowed to be within $\pm 0.35$ of the average $\widehat{M}_x$ calculated across all loci.

### Accounting for Stutters

When doing STR analysis, STR loci are usually amplified through the polymerase chain reaction (PCR) process; this allows small amounts of DNA to be analysed. However, there is a possibility that artefacts such as drop-in, drop-out and stutters can happen during PCR. In the case of allele drop-in, contamination on DNA are magnified alongside the samples during PCR-amplification; causing spurious allele peak to be observed. On the other hand, true alleles that fail to PCR-amplify can cause drop-out where alleles cannot be visualized as it falls below the detection threshold. Stutters are artificial peaks caused by stochastic effects such as miscopying or slippage in the PCR process.

Looking at Figure 2.1, the $b$ band can be considered as a stutter. Assuming that the mixture is composed of DNA from 2 persons, the genotype combinations of the minor contributor could be $aa$, $ac$ or $ad$ if peak $b$ is a stutter. If peak $b$ is taken as allelic (that is, the peak is an effect of a true allele and not an artefact), the genotype combination would then most likely be $ab$ (Buckleton, et al., 2013).

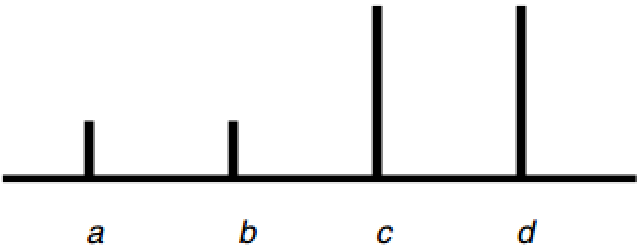


**Figure 2.1** Madonna Plot showing a DNA profile comprising two minor bands $a$, $b$ and two major bands $c$, $d$. The $b$ band can be considered as a stutter when it is <15% the area of the major band and the distance between $b$ and $c$ is one repeat unit.

We can treat such a case by assuming that the peak at $b$ has a probability $p(S)$ of being a stutter and $p(\bar{S})$ as being allelic. The likelihood ratio would then take a bound as:

$$LR = \frac{1}{p(\bar{S})2p_a p_b + p(S)\{p_a^2 + 2p_a p_c + 2p_a p_d\}}$$

$$\geq \frac{1}{\{p_a^2 + 2p_a p_b + 2p_a p_c + 2p_a p_d\}}$$

### 2.3.3 Continuous Model

While the binary model improves on earlier model by utilizing quantitative information of a DNA profile in some properties, it has not made full use of the data. In fact, the binary model selects genotype combinations in a discrete manner – the genotype combinations are either selected, or not selected.

In order to ensure that every genotype combination is accounted for in the probability model, it is best to give a weight to each allele peak observed in the DNA profile.

Denote $d$ as the vector of peak areas:

$$d = \{\phi_a, \phi_b, \phi_c, \phi_d\}$$

Furthermore, we suppose there are $p$ combinations of possible genotypes under $H_p$ denoted as $S_1, \dots, S_p$. Similarly, there are $q$ combinations under $H_d$: $S_1, \dots, S_q$. Then, our objective is to calculate the likelihood ratio:

$$LR = \frac{\sum_{j=1}^{p} \Pr(d \mid S_j, H_p)\Pr(S_j \mid H_p)}{\sum_{k=1}^{q} \Pr(d \mid S_k, H_p)\Pr(S_k \mid H_p)}$$

**Multilocus Combination**

Recall that it was demonstrated that the mixture proportion through all loci are almost the same. We refer to Figure 2.2 and see that the most probable loci combination for locus 1 would be $ad$: $bc$ or $bc$: $ad$ and at locus 2 – $eh$: $fg$ or $fg$: $eh$. In a locus-by-locus approach, the whole genotype is not considered when comparing the probabilities of the genotype combinations. To illustrate, a locus-by-locus approach might not see a difference between the genotype combinations $adeh$: $bcfg$ and $adfg$: $bceh$; in which case the former is more supported than the latter.

Using the multilocus approach and introducing the mixing proportion $w$, we obtain:

$$LR = \frac{\sum_{j=1}^{p} \int Pr(d \mid S_j, w, H_p)\Pr(S_j \mid w, H_p)\Pr(w \mid H_p)dw}{\sum_{k=1}^{q} \int Pr(d \mid S_k, w, H_d)\Pr(S_k \mid w, H_d)\Pr(w \mid H_d)dw}$$

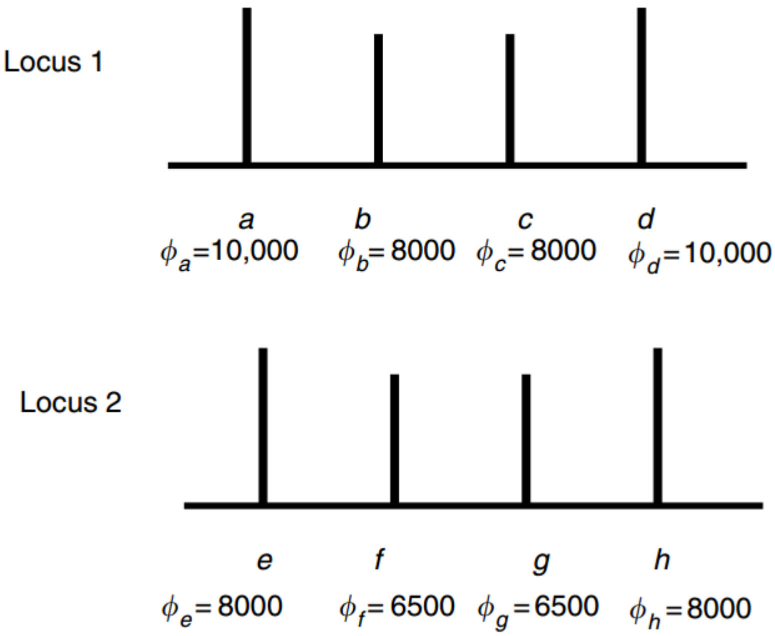


**Figure 2.2** Madonna plot for two loci.

### Markov chain Monte Carlo

There are researchers who try to employ MCMC methods to interpret DNA mixtures. The mass of an allele which encapsulates the concepts of template DNA amount, degradation level, locus amplification efficiency and replicate amplification efficiency is introduced and denoted $M$.

Including the mass parameter into the likelihood ratio gives the exact form:

$$LR = \frac{\sum_{j=1}^{p} Pr(d \mid S_j, M, H_p)\Pr(S_j \mid H_p)\Pr(M)}{\sum_{k=1}^{q} Pr(d \mid S_k, M, H_d)\Pr(S_k \mid H_d)\Pr(M)}$$

We do not know the mass parameter point values. However, we can integrate across all possible values to give:

$$LR = \frac{\int \sum_{j=1}^{p} Pr(d \mid S_j, M, H_p)\Pr(S_j \mid H_p)\Pr(M)\, dM}{\int \sum_{k=1}^{q} Pr(d \mid S_k, M, H_d)\Pr(S_k \mid H_d)\Pr(M)\, dM}$$

This integral is subsequently assessed using MCMC and a probability of acceptance/rejection would be calculated for each hypothesis.

# Chapter 3: System Design

## 3.1 Mixing DNA samples

We test our method on simulated DNA mixtures, a program is developed to mix two DNA samples $M_1, M_2$ with their mixture proportions $p_1, p_2 \in [0, 1]$. A $p_1$ value of 0.15 and $p_2$ value of 0.50 will generate a mixture with 15% of $M_1$ and 50% of $M_2$. To generate mixture samples with more than 2 contributors, we simply mix the two-person mixture at 100% proportion with another sample. That is, to generate a three-person mixture consisting of 10%, 20% and 30% of DNA from person 1, 2 and 3 respectively; we first mix the DNA of person 1 and 2 at mixture proportions $p_1 = 0.10$ and $p_2 = 0.20$ and then mix the subsequent mixture with person 3 at mixture proportions $p_1 = 1.00, p_2 = 0.30$.

Let $M$ be the mixture sample consisting of $N$ loci: $M = \{\ell_1, \ell_2, \dots, \ell_N\}$. We denote $\ell$ as the set of alleles at the specific locus. Each locus has $L$ alleles $\ell = \{\alpha_1, \alpha_2, \dots, \alpha_L\}$. Let $r_a$ be the number of reads supporting allele $a$. Let $rand$ be the function that returns a single uniformly distributed random number in the interval $[0,1]$. The pseudo code of the algorithm is as follows:

**Algorithm 1** Mix DNA

**Input**: NGS data from 2 individuals $M_1, M_2$ and their mixture proportions $p_1, p_2$.

**Output**: Output mixture $M_{output}$ composed of $M_1, M_2$ of specified proportions $p_1, p_2$.

```
I.   $M_{output} \leftarrow \emptyset$
II.  for all locus $\ell \in M_1$ do
         $\ell_{new} \leftarrow \emptyset$
         for all allele $\alpha \in \ell$ do
             $r_\alpha \leftarrow 0$
             for each read of allele $\alpha$ do
                 if $rand < p_1$ then
                     $r_\alpha \leftarrow r_\alpha + 1$
                 end if
             end for
             $\ell_{new} \leftarrow \ell_{new} \cup \{\alpha\}$
         end for
         $M_{output} \leftarrow M_{output} \cup \{ \ell_{new}\}$
     end for
III. for all locus $\ell \in M_2$ do
         $\ell_{new} \leftarrow \emptyset$
         for all allele $\alpha \in \ell$ do
             $r_\alpha \leftarrow 0$
             for each read of allele $\alpha$ do
                 if $rand < p_2$ then
                     $r_\alpha \leftarrow r_\alpha + 1$
                 end if
             end for
             $\ell_{new} \leftarrow \ell_{new} \cup \{\alpha\}$
         end for
         $M_{output} \leftarrow M_{output} \cup \{ \ell_{new}\}$
     end for
IV.  return $M_{output}$
```

## 3.2 Determining number of contributors in DNA sample

Let $n_i$ be the allele count for $i$-th allele and represent each genotype in locus $\ell$ as a vector of allele counts: $g_\ell = \{n_1, n_2, \dots, n_L\}$. We say that a genotype is *consistent* with a locus if $\forall n \in g_\ell, n > 0$. Denote $\mathbb{G}_c^\ell$ as the set of size $\binom{L+(2c-L)-1}{2c-L}$ which contains the genotypes that are consistent with $\ell$, assuming that the mixture is made up of $c$ contributors.

Consider a locus $\ell$ in a DNA mixture that is made up of 2 contributors of 3 alleles $a_1, a_2$, and $a_3$. In this case, there are $\binom{3+(4-3)-1}{4-3} = 3$ valid genotypes:

$$\mathbb{G}_2^\ell = \{\{2,1,1\}, \{1,2,1\}, \{1,1,2\}\}$$

Consider the 1st genotype in the set $\mathbb{G}_2^\ell$, in which $g = \{2,1,1\}$, the probability of observing 2 $a_1$, 1 $a_2$, and 1 $a_3$ is $p_1^2 p_2 p_3$, where $p_i$ is the allele frequency for allele $i$. There are $\frac{(2+1+1)!}{2!\times 1!\times 1!} = 12$ unique permutations of the genotype in the form of:

| Person 1 | Person 2 | Joint Probability |
|---|---|---|
| $a_1 a_1$ | $a_2 a_3$ | $p_1^2 p_2 p_3$ |
| $a_1 a_1$ | $a_3 a_2$ | $p_1^2 p_2 p_3$ |
| $a_1 a_2$ | $a_1 a_3$ | $p_1^2 p_2 p_3$ |
| $a_1 a_2$ | $a_3 a_1$ | $p_1^2 p_2 p_3$ |
| $a_1 a_3$ | $a_1 a_2$ | $p_1^2 p_2 p_3$ |
| $a_1 a_3$ | $a_2 a_1$ | $p_1^2 p_2 p_3$ |
| $a_2 a_1$ | $a_1 a_3$ | $p_1^2 p_2 p_3$ |
| $a_2 a_1$ | $a_3 a_1$ | $p_1^2 p_2 p_3$ |
| $a_3 a_1$ | $a_1 a_2$ | $p_1^2 p_2 p_3$ |
| $a_3 a_1$ | $a_2 a_1$ | $p_1^2 p_2 p_3$ |
| $a_2 a_3$ | $a_1 a_1$ | $p_1^2 p_2 p_3$ |
| $a_3 a_2$ | $a_1 a_1$ | $p_1^2 p_2 p_3$ |
| | | $Sum = 12 p_1^2 p_2 p_3$ |

The probability of observing a certain genotype combination $g = \{n_1, n_2, \dots, n_L\}$ is:

$$P(g) = \frac{(n_1 + n_2 + \cdots + n_L)!}{\prod_{i=1}^{L} n_i!} \prod_{i=1}^{L} p_i^{n_i}$$

The probability of observing alleles of locus $\ell$ for a mixture with $c$ contributors is then the summation over $\mathbb{G}_c^\ell$:

$$P_c(\ell) = \sum_{g \in \mathbb{G}_c^\ell} P(g)$$

We can then calculate the probability of observing a $c$-contributors DNA mixture $M$ as:

$$P_c(M) = \prod_{\ell \in M} P_c(\ell) \qquad (1)$$

Note that these equations only work if $2c \geq L$. To illustrate, if there are 3 alleles observed at locus $\ell = \{a_1, a_2, a_3\}$ and we want to say that the DNA sample is from 1 contributor $c = 1$; it is not possible as 1 person can only carry 2 alleles. In fact, there are $\binom{3+(2-3)-1}{2-3} = \text{undefined}$ genotypes that is consistent with locus $\ell$. This essentially limits the minimum number of contributor for a DNA mixture to $\left\lceil \frac{A}{2} \right\rceil$, where $A$ is the maximum number of observed alleles across all loci. To overcome this limitation, we take into account sequencing error in our mathematical model.

In NGS, an allele is said to be present if there are supporting reads reported in the sequencing process. A high number of reads means that there is a high chance the DNA contains the allele at the specific locus. We assume a binomial distribution for sequencing errors with probability of success equals to $p$ and error $q = 1 - p$, an allele is included and said to be truly observed if there is at least 1 supporting read.

Let $r_a$ be the number of reads supporting allele $a$, the probability of an allele being included is then $P(X \geq 1) = 1 - P(X = 0) = 1 - q^{r_a}$. Consider a scenario where a DNA mixture has $\leq$ 4 alleles in every locus except for one that contains 5 alleles ($L = 5$), we can now say that 1 of the 5 alleles is due to sequencing error and allow the possibility that the mixture is made up of 2 contributors despite $2c < L$.

Denote $\wp(\ell)$ as the power set of $\ell$. Note that there is a possibility for all $l \in \wp(\ell)$ to be the correct set of alleles in locus $\ell$ considering the existence of sequencing errors. However, some of these subsets cannot be explained by $c$ contributors, that is, when $2c < |l|$. To illustrate, consider a locus of the DNA sample of a single person with 3 alleles $\ell = \{a_1, a_2, a_3\}$ and the allele read counts are $r_{a_1} = 1, r_{a_2} = 2, \mathrm{r}_{\mathrm{a}_3} = 2$.

| $\boldsymbol{k}$ | $\boldsymbol{l}$ | $\boldsymbol{\ell - l}$ | Probability of Observing $\boldsymbol{l}$ (Equation) | Probability of Observing $\boldsymbol{l}$ (Value) |
|---|---|---|---|---|
| 0 | $\{\ \}$ | $\{a_1, a_2, a_3\}$ | $qq^2q^2$ | 0.0000003125 |
| 1 | $\{a_1\}$ | $\{a_2, a_3\}$ | $(1-q)q^2q^2$ | 0.0000059375 |
| 1 | $\{a_2\}$ | $\{a_1, a_3\}$ | $q(1-q^2)q^2$ | 0.0001246875 |
| 1 | $\{a_3\}$ | $\{a_1, a_2\}$ | $qq^2(1-q^2)$ | 0.0001246875 |
| 2 | $\{a_1, a_2\}$ | $\{a_3\}$ | $(1-q)(1-q^2)q^2$ | 0.0023690625 |
| 2 | $\{a_1, a_3\}$ | $\{a_2\}$ | $(1-q)q^2(1-q^2)$ | 0.0023690625 |
| 2 | $\{a_2, a_3\}$ | $\{a_1\}$ | $q(1-q^2)(1-q^2)$ | 0.0497503125 |
| 3 | $\{a_1, a_2, a_3\}$ | $\{\}$ | $(1-q)(1-q^2)(1-q^2)$ | 0.9452559375 |

Since the DNA sample is contributed by only 1 person, we ought to attribute at least 1 of the 3 alleles to sequencing error. The case in which $l = \{a_1, a_2, a_3\}$ cannot possibly be explained by 1 contributor despite having the highest probability, hence we know that $P_c(l) = 0, \text{if } 2c < |l|$.

We calculate the probability of observing $l$ as the set of observed alleles at locus $\ell$ as:

$$P(l \mid \ell) = \prod_{a \in l} 1 - q^{r_a} * \prod_{a \in \ell - l} q^{r_a}$$

Equation (1) then becomes:

$$P_c(M) = \prod_{\ell \in M} \sum_{l \in \wp(\ell)} (P(l \mid \ell) P_c(l)) \qquad (2)$$

**Algorithm 2** Calculate probability of observing alleles in locus $\ell$ given $c$ contributors

```
Input: locus $\ell$, number of contributor $c$
Output: $P_c(\ell)$
$P_c(\ell) \leftarrow 0$
if $2c < |\ell| \vee |\ell| = 0$ then
    return $P_c(\ell)$
$\mathbb{G} \leftarrow combinations\_with\_replacement(\ell, 2c - L)$
for $g \in \mathbb{G}$ do
    $g \leftarrow g \cup g_\ell$
    $P_c(\ell) \leftarrow P_c(\ell) + \frac{(n_1+n_2+\cdots,+n_L)!}{\prod_{i=1}^{L} n_i!} \prod_{i=1}^{L} p_i^{n_i}$
end for
return $P_c(\ell)$
```

**Algorithm 3** Calculate probability of observed mixture made up of $c$ contributors

```
Input: mixture $M$, number of contributor $c$
Output: $P_c(M)$
$P \leftarrow \{\}$
for $\ell \in M$ do
    $P_c(\ell) \leftarrow 0$
    for $l \in \wp(\ell)$ do
        $P_c(\ell) \leftarrow P_c(\ell) + P(l \mid \ell) P_c(l)$
    end for
    $P \leftarrow P \cup \{P_c(\ell)\}$
end for
$P_c(M) = \prod_{p \in P} p$
return $P_c(M)$
```

## 3.3 Identify person through DNA fingerprinting

To interpret DNA mixtures using likelihood ratios, we need to calculate the likelihoods of the mixture explained under different hypotheses. The weight of evidence can then be measured by comparing the posterior probabilities of the mixture under alternative hypotheses. The key to calculating the probability is to specify the known contributors to the mixture and the number of unknown contributors.

For explaining the calculation of likelihood ratio, we follow the same definitions earlier with some additional nomenclature provided in Table 3.1.
The likelihood ratio is:

$$LR = \frac{\mathrm{P}(M \mid H_p)}{\mathrm{P}(M \mid H_d)} = \frac{\prod_{\ell \in M} P_x\left(\mathbb{U}_{\ell_O,\ell_K} \mid \ell_O, \ell_K\right)}{\prod_{\ell \in M} P_x\left(\mathbb{U}_{\ell_O,\ell_K} \mid \ell_O, \ell_K\right)}$$

where

$$P_x\left(\mathbb{U}_{\ell_O,\ell_K} \mid \ell_O, \ell_K\right) = \sum_{g \in \mathbb{C}_x^{\mathbb{U}_{\ell_O,\ell_K}}} P(g)$$

**Table 3.1 Nomenclature for Calculations of Likelihood Ratio**

| | |
|---|---|
| $M$ | The mixture that contains a set of loci. |
| $c$ | The number of contributors in the mixture. |
| $x$ | The number of unknown contributors in the mixture. |
| $\ell_O$ | The set of observed alleles. |
| $\ell_K$ | The set of alleles from all known contributors. |
| $\mathbb{M}_{\ell_O,\ell_K}$ | The set of alleles that are in both $\ell_O$ and $\ell_K$ $(\ell_O \cap \ell_K)$. |
| $\mathbb{U}_{\ell_O,\ell_K}$ | The set of unexplained alleles $(\ell_O - \ell_K)$. |
| $\emptyset$ | The empty set. |
| $\mathbb{C}_x^{\mathbb{U}_{\ell_O,\ell_K}}$ | The set of all valid genotypes for $x$ contributors to carry at least 1 of each alleles in $\mathbb{U}_{\ell_O,\ell_K}$ and any in $\ell_O$ for the remaining $(2x - \lvert\mathbb{U}_{\ell_O,\ell_K}\rvert)$ alleles. |
| $P_x\left(\mathbb{U}_{\ell_O,\ell_K} \mid \ell_O, \ell_K\right)$ | The probability that $x$ unknown contributors carry the alleles in $\mathbb{U}_{\ell_O,\ell_K}$ and none of the alleles outside of $\ell_O$. |

To illustrate, suppose a locus $\ell$ of a 2-contributor DNA mixture composed of a victim and an attacker in an assault case contains alleles $abcd$. Furthermore, assume that the victim has alleles $ab$ and a suspect has alleles $cd$. Consider a hypothesis that states that the known contributors of the mixture is the victim and suspect. Since there are no unknown contributors and the alleles in $\ell$ can be fully explained from the victim and suspect. The probability of observing locus $\ell$ under such hypothesis can be calculated as:

$$P_0\left(\mathbb{U}_{\ell_O,\ell_K} = \emptyset \,\middle|\, \ell_O = abcd, \ell_K = abcd\right) = 1$$

On the other hand, consider a hypothesis that states that only the victim is the known contributor, and the attacker is unknown. It is observable that the unknown contributor's genotype combination can only be $cd$ or $dc$. Hence, the probability is calculated as:

$$P_1\left(\mathbb{U}_{\ell_O,\ell_K} = cd \,\middle|\, \ell_O = abcd, \ell_K = ab\right) = 2p_c p_d$$

Taking into account sequencing errors in a DNA mixture, the probability of observing mixture $M$ under hypothesis $H$ is calculated as:

$$\mathrm{P}(M \mid H) = \prod_{\ell \in M} \sum_{l \in \wp(\ell)} \left(P_x\left(\mathbb{U}_{l_O,l_K} \mid l_O, l_K\right) P(l \mid \ell)\right)$$

However, this method of calculation assumes that there is no sequencing error in the DNA samples of the specified contributors under $H$ and causes the probability model to be inflexible. Notably, the model would not tolerate a single difference in alleles across all loci between the mixture and a known contributor:

$$\text{if } \exists \ell \in M, \ell_K - \ell_O \neq \emptyset, \mathrm{P}(M \mid H) = 0$$

Once again, we assume a binomial distribution in allele reads and take into account sequencing error to amend on this issue. Let $P\left(l_K \mid \mathbb{M}_{l_O,\ell_K}\right)$ be the probability of observing $l_K$ as the set of truly observed alleles given alleles in $\mathbb{M}_{l_O,\ell_K}$. The probability of observing mixture $M$ under hypothesis $H$ is then:

$$\mathrm{P}(M \mid H) = \prod_{\ell \in M} \sum_{l \in \wp(\ell)} \left( P(l \mid \ell) \prod_{a \in \ell_K - l} q^{r_a} \sum_{l_K \in \wp\left(\mathbb{M}_{l_O,\ell_K}\right)} \left(P_x\left(\mathbb{U}_{l_O,l_K} \mid l_O, l_K\right) P\left(l_K \mid \mathbb{M}_{l_O,\ell_K}\right)\right) \right)$$

# Chapter 4: Methodology and Tools

## 4.1 Design Specifications

### 4.1.1 Methodology and General Work Procedure

For this research project, we want to design a likelihood ratio method that utilizes NGS data for mixture sample interpretation in forensic applications. Past researches about the problem are studied and used as a basis for the new method. A new mathematical model will be formulated to allow the calculation of likelihood ratio with NGS data. Whenever an idea is thought out, a software prototype will be programmed and empirical analysis will be done to verify the correctness of the idea.

We separate the project to a few phases: First, we develop a program to mix multiple NGS DNA samples. Second, we devise a mathematical model and algorithm to determine the number of contributors of a DNA sample. Lastly, we'll tackle the main problem which is to identify whether a person is a contributor to a DNA mixture using a likelihood ratio method.

### 4.1.2 Tools Used

BWA (Li and Durbin, 2009)

A software package used to map DNA sequences of the NGS data from BGI against the reference human genome.

lobSTR (Gymrek *et al.*, 2012)

A software tool to do STR profiling on the downloaded NGS data of 4 family trios from Beijing Genomics Institute (BGI).

Python

The Python programming language is used to develop the software toolkit in order to verify our method.

R

The R programming language is used to analyse and visualize the results obtained from experiments.

SAMTOOLS (Li *et al.*, 2009)

A software tool used to subsample the sequence alignment data generated from BWA.

### 4.1.3 System Performance Definition

The essential improvements of utilizing NGS compared to the most commonly used Sanger sequencing currently can be concluded as:

i. The high-throughput of NGS can generate genome-wide data with multiple sequencing depth with low cost. This will relieve the loci number limitation in Sanger sequencing technique.

ii. The high sequencing coverage gives intuitive view on candidate ratio on mixture interpretation.

iii. Since more loci are included in the calculation of the probability, we expect more accurate result of the analysis.

### 4.1.4 Verification Plan

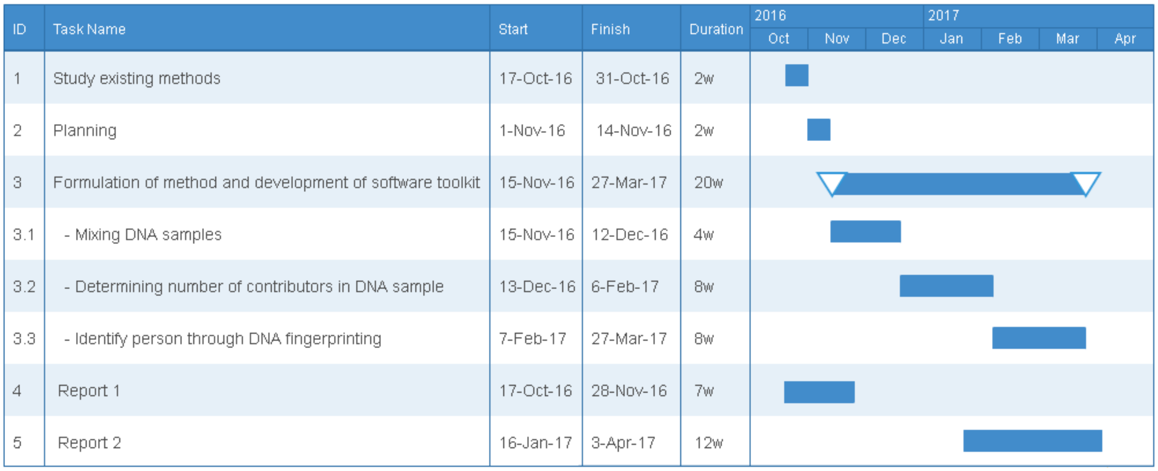
To test our method, we use the NGS data of 8 unrelated individuals in an unpublished dataset from BGI. The DNA sequences are aligned to the human genome reference using BWA with default settings and subsampled to half-fold using SAMTOOLS. The data is then further processed using the tool lobSTR for STR profiling. After that, the data is transformed to add information such as possible alleles and allele frequencies counted from the population. We randomly mix the sequencing data of multiple person from the data set with different mixture ratios to test the accuracy. To evaluate the robustness of the system, each test is run multiple times.

## 4.2 Implementation issues and challenges

Several difficulties and challenges are faced throughout the project. First of all, the research topic is considered to be of an interdisciplinary field that encapsulates computer science, statistic, mathematics and biology – bioinformatics. There is a substantial amount of knowledge to cover for a better understanding of the problem. Also, empirical analysis of the method is time-consuming since the processing of the large amount of NGS data is computationally intensive.

## 4.3 Timeline

The project spans the duration of two trimesters, it is approximately 24 weeks. The figure below shows the Gantt chart for the research project. There will be two report submissions during each trimester and a viva presentation to demonstrate the work done. The first report and presentation will show a system prototype and preliminary results. The full work and results will be demonstrated during the second session.



| ID | Task Name | Start | Finish | Duration |
|---|---|---|---|---|
| 1 | Study existing methods | 17-Oct-16 | 31-Oct-16 | 2w |
| 2 | Planning | 1-Nov-16 | 14-Nov-16 | 2w |
| 3 | Formulation of method and development of software toolkit | 15-Nov-16 | 27-Mar-17 | 20w |
| 3.1 | - Mixing DNA samples | 15-Nov-16 | 12-Dec-16 | 4w |
| 3.2 | - Determining number of contributors in DNA sample | 13-Dec-16 | 6-Feb-17 | 8w |
| 3.3 | - Identify person through DNA fingerprinting | 7-Feb-17 | 27-Mar-17 | 8w |
| 4 | Report 1 | 17-Oct-16 | 28-Nov-16 | 7w |
| 5 | Report 2 | 16-Jan-17 | 3-Apr-17 | 12w |

**Figure 4.1** Gantt chart

# Chapter 5: Implementation and Testing

## 5.1 Experimental Design

In testing our method, DNA mixtures were generated using NGS data of 8 unrelated individuals in an unpublished dataset from BGI. The data was sequenced using Illumina's HiSeq X Ten platform with read length of 150bp (base pair) and aligned to the HG19 reference human genome using BWA. The data is then processed using lobSTR for STR profiling, after which is transformed to incorporate information such as STR loci count, allele frequencies and possible alleles from population. In our experiments, we randomly mixed the sequencing data of multiple persons from the data set with different mixture proportions.

In this study, we simulated two-person and three-person mixtures to test our method. Also, the dataset we used has an average sequencing depth of 16-folds after sub-sampling. In our calculations of the likelihood ratio using the proposed method, it is assumed that the sequencing errors follow a binomial distribution with probability of observing an error equals to $q = 0.05$. In the experiment, we calculate the likelihood ratio as:

$$LR = \frac{P(M|H_p)}{P(M|H_d)}$$

## 5.2 Results

To find out how our LR method performs in general, we experiment on DNA mixtures of random mixture proportions. For simplicity, we assume $H_p$ to always be the better or more probable hypothesis. In practice, for a mixture containing DNA of individuals A and B, we set $H_p$ as the hypothesis that states that the mixture is made up of individuals A and B and $H_d$ as the hypothesis that states that the mixture is made up of 2 random persons. Following that, we say that a test result is correct whenever the calculated $\log_{10} LR$ is greater than 0.

We simulated DNA mixtures at random mixture proportions for all $\binom{8}{2} = 28$ two person combinations and $\binom{8}{3} = 56$ three person combinations. The experiment was conducted 100 times for two-person mixtures and 30 times for three-person mixtures. In total, we tested our method on a total of 4480 mixtures: 2800 two-person mixtures and 1680 three-person mixtures.

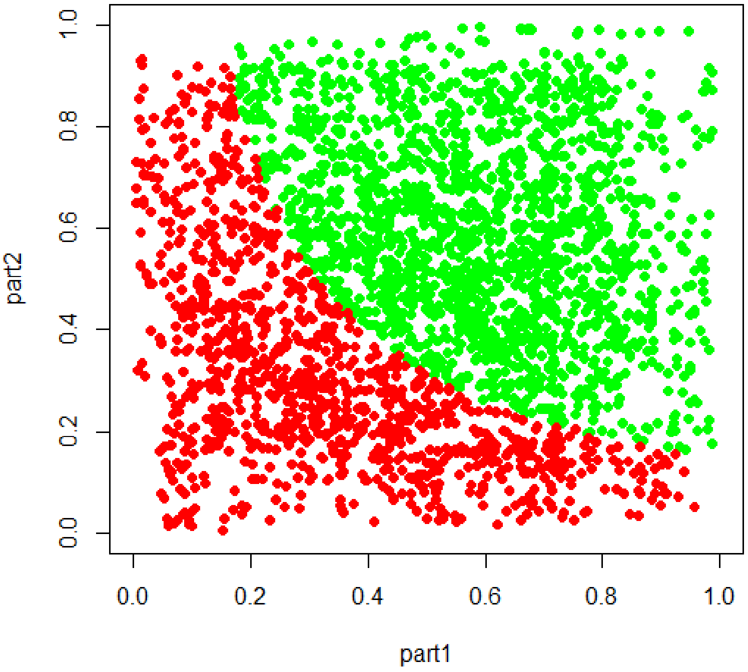


**Figure 5.1** Scatterplot showing mixture proportions of two-person mixtures. Green represent mixtures that are correctly labelled, red represents otherwise.

As shown in Figure 5.1, the mixture proportion plays a huge factor on whether or not the LR method gives correct results. In all cases where the LR method fails to give the correct answer, the DNA mixture is composed of low-template components. That is, at least 1 of the 2 contributors has less than 40% DNA information in the mixture.

The frequency distribution of the common logarithm of likelihood ratios for the cases that yielded positive results is constructed and shown in Figure 5.2. It is observed that most cases have a large value of likelihood ratio with a mean $\log_{10} LR$ as high as 285978.

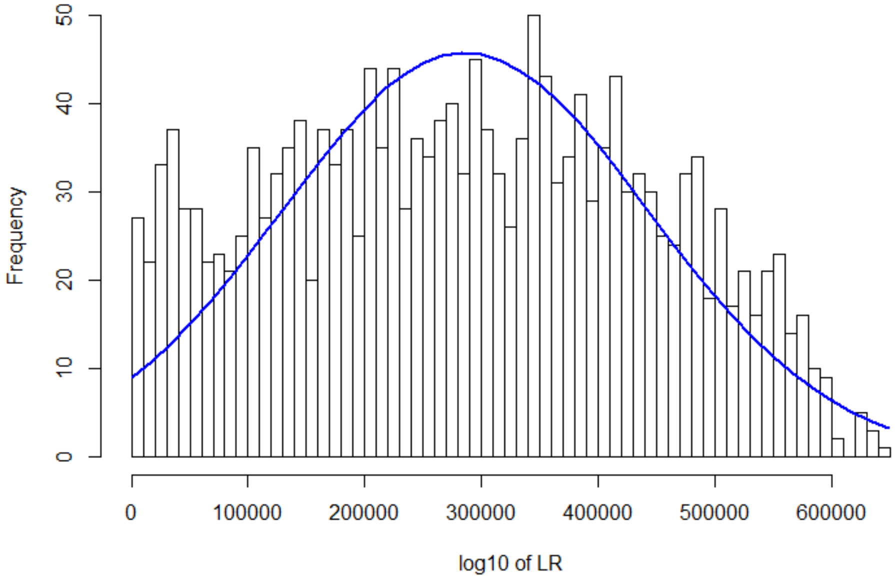


**Figure 5.2** Frequency distribution of common logarithm of LR of positive results for two-person mixtures. The mean of the frequency distribution is as high as 285978.

Similar results are observed for cases of three-person mixtures, Figure 5.3 depicts the 3D scatterplot of the mixture proportions for failed cases. Again, only mixtures with at least 1 low-template contributor are interpreted wrongly.

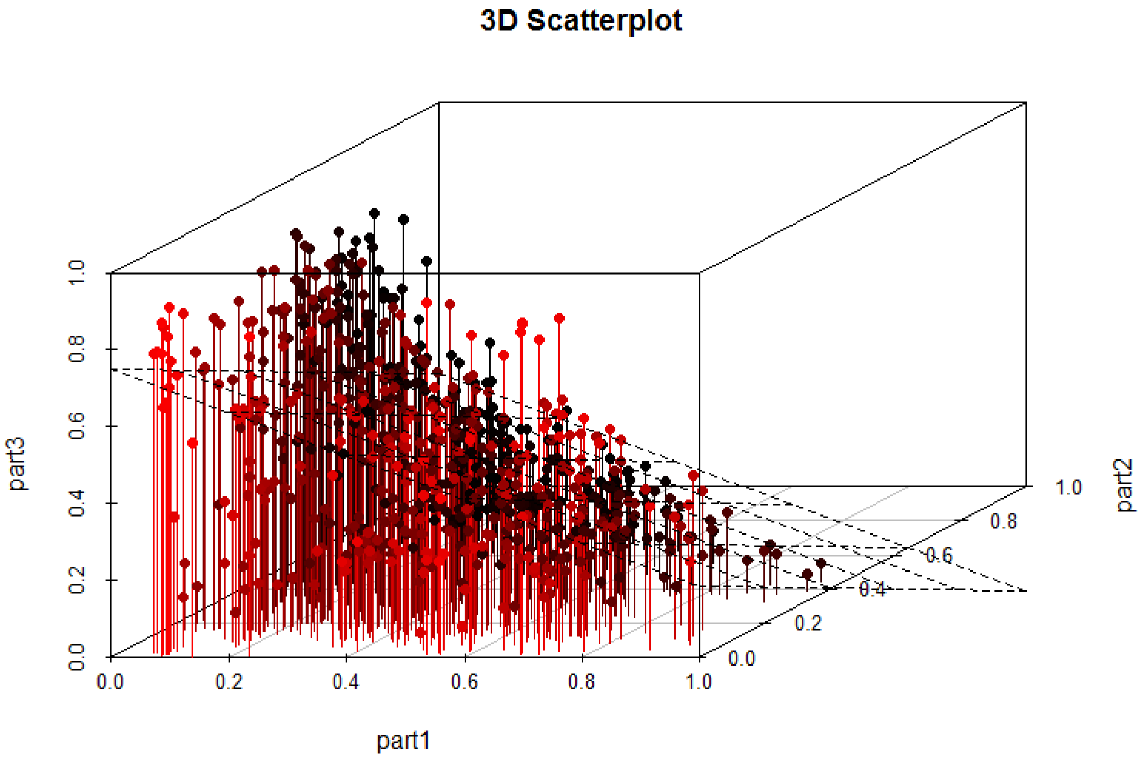


**Figure 5.3** 3D Scatterplot showing mixture proportions of falsely interpreted three-person mixtures.

### Identifying Person in Low-Template Mixtures

When analysing a low-template DNA mixture, the sequencing data is subject to high allele dropouts where alleles at a locus is not observed. As opposed to traditional DNA interpretation, our method processes millions of markers including STR and Single Nucleotide Polymorphism; this means that high allelic dropouts will greatly affect the hypothesis that has a fixed set of observed alleles.

To illustrate, consider a locus $\ell$ in a DNA sample of a person A with alleles $a_1$ and $a_2$. Due to allele dropout, the only observed allele in the locus is $a_1$. Now, since person A has a conflicting allele $a_2$, there exist a bias towards the probability that the sample is from a random person than that the sample is of person A.

In such cases, it might not be appropriate to make a judgement merely based on whether the calculated $\log_{10} LR$ is greater than 0. To handle such cases, we adopted a similar way used for prenatal paternity testing in a past research by Ryan *et al.* in 2012. In that study, hypothesis tests with $P$ value < 0.0001 *were* used to confirm paternity in 100% (20/20) of the cases using low fraction of fetal cfDNA (approximately 2.6% ~ 11.7%). For our use case, we assume that only 1% of DNA from every contributor is observed in a DNA mixture. Again, we simulate DNA mixtures from the dataset in all combinations.

Let $H_p$ be the hypothesis that states that person A is a contributor to the DNA mixture and $H_d$ be the hypothesis that states otherwise. We calculate the LR for every person in the dataset and construct a test statistic using the calculated LR. As shown in Figure 5.4, the LR calculated from the tests form a multinomial distribution with 2 separated clusters. It is observed that the test statistic when testing a correct individual always falls under the distribution with higher mean $\log_{10} LR$ (marked blue). Following this, we say that person A is a contributor to the DNA mixture if the calculated$\log_{10} LR$ is in the blue cluster. Using this method, we were able to identify correctly the contributors of low-template DNA mixtures with only 1% DNA from 2 persons in 100% (56/56) of the cases.

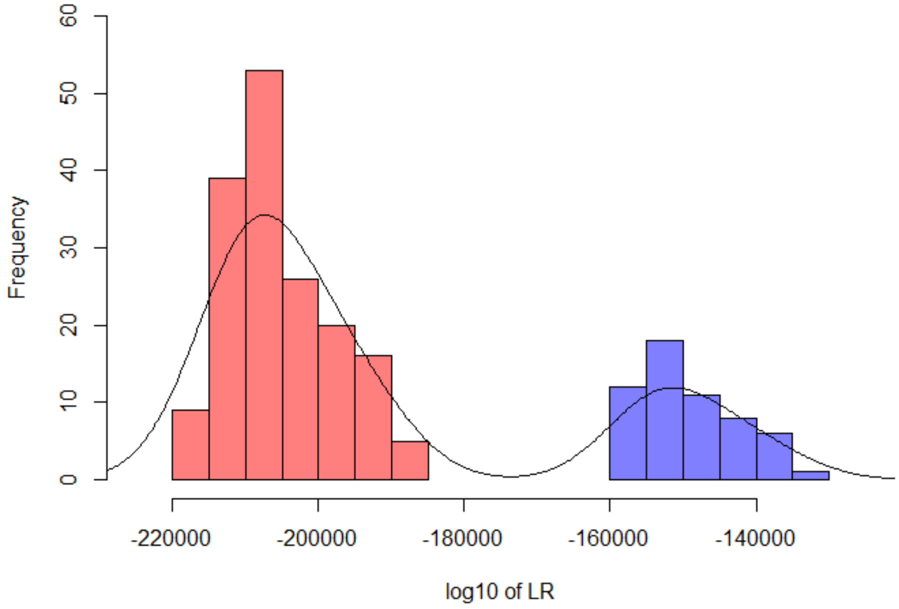


**Figure 5.4** Frequency distributions of calculated common logarithm of LR from identity tests for two-person mixtures with 1% DNA from 2 individuals.

As shown in Figure 5.5, we observe similar results when using the method on low template three-person mixtures. In all 56 three-person mixtures, 100% (168/168) of the contributors to the DNA mixtures were identified correctly.

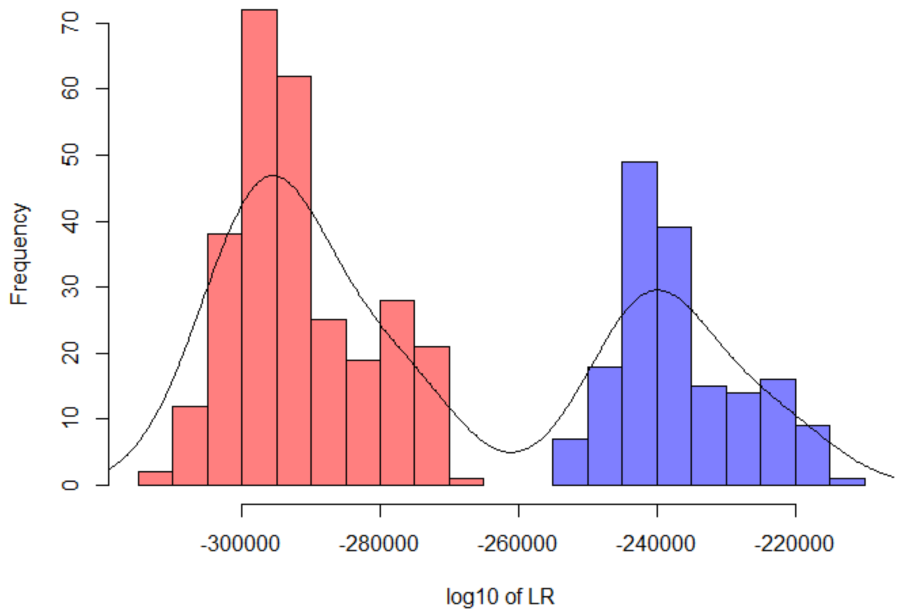


**Figure 5.5** Frequency distributions of calculated common logarithm of LR from identity tests for three-person mixtures with 1% DNA from 3 individuals.

## Chapter 6: Conclusion

### 6.1 Project Review

DNA fingerprinting has been useful in various court cases. However, there are cases where mixture samples are too difficult to be analysed using existing methods. It is crucial to ensure that a DNA mixture is interpreted correctly to prevent false inclusion or exclusion of a suspect.

In this project, we proposed a likelihood ratio method that uses NGS data to analyse DNA mixtures. By applying the method to 8 unrelated individuals of an unpublished dataset from BGI, we observed good results and high values of likelihood ratio in the interpretation of the DNA sample. By taking into account sequencing errors, the probability model proves to be advantageous and more robust.

The final result successfully meets the project objective to formulate a likelihood ratio method that utilizes NGS data as well as developing a software toolkit that uses the method.

### 6.2 Discussions

This study confirmed that NGS data can be incorporated into the analysis of multiple contributor DNA samples. From the experiments, the most obvious improvement of utilizing NGS data is the high power of discrimination it gives in interpreting DNA mixtures. Traditional mixture interpretation methods typically yields LR in the range of $(-\log_{10} 10\,, \log_{10} 10)$. Our method, however, is multiple orders of magnitude greater than traditional methods. This is attributed to the fact that genome-wide data is considered as opposed to only a number of selected loci in the calculations of the LR. Since more loci are included in the calculation of the probability, we expect more accurate results.

## 6.3 Future Work

In this study, the method is tested on simulated DNA mixtures from a sample of only 8 individuals. Considering a rather small sample size, a more large-scale experiment will be needed to confirm the effectiveness of the method.

The experiments were carried out under the assumption that sequencing errors follow a binomial distribution with 5% chance of observing an error. Further studies can be conducted to test the method using different control variables such as probability of sequencing error, sequencing depth or against higher order mixtures (four-person mixtures, five-person mixtures).

Other than that, mixtures with different proportions should be tested out to find out how it affects the results of the method. Before the method is used for real life applications, it should be tested against real DNA mixtures prepared in laboratories to verify its accuracy.

## Bibliography

Bill, M. et al., 2005. PENDULUM—a guideline-based approach to the interpretation of STR mixtures. *Forensic Science International*, 148(2-3), pp. 181-189.

Buckleton, J., Taylor, D. & Bright, J.-A., 2013. The interpretation of single source and mixed DNA profiles. *Forensic Science International: Genetics*, 7(5), pp. 516-528.

Buckleton, J., Triggs, C. M. & Walsh, S. J., 2005. Mixtures. In: *Forensic DNA evidence interpretation.* Boca Raton: CRC Press, pp. 226-283.

Dror, I. E. & Hampikian, G., 2011. Subjectivity and bias in forensic DNA mixture interpretation. *Science & Justice*, 51(4), pp. 204-208.

Evett, I. W., Buffery, C., Willott, G. & Stoney, D., 1991. A guide to interpreting single locus profiles of DNA mixtures in forensic cases. *Journal of the Forensic Science Society*, 31(1), pp. 41-47.

Gymrek, M., Golan, D., Rosset, S. & Erlich, Y., 2012. lobSTR: A short tandem repeat profiler for personal genomes. *Genome Research*, 22(6), pp. 1154-1162.

Haned, H., Benschop, C. C., Gill, P. D. & Sijen, T., 2015. Complex DNA mixture analysis in a forensic context: Evaluating the probative value using a likelihood ratio model. *Forensic Science International: Genetics*, Volume 16, pp. 17-25.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14), pp.1754-1760.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G. and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16), pp.2078-2079.

Norrgard, K., 2008. *Forensics, DNA Fingerprinting, and CODIS.* [Online]
Available at: http://www.nature.com/scitable/topicpage/forensics-dna-fingerprinting-and-codis-736
[Accessed 18 August 2016].

Perlin, M. W. & Szabady, B., 2001. Linear Mixture Analysis: A Mathematical Approach to Resolving Mixed DNA Samples. *Journal of Forensic Sciences,* 46(6), p. 15158J.

Reilly, P., 2001. Legal and public policy issues in DNA forensics. *Nature Reviews Genetics,* 2(4), pp. 313-317.

## Bibliography

Ryan, A., Baner, J., Demko, Z., Hill, M., Sigurjonsson, S., Baird, M. and Rabinowitz, M. (2012). Informatics-based, highly accurate, noninvasive prenatal paternity testing. *Genetics in Medicine*, 15(6), pp.473-477.

Weir, B. S., 1996. *Genetic Data Analysis II: Methods for Discrete Population Genetic Data.* 1st ed. Sunderland: Sinauer Associates.

Weir, B. S. et al., 1997. Interpreting DNA Mixtures. *Journal of Forensic Sciences*, 42(2), pp. 213-222.

Yang, Y., Xie, B. & Yan, J., 2014. Application of Next-generation Sequencing Technology in Forensic Science. *Genomics, Proteomics & Bioinformatics,* 12(5), pp. 190-197.

**Appendix A**

## Source Codes

### A-1 mix_dna.py – used to simulate mixtures

```
import argparse
import time
import random
import os
import numpy as np
import pandas as pd
from collections import defaultdict

def basename(path):
    return os.path.splitext(os.path.basename(path))[0]

def restricted_float(x):
    x = float(x)
    if x < 0.0 or x > 1.0:
        raise argparse.ArgumentTypeError("%r should be in range [0.0, 1.0]"%(x,))
    return x

def process(df, part, mixture_loci):
    for row in df[['ID', 'GB:ALLREADS']].itertuples():
        ID = row[1]
        allele_info = row[2].split(',')
        try:
            for allele in allele_info:
                allele_no, read_cnt = allele.split(':')[0], int(allele.split(':')[-1])
                add = len([1 for i in xrange(read_cnt) if random.random() < part])
                if add > 0: mixture_loci[ID][allele_no] += add
        except ValueError:
            pass

def run(input1, input2, part1, part2, output_dir=None):
    start_time = time.time()
    print(time.ctime())
    print
    print "Reading input.."
    df1 = pd.read_table(input1, usecols=['ID', 'GB:ALLREADS',
'ALLELE:ALLELE_FREQ']).dropna(0)
    df2 = pd.read_table(input2, usecols=['ID', 'GB:ALLREADS',
'ALLELE:ALLELE_FREQ']).dropna(0)

    output_df = pd.concat([df1, df2]).drop_duplicates('ID').set_index('ID', drop=False)
    output_df['GB:ALLREADS'] = ''
    mixture_loci = defaultdict(lambda: defaultdict(int))
    print "Generating mixture with",'%.1f' % (part1*100)+"% of
data",basename(input1),"and",'%.1f' % (part2*100)+"% of data",basename(input2)+"."
    print 'Working on data',basename(input1)+'..'
    process(df1, part1, mixture_loci)
    print 'Working on data',basename(input2)+'..'
    process(df2, part2, mixture_loci)

    print 'Finalizing..'
    for locus in mixture_loci:
        output_df.set_value(locus, 'GB:ALLREADS',
','.join([str(allele)+':'+str(mixture_loci[locus][allele]) for allele in mixture_loci[locus]]))

    output_df = output_df[output_df['GB:ALLREADS']!='']

    filename = '_'.join(['mix', basename(input1), basename(input2), 'part', '%.3f' % part1,
'%.3f' % part2]) +'.cts'
    if output_dir is None:
        output_df.to_csv(filename, sep = '\t', index=False)
        filepath = filename
    else:
        if not os.path.exists(output_dir):
            os.makedirs(output_dir)
        output_df.to_csv(output_dir + os.sep + filename, sep = '\t', index=False)
        filepath = output_dir + os.sep + filename

    print
    print("--- %s seconds ---" % (time.time() - start_time))

    return filepath

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Mix DNA samples at specified
proportions.')
    parser.add_argument('input1', help='STR data for DNA sample 1')
    parser.add_argument('input2', help='STR data for DNA sample 2')
    parser.add_argument('part1', type=restricted_float, nargs='?', default=random.random(),
help='the percentage of DNA sample 1 to be mixed [0.0 - 1.0]')
    parser.add_argument('part2', type=restricted_float, nargs='?', default=random.random(),
help='the percentage of DNA sample 2 to be mixed [0.0 - 1.0]')
    parser.add_argument('output_dir', nargs='?', default=None, help='the output directory')
    args = parser.parse_args()
    args.part1, args.part2 = round(args.part1, 3), round(args.part2, 3)

    run(args.input1, args.input2, args.part1, args.part2, args.output_dir)
```

## A-2 MixtureAnalysis.py – python class embodying functions for calculations of LR

```
import sys
import numpy as np
import pandas as pd
import timeit
from collections import Counter, defaultdict
from operator import mul
from itertools import combinations, combinations_with_replacement
from math import factorial, log10

# product of list
def prod(list):
    return 1 if not list else reduce(mul, list)

def calculate_genotype_prob(obs_alleles, mandatory_alleles, allele_freq, nc):
    gt_list = list(combinations_with_replacement(obs_alleles, nc*2 -
len(mandatory_alleles)))
    prob = 0
    for i in gt_list:
        i = list(i)
        i.extend(mandatory_alleles)
        prob += prod([allele_freq[j] for j in i]) * factorial(len(i))/prod([factorial(c) for c in
Counter(i).values()])
    return prob

def mixture_prob(obs_alleles_s, allele_freq_s, nc):
    try:
        allele_freq = {}
        allele_freq.update({allele.split(':')[0] : float(allele.split(':')[-1]) for allele in
allele_freq_s.split(',')})
        obs_alleles = {}
        obs_alleles.update({str(round(float(allele.split(':')[0]), 5)) : int(allele.split(':')[-1]) for
allele in obs_alleles_s.split(',')})

        # if allele frequency for any allele is not found
        if not all(allele in allele_freq.keys() for allele in obs_alleles):
            return np.nan

        cprob = 0
        p = 0.95
        q = 0.05
        for i in xrange(1, min(len(obs_alleles), nc*2)+1):
            for alleles in combinations(obs_alleles, i):
                error_correction = prod([1 - q**obs_alleles[a] if a in alleles else
q**obs_alleles[a] for a in obs_alleles])
```

```
                cprob += error_correction * calculate_genotype_prob(alleles, alleles, allele_freq, nc)
    except (AttributeError, ValueError):
        return np.nan

    return cprob

def calc_posterior_prob(obs_alleles_s, hypothesis_alleles_list, allele_freq_s, nc_total):
    try:
        allele_freq = {}
        allele_freq.update({allele.split(':')[0] : float(allele.split(':')[-1]) for allele in allele_freq_s.split(',')})
        obs_alleles = {}
        obs_alleles.update({str(round(float(allele.split(':')[0]), 5)) : int(allele.split(':')[-1]) for allele in obs_alleles_s.split(',')})

        # if allele frequency for any allele is not found
        if not all(allele in allele_freq.keys() for allele in obs_alleles):
            return np.nan

        h_alleles_all = defaultdict(int)
        for h_alleles_s in hypothesis_alleles_list:
            for allele in h_alleles_s.split(','):
                allele_no, read_cnt = str(round(float(allele.split(':')[0]), 5)), int(allele.split(':')[-1])
                h_alleles_all[allele_no] += read_cnt

        nc_known = len(hypothesis_alleles_list)
        nc_unknown = nc_total-nc_known
        cprob = 0
        p = 0.95
        q = 0.05
        for i in xrange(1, min(len(obs_alleles), nc_total*2)+1):
            for alleles in combinations(obs_alleles, i):
                alleles = set(alleles)
                alleles_mutual = alleles & set(h_alleles_all)
                if len(alleles_mutual) < max(0, i-nc_unknown*2):
                    continue

                error_initial = prod([1 - q**obs_alleles[a] if a in alleles else q**obs_alleles[a] for a in obs_alleles])
                error_initial *= prod([q**h_alleles_all[a] for a in h_alleles_all if a not in alleles])
                for j in xrange(max(0, i-nc_unknown*2), len(alleles_mutual)+1):
                    for known_alleles in combinations(alleles_mutual, j):
                        error_new = error_initial * prod([1 - q**h_alleles_all[a] if a in
known_alleles else q**h_alleles_all[a] for a in alleles_mutual])
                        if nc_unknown > 0:
                            cprob += error_new * calculate_genotype_prob(alleles, alleles-
set(known_alleles), allele_freq, nc_unknown)
                        elif i-j == 0:
                            cprob += error_new
    except (AttributeError, ValueError):
        return np.nan

    return cprob if cprob != 0 else 0.05**(sum(obs_alleles.values()) +
sum(h_alleles_all.values()))
```

### A-3 find_nc.py – used to find out number of contributors

```
import argparse
import time
import numpy as np
import pandas as pd
import MixtureAnalysis as ma
from math import log10

def run(input, N):
    start_time = time.time()
    print(time.ctime())

    df = pd.read_table(input, usecols=['ID', 'GB:ALLREADS',
'ALLELE:ALLELE_FREQ'])
    df.rename(columns={'ID': 'Id'}, inplace=True)

    prob_lists = [[ma.mixture_prob(row[2], row[3], nc+1) for row in df.itertuples()] for nc
in xrange(N)]
    for i in xrange(N):
        df['P('+str(i+1)+')']=prob_lists[i]

    df.dropna(0, inplace=True)
    print df.tail()
    print
    for i in xrange(N):
        print "The probability of having",i+1,"contributor(s)
is:",np.log10(df['P('+str(i+1)+')']).sum(),"(Log 10)"

    print
    df.ix[:, df.columns != 'ALLELE:ALLELE_FREQ'].to_csv(input + '_nc' +
time.strftime("%Y%m%d%H%M%S") + '.tsv', sep = '\t', index=False)
    print("--- %s seconds ---" % (time.time() - start_time))

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Calculate the probability of the number
of contributors in a DNA mixture.')
    parser.add_argument('input', help='STR data for DNA mixture')
    parser.add_argument('N', type=int, nargs='?', default=2, help='the number of
contributors to calculate up to')
    args = parser.parse_args()

    run(args.input, args.N)
```

**Appendix A**

## A-4 calc_lr.py – used to calculate likelihood ratios

```
import argparse
import time
import os
import numpy as np
import pandas as pd
import MixtureAnalysis as ma
from math import log10

def basename(path):
    return os.path.splitext(os.path.basename(path))[0]

def run(M, Hp, Hd, N, do_output):
    start_time = time.time()
    print(time.ctime())
    print
    print "DNA Mixture:",basename(M)
    print "Hp contains:",[basename(p) for p in Hp]
    print "Hd contains:",[basename(p) for p in Hd]
    Hp_n, Hd_n = len(Hp), len(Hd)

    if N == -1:
        N = max(Hp_n, Hd_n)
    print "Number of contributors:",N

    main_df = pd.read_table(M, usecols=['ID', 'GB:ALLREADS',
'ALLELE:ALLELE_FREQ'])
    main_df.rename(columns={'GB:ALLREADS': M}, inplace=True)

    for strfile in Hp:
        df = pd.read_table(strfile, usecols=['ID', 'GB:ALLREADS'])
        df.rename(columns={'GB:ALLREADS': strfile+'_Hp'}, inplace=True)
        main_df = pd.merge(main_df, df, on='ID', how='inner')

    for strfile in Hd:
        df = pd.read_table(strfile, usecols=['ID', 'GB:ALLREADS'])
        df.rename(columns={'GB:ALLREADS': strfile+'_Hd'}, inplace=True)
        main_df = pd.merge(main_df, df, on='ID', how='inner')

    main_df.rename(columns={'ID': 'Id'}, inplace=True)
    cols = main_df.columns.tolist()
    cols.append(cols.pop(cols.index('ALLELE:ALLELE_FREQ')))
    main_df = main_df.reindex(columns=cols)

    Lh_Hp_L, Lh_Hd_L, Log10_LR = [], [], []
    for row in main_df.itertuples():
        Lh_Hp = ma.calc_posterior_prob(row[2], [row[3+i] for i in xrange(Hp_n)], row[3+Hp_n+Hd_n], N)
        Lh_Hd = ma.calc_posterior_prob(row[2], [row[3+Hp_n+i] for i in xrange(Hd_n)], row[3+Hp_n+Hd_n], N)
        Lh_Hp_L.append(Lh_Hp)
        Lh_Hd_L.append(Lh_Hd)
        Log10_LR.append(log10(Lh_Hp/Lh_Hd))

    row_cnt = len(main_df.index)
    main_df['Lh Hp'] = Lh_Hp_L
    main_df['Lh Hd'] = Lh_Hd_L
    main_df['Log(10, LR)'] = Log10_LR
    main_df.dropna(0, inplace=True)
    main_df['Log(10, cumulative-LR)'] = main_df['Log(10, LR)'].cumsum()
    cumu_LR = main_df['Log(10, cumulative-LR)'].iloc[-1]

    print "Dropped",(row_cnt - len(main_df.index)),"out of",row_cnt,"rows due to data errors.\n"
    print "Log(10, cumulative-LR):",cumu_LR
    if cumu_LR > 0:
        print "It is more probable that Hp is True.\n"
    else:
        print "It is more probable that Hd is True.\n"

    filename = '_'.join([basename(M), 'Hp', '_'.join(basename(f) for f in Hp), 'Hd', '_'.join(basename(f) for f in Hd), 'N', str(N)]) + '.tsv'
    if do_output:
        main_df.ix[:, main_df.columns != 'ALLELE:ALLELE_FREQ'].to_csv(filename, sep = '\t', index=False)
    print("--- %s seconds ---" % (time.time() - start_time))
    return {'filename': filename, 'cumu_LR': cumu_LR}

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Calculate the likelihood ratio for the specified DNA mixture.')
    parser.add_argument('M', help='STR data for evidence mixture')
    parser.add_argument('-Hp', nargs='+', default=[], help='STR data for individuals in Hp')
    parser.add_argument('-Hd', nargs='+', default=[], help='STR data for individuals in Hd')
    parser.add_argument('-N', type=int, default=-1, help='the number of contributors')
    parser.add_argument('--do_output', action='store_true', help='set this flag to output generated data files')
    args = parser.parse_args()

    run(args.M, args.Hp, args.Hd, args.N, args.do_output)
```

### A-5 test.py – use to simulate experiments for empirical analysis

```
import argparse
import mix_dna
import calc_lr
import glob
import os
import gc
import random
import time
import numpy as np
import pandas as pd
from threading import Thread
from itertools import combinations

def basename(path):
    return os.path.splitext(os.path.basename(path))[0]

def restricted_float(x):
    x = float(x)
    if x < 0.0 or x > 1.0:
        raise argparse.ArgumentTypeError('%r should be in range [0.0, 1.0]'%(x,))
    return x

def run(data_in_dir, data_out_dir, format, do_output, result_output_name, parts):
    # Open result output file to append data
    if os.path.isfile(result_output_name):
        df = pd.read_table(result_output_name)
    else:
        df = pd.DataFrame(columns=['M', 'part1', 'part2', 'part3', 'Hp', 'Hd', 'N', 'cumuLR',
'result'])
    df['N']=df['N'].astype(np.int16)

    if do_output and not os.path.exists(data_out_dir):
        os.makedirs(data_out_dir)

    # Get data input files
    cts_files = glob.glob(data_in_dir + os.sep + '*.' + format)

    comb = list(combinations(cts_files, 2)) + list(combinations(cts_files, 3))
    i=1

    for combination_list in comb:
        print 'Working on ' + str(i) + '/' + str(len(comb)) + ' test runs.'
        print
        i+=1
```

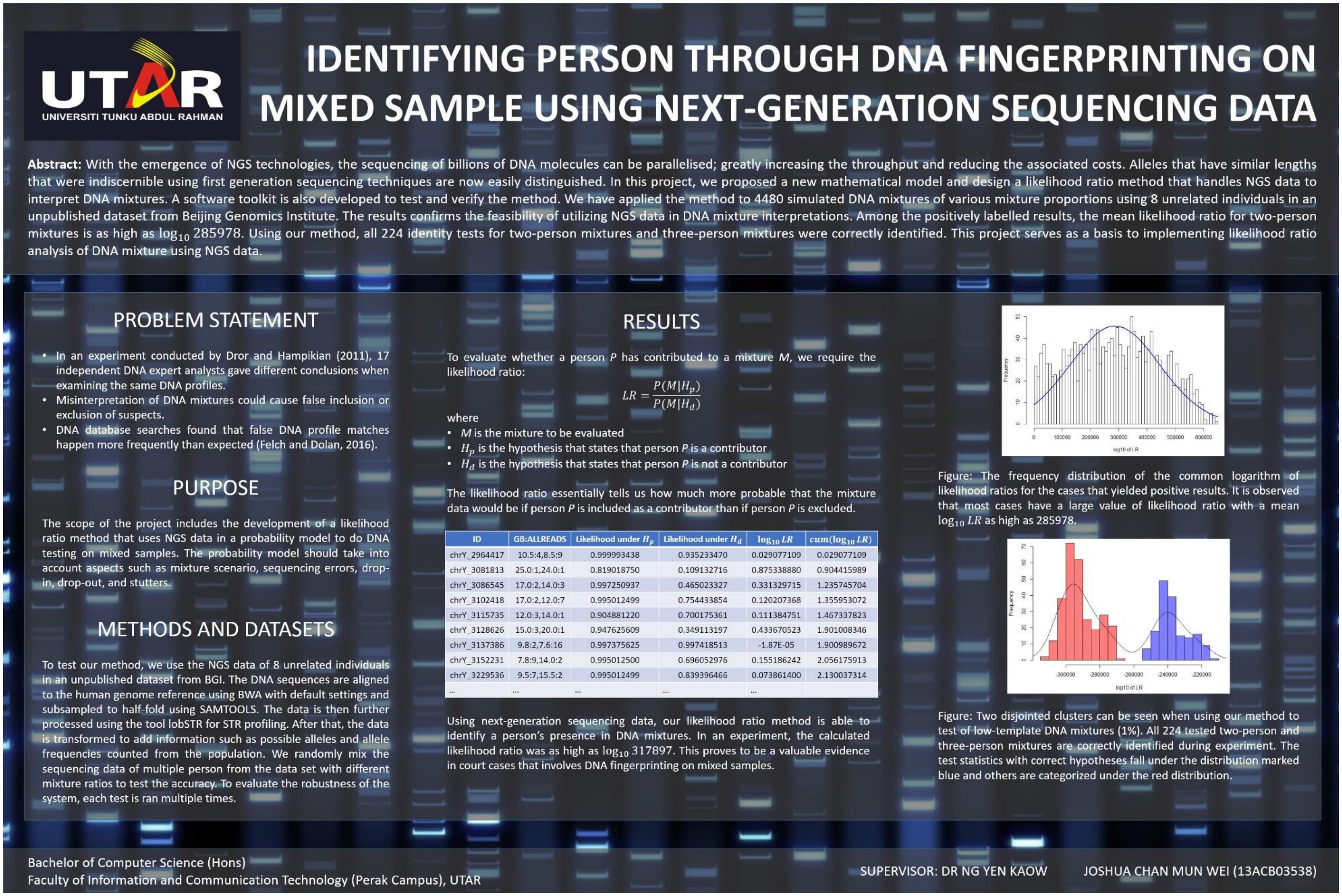
**Appendix A**

```
        timestr = time.strftime('_%Y%m%d-%H%M%S')
        if len(combination_list) == 2:
            pairs = combination_list
            row = {}
            fn = mix_dna.run(pairs[0], pairs[1], parts[0], parts[1], data_out_dir)
            filename = 'mix2_' + '_'.join(basename(f) for f in pairs) + '_part_' + '_'.join('%.3f' %
(p) for p in parts[:2]) + timestr + '.cts'
            N = 2
        else:
            triplets = combination_list
            row = {}
            ft = mix_dna.run(triplets[0], triplets[1], parts[0], parts[1], data_out_dir)
            fn = mix_dna.run(ft, triplets[2], 1, parts[2], data_out_dir)
            filename = 'mix3_' + '_'.join(basename(f) for f in triplets) + '_part_' +
'_'.join('%.3f' % (p) for p in parts) + timestr + '.cts'
            os.remove(ft)
            N = 3

        os.rename(fn, data_out_dir + os.sep + filename)
        gc.collect()
        for p in cts_files:
            Hp = [p]
            Hd = []
            calc_lr_res = calc_lr.run(data_out_dir + os.sep + filename, Hp, Hd, N, do_output)
            if do_output:
                os.rename(calc_lr_res['filename'], data_out_dir + os.sep +
calc_lr_res['filename'])
            cumu_LR = calc_lr_res['cumu_LR']
            row['M'] = filename
            row['part1'] = '%.3f' % parts[0]
            row['part2'] = '%.3f' % parts[1]
            row['part3'] = '%.3f' % parts[2] if N == 3 else ''
            row['Hp'] = ','.join(basename(f) for f in Hp)
            row['Hd'] = ','.join(basename(f) for f in Hd)
            row['N'] = N
            row['cumuLR'] = cumu_LR
            row['result'] = 'T' if cumu_LR > 0 else 'F'
            df = df.append(pd.DataFrame.from_records([row], columns=df.columns))
            # do this in a thread to prevent KeyboardInterrupt from ruining everything
            a = Thread(target=df.to_csv(result_output_name, sep='\t', index=False))
            a.start()
            a.join()
            gc.collect()

        if not do_output:
            os.remove(data_out_dir + os.sep + filename)
```

**Appendix A**

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Script to do experiments on the LR method.')
    parser.add_argument('--data_in_dir', default='data', help='directory path to input data files, defaults to data')
    parser.add_argument('--data_out_dir', default='output_data', help='directory path to output data files, defaults to output_data')
    parser.add_argument('--format', default='cts', help='format of data files, defaults to cts')
    parser.add_argument('--do_output', action='store_true', help='set this flag to output generated data files')
    parser.add_argument('--result_output_filename', default='results.tsv', help='output file name for test results, defaults to results.tsv')
    parser.add_argument('-p1', type=restricted_float, default=random.random(), help='the percentage of DNA from person 1 to be added into the mixture [0.0 - 1.0]')
    parser.add_argument('-p2', type=restricted_float, default=random.random(), help='the percentage of DNA from person 2 to be added into the mixture [0.0 - 1.0]')
    parser.add_argument('-p3', type=restricted_float, default=random.random(), help='the percentage of DNA from person 3 to be added into the mixture [0.0 - 1.0]')
    args = parser.parse_args()

    parts = [args.p1, args.p2, args.p3]
    run(args.data_in_dir, args.data_out_dir, args.format, args.do_output, args.result_output_filename, parts)
```

# Appendix B



UTAR UNIVERSITI TUNKU ABDUL RAHMAN

## IDENTIFYING PERSON THROUGH DNA FINGERPRINTING ON MIXED SAMPLE USING NEXT-GENERATION SEQUENCING DATA

**Abstract:** With the emergence of NGS technologies, the sequencing of billions of DNA molecules can be parallelised; greatly increasing the throughput and reducing the associated costs. Alleles that have similar lengths that were indiscernible using first generation sequencing techniques are now easily distinguished. In this project, we proposed a new mathematical model and design a likelihood ratio method that handles NGS data to interpret DNA mixtures. A software toolkit is also developed to test and verify the method. We have applied the method to 4480 simulated DNA mixtures of various mixture proportions using 8 unrelated individuals in an unpublished dataset from Beijing Genomics Institute. The results confirms the feasibility of utilizing NGS data in DNA mixture interpretations. Among the positively labelled results, the mean likelihood ratio for two-person mixtures is as high as $\log_{10} 285978$. Using our method, all 224 identity tests for two-person mixtures and three-person mixtures were correctly identified. This project serves as a basis to implementing likelihood ratio analysis of DNA mixture using NGS data.

### PROBLEM STATEMENT

- In an experiment conducted by Dror and Hampikian (2011), 17 independent DNA expert analysts gave different conclusions when examining the same DNA profiles.
- Misinterpretation of DNA mixtures could cause false inclusion or exclusion of suspects.
- DNA database searches found that false DNA profile matches happen more frequently than expected (Felch and Dolan, 2016).

### PURPOSE

The scope of the project includes the development of a likelihood ratio method that uses NGS data in a probability model to do DNA testing on mixed samples. The probability model should take into account aspects such as mixture scenario, sequencing errors, drop-in, drop-out, and stutters.

### METHODS AND DATASETS

To test our method, we use the NGS data of 8 unrelated individuals in an unpublished dataset from BGI. The DNA sequences are aligned to the human genome reference using BWA with default settings and subsampled to half-fold using SAMTOOLS. The data is then further processed using the tool lobSTR for STR profiling. After that, the data is transformed to add information such as possible alleles and allele frequencies counted from the population. We randomly mix the sequencing data of multiple person from the data set with different mixture ratios to test the accuracy. To evaluate the robustness of the system, each test is ran multiple times.

### RESULTS

To evaluate whether a person *P* has contributed to a mixture *M*, we require the likelihood ratio:

$$LR = \frac{P(M|H_p)}{P(M|H_d)}$$

where

- *M* is the mixture to be evaluated
- $H_p$ is the hypothesis that states that person *P* is a contributor
- $H_d$ is the hypothesis that states that person *P* is not a contributor

The likelihood ratio essentially tells us how much more probable that the mixture data would be if person *P* is included as a contributor than if person *P* is excluded.

| ID | GB:ALLREADS | Likelihood under $H_p$ | Likelihood under $H_d$ | $\log_{10} LR$ | $cum(\log_{10} LR)$ |
|---|---|---|---|---|---|
| chrY_2964417 | 10.5:4,8.5:9 | 0.999993438 | 0.935233470 | 0.029077109 | 0.029077109 |
| chrY_3081813 | 25.0:1,24.0:1 | 0.819018750 | 0.109132716 | 0.875338880 | 0.904415989 |
| chrY_3086545 | 17.0:2,14.0:3 | 0.997250937 | 0.465023327 | 0.331329715 | 1.235745704 |
| chrY_3102418 | 17.0:2,12.0:7 | 0.995012499 | 0.754433854 | 0.120207368 | 1.355953072 |
| chrY_3115735 | 12.0:3,14.0:1 | 0.904881220 | 0.700175361 | 0.111384751 | 1.467337823 |
| chrY_3128626 | 15.0:3,20.0:1 | 0.947625609 | 0.349113197 | 0.433670523 | 1.901008346 |
| chrY_3137386 | 9.8:2,7.6:16 | 0.997375625 | 0.997418513 | -1.87E-05 | 1.900989672 |
| chrY_3152231 | 7.8:9,14.0:2 | 0.995012500 | 0.696052976 | 0.155186242 | 2.056175913 |
| chrY_3229536 | 9.5:7,15.5:2 | 0.995012499 | 0.839396466 | 0.073861400 | 2.130037314 |
| ... | ... | ... | ... | ... | |

Using next-generation sequencing data, our likelihood ratio method is able to identify a person's presence in DNA mixtures. In an experiment, the calculated likelihood ratio was as high as $\log_{10} 317897$. This proves to be a valuable evidence in court cases that involves DNA fingerprinting on mixed samples.

Figure: The frequency distribution of the common logarithm of likelihood ratios for the cases that yielded positive results. It is observed that most cases have a large value of likelihood ratio with a mean $\log_{10} LR$ as high as 285978.

Figure: Two disjointed clusters can be seen when using our method to test of low-template DNA mixtures (1%). All 224 tested two-person and three-person mixtures are correctly identified during experiment. The test statistics with correct hypotheses fall under the distribution marked blue and others are categorized under the red distribution.

Bachelor of Computer Science (Hons)
Faculty of Information and Communication Technology (Perak Campus), UTAR

SUPERVISOR: DR NG YEN KAOW

JOSHUA CHAN MUN WEI (13ACB03538)