Norges miljø- og
biovitenskapelige
universitet

Master Thesis 2016 60 ECTS
Norwegian University of Life Sciences
Department of Chemistry, Biotechnology and Food Science

Statistical Methods for Match Probabilities with Applications to Y Chromosome Data

Daniel Lanes
Applied Statistics

Abstract

Accurate estimates of Y-STR haplotype frequencies is an interesting problem in itself, but is especially important in forensic genetics, where the frequencies are used to calculate the likelihood ratio (LR) for the evidential weight of a DNA profile found at a crime scene.

In this thesis, four methods for Y-STR haplotype frequency estimation are compared with respect to accuracy and bias. This is performed on data from simulated from Wright-Fisher populations with empirical mutation rates and different sampling factors like sample size and the number of markers comprising the haplotypes. Three of the four methods are count based methods (CBMs) and the last method is an allele based method (ABM), defined by if the method represent Y-STR haplotypes as indecomposable objects or not.

The first method, named the Count Method (CM) is derived from the empirical frequency of the haplotype. The next two CBMs, the Kappa Model (KM) and the Good-Turing estimator (GTE) are based on the proportion of haplotypes observed a particular number of times in the sample. Last, the Discrete Laplace Method (DLM) identifies subpopulation centers by clustering and models allele frequencies at each loci with discrete Laplace distributions. Hapltotype frequency estimates are then obtained by multiplying estimated allele frequencies across loci.

The CBMs underestimated the LR in all scenarios. The DLM have the highest mean accuracy in general, but also have more variance and a tendency to overestimate the LR slightly when haplotypes are composed of more markers. More research into which factors significantly affect haplotype frequency estimates is encouraged.

Sammendrag

Nøyaktige estimater for Y-STR haplotype frekvenser er et interessant problem i seg selv, men er spesielt viktig innen rettsgenetikk, hvor frekvensene brukes til å beregne 'likelihood ratio' (LR) for bevisstyrken til en DNA-profil funnet ved ett åsted.

I denne avhandlingen presenterer fire metoder for beregning av Y-STR haplotype frekvenser, disse sammenlignes med hensyn på nøyaktighet og bias. Dette gjennomføres på data fra simulerte Wright-Fisher populasjoner med empiriske mutasjonsrater og under forskjellige utvalgsfaktorer som utvalgsstørrelse og antall markører haplotypene består av. Tre av de fire metodene er 'tellebaserte' og den siste er 'allelbasert', definert som om metodene representerer Y-STR haplotypene som udekomponerbare enheter eller ikke.

Den første metoden, kalt 'the Count Method' (CM) er utledet basert på den empiriske frekvensen til haplotypen som undersøkes. De neste to metodene (tellebaserte), 'the Kappa Model' (KM) og 'the Good-Turing estimator' (GTE) er basert på andelen haplotyper som er observert ett gitt antall ganger i utvalget. Den siste metoden, 'the Discrete Laplace Model' (DLM) identifiserer subpopulasjonssentere ved samle lignende haplotyper i klynger og deretter modellere allelfrekvenser ved hver loci med discrete Laplace fordelinger. Haplotypefrekvens estimatene beregnes deretter ved å multiplisere sammen estimerte allelfrekvenser over alle loci.

De tellebaserte metodene underestimerte LR i alle scenarioene. DLM har høyest gjennomsnittlig nøyaktighet genrerelt, men hadde også mer varians og en tendens til å overestimere LR i en liten grad, når haplotypene består av flere markører. Mer forskning på hvilke faktorer som signifikant påvirker haplotypefrekvens estimatene oppfordres.

Acknowledgements

The primary reason for including an acknowledgements section at all, is to honour my excellent supervisor Professor Thore Egeland. Thore has been helpful and always providing great guidance when needed, thank you!

I also want to thank the Biostatistics group at the Norwegian University of Life Sciences (NMBU) for many interesting group meetings and good cake.

Also, a big thanks is in order for my parents, family and friends for support throughout the thesis.

Daniel Lanes
Risor, May 2016

Contents

Chapter 1

Introduction

1.1 Problem

Modern forensic science has progressed rapidly after DNA typing was developed by Sir Alec Jeffreys around 1985. Continuing advances in both typing kits and computing power give more certainty to identification of individuals. The main principle behind DNA typing is to assess the relationship between two DNA profiles. A common case is paternity cases, where a DNA profile of an alleged father is compared with a DNA profile from the child in question. Further, in many such cases, which in general is termed relationship inference, the inference is dependent on the distribution of the DNA profiles in the population. Thus, this field is dependent on statistics to make inferences. In this thesis I will restrict myself to a special case of relationship inference, where the problem is to assess how likely a DNA profile from a suspect matches that on a given crime scene. A further restriction is that I will only focus on the male population. The central part of the thesis will be centred around the statistical aspects of this field, but a fair bit of genetics is needed to understand the difficulty with regards to modelling the distributions needed for inference. I assume some familiarity with genetics, but I will briefly elucidate the most central concepts when needed during the thesis, mostly during the introductory part. The statistical aspects of the thesis, which is the major point of inquiry, will be the focal point after some context has been established.

DNA profiles from the Y-chromosome is inherited in a patrilineal fashion, that is, directly inherited from fathers to sons. So, throughout the thesis we do not consider the small part of the Y chromosome that may recombine. A set of genetic markers with the property of direct inheritance from either parent is termed lineage markers. The usefulness of lineage markers are common practice in modern genealogy, but the forensic science applications are also of great interest. Consider the case where a Y-chromosomal DNA sample is found at some crime scene and that the sample matches some suspect, then what is the probability of the crime scene sample matching a person unconnected to the crime scene? This is an important question to answer, since the evidence should be evaluated according to at least one alternative hypothesis. It would not be fair to convict the suspect based on this match alone. Lets go back to the most recent question. Intuitively we can see that, if the crime scene type is common in the general population (say 0.5 of population has the type), then it is not unlikely to expect a match with an individual drawn at random. In the converse scenario, a match is very unlikely if almost all individuals carries unique types. This illustrates the importance of having a good estimate for the rarity of a type, thereby having a good estimate for a coincidental match between a crime scene type and a arbitrary male. The problem of obtaining such estimate is called the evidential value of Y-STR haplotype match probability by Charles H. Brenner. The problem has been investigated by several people in the last decade (Brenner, 2013). This kind of DNA evidence can be used in special cases where autosomal DNA is unavailable or where the inference spans across several generations within a family tree. It can also be used in conjunction with nuclear DNA to gain more evidential strength. Before formalizing the problem, a small amount of context is needed, both with respect to genetics and to the practice of quantifying evidence and principles for making inferences in forensic science.

1.2 Forensic Genetics

This section contains a brief introduction to the most central concepts in forensic genetics. The purpose of this section is to clarify the forensic genetics terminology I will use. In addition I find it useful to also summarize the main principles behind relationship inference to provide further context. I. Evett and B. Weir formulated three principles for evaluation DNA evidence (Egeland et al, 2016):

1. To evaluate the uncertainty of any given proposition it is necessary to consider at least one alternative proposition.
2. Scientific interpretation is based on question of the following kind: What is the probability of the evidence given the proposition?

3. Scientific evidence is conditioned not only on the competing propositions, but also by the framework of circumstances within which they are to be evaluated.

In principle, the number of possible alternative propositions are limited only by imagination, but in practice (due to the third principle) it often suffices to state two possible explanations. For example, a suspect S is the culprit in some rape case or S is not. However, if the brother of S is previously convicted for a similar crime, then it is wise to consider him as a possible suspect as well. I will restrict myself to the basis case where the competing explanations are, first, suspect S is the source of crime scene DNA profile, against the alternative case that some other arbitrary person is the source of the DNA profile. So, we condition on a possible relationship between a DNA sample and a particular person, then compute the probability of observing the DNA sample given this relationship. Consider the following silly example to get a sense of how this process is done. Assume you submit a DNA profile to a company that provides paternity tests for a small fee. They have also obtained a DNA profile from former North-Korean dictator Kim Jong-Il. The probability for observing your particular DNA profile, when conditioning on ”Kim Jong-Il is your father”, is very likely to be close to zero. This is agreement with the second principle. To rephrase, I restrict myself to the two following hypotheses throughout the thesis, consistently denoted H_1 and H_2, expressed as:

H_1: Some suspect S is the source of the crime scene DNA profile.

H_2: Some other arbitrary male (not connected to crime) is the source of the crime scene DNA profile.

A general term for a random match between a particular DNA profile to an identical one in the population under consideration is called the random match probability (RMP) (Egeland et al, 2016). In a random draw from a set of k objects with equal frequency, the RMP for matching a particular one is $\frac{1}{k}$. However, this changes if the count of the various objects are not uniform.

1.2.1 Likelihood Ratio

The strength of different hypotheses relative to each other can be expressed as a likelihood ratio (LR)

$$LR = \frac{Pr(D|H_1)}{Pr(D|H_2)}$$

where D is our data. Assuming we have we have k hypotheses, , some prior probabilities $Pr(H_i)$ and and some genotype data, then application of Bayes theorem gives us the following expression for what is called the posterior probability:

$$Pr(H_i|D) = \frac{Pr(D|H_i)Pr(H_i)}{\sum_{j=1}^{k} Pr(D|H_j)Pr(H_j)}$$

By taking the ratio of the posterior probability of two hypotheses H_i and H_j where $i \neq j$, we get what is called the posterior odds (Egeland et al, 2016):

$$\frac{Pr(H_i|D)}{Pr(H_j|D)} = LR * \frac{Pr(H_i)}{Pr(H_j)} = \frac{Pr(D|H_i)}{Pr(D|H_j)} * \frac{Pr(H_i)}{Pr(H_j)}$$

The sums in the denominators of the expression for posterior probability cancels out. Now the posterior odds is a product between the LR and the prior odds. The LR takes values in $[0, \infty)$. Saying that the data is LR

times more likely given hypothesis i relative to hypothesis j is a reasonable way to interpret the result. In some judicial cases, there are requests for representing the evidence in terms of probabilities to the different hypothesis rather than likelihood ratios. It is sometimes possible to transform the LR's into a posterior probability known as the Essen-Moller index W. Assume that we only have two mutually exclusive hypotheses i, j with equal prior probabilities, then the posterior probability for hypothesis i can be expressed as

$$W = Pr(H_i|D) = \frac{LR}{LR+1}$$

For the derivation of general case with k hypotheses, see (Egeland et al, 2016).

1.2.2 Forensic Markers and Populations

DNA profiles used in forensic genetics are obviously composed of a very small portion of a persons total DNA. This portion again is often composed of small sequences spread across many chromosomes, as the current purpose of a forensic DNA profile is discriminatory power, not phenotype categorization. Same sex humans are approximately 0.99 identical, most of us carries the same genes, but we have different variants of these genes in various locations on the chromosome. In forensic genetics settings, we call these selected locations along the genome forensic markers or loci. Some markers are usually highly polymorphic, with number of different marker variants are in the range of 6 to 70 (Egeland et al, 2016). These differing variants are referred to as alleles. In autosomal DNA, in contrast with allosome (sex chromosome), there are two allele variants at each marker, one inherited from each parent.

When both alleles are identical at a marker (see TH01 marker in Table 1.1 as example), we call the genotype homozygous, and heterozygous conversely. Markers on the autosomes(and possibly allosomes) are assumed to follow an important population genetic property known as the Hardy-Weinberg Equilibrium (HWE), first proven mathematically by G. H. Hardy and Wilhelm Weinberg. HWE states that the allele frequency distribution is at equilibrium and is not disturbed by effects like non-random mating, population substructure, ... , and natural selection (Egeland et al, 2016). Assume that a population is in HWE and with population frequencies p_i and p_j for a autosomal two allele marker M. Then the probabilities for observing the following genotypes G in an arbitrary individual is

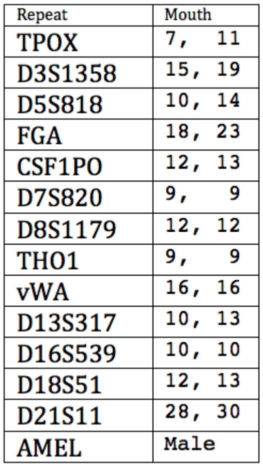
$$Pr(G = i/j) = 2p_ip_j$$

$$Pr(G = i/i) = p_i^2$$

The factor 2 comes from the fact that the genotypes $G_1 = i/j$ and $G_2 = j/i$ is considered equivalent (Egeland et al, 2016). Also,

$$Pr(G = i/i \cup G = i/j \cup G = j/j) = p_i^2 + 2p_ip_j + p_j^2 = 1$$

This can also be modelled with a multinomial distribution with probability parameter $(p_i^2, 2p_i(1 - p_i), (1 - p_i)^2)$.



Repeat	Mouth
TPOX	7, 11
D3S1358	15, 19
D5S818	10, 14
FGA	18, 23
CSF1PO	12, 13
D7S820	9, 9
D8S1179	12, 12
THO1	9, 9
vWA	16, 16
D13S317	10, 13
D16S539	10, 10
D18S51	12, 13
D21S11	28, 30
AMEL	Male

Table 1.1. Example of a set of 13 STR markers and one marker commonly used for sex determination (AMEL).

In contrast with autosomal DNA, the alleles at the Y chromosomal markers are inherited directly from fathers to sons, and only has one allele per marker. This means that Y forensic markers is what is called linked markers. One could imagine the markers in Table 1.1 to be markers on the Y-chromosome, but at each markers there is only one number. This direct linked inheritance means that the mechanisms for how Y-chromosomal alleles are distributed in the population differ from autosomal alleles (Brenner, 2013).

Population substructure

One of the major problems in developing good models for estimating Y haplotype frequency, is the problem of population substructure. This problem is especially prevalent in frequency estimation based on lineage markers and is addressed in more detail in section 2.1. Modelling population substructure, also known as population stratification (PS) is a major field of research in itself. The goal of such models is to quantify differences in allele frequencies in subpopulations. By applying this process, it is possible to gain knowledge about the phylogeny of the different subpopulations or clusters within the population. Simple models for dealing with possible subpopulation structure during computations for forensic inference typically entails introducing a parameter θ, defined on $[0, 1]$. θ aims to correct for the possible substructure (deviation for HWE) in the population (Egeland et al, 2016). A synonymous statistic with θ, is the Fixation index F_{st}, derived from the Fisher F-statistic, one of the most frequent used statistics in population genetics. The definition is commonly defined by the variance of allele frequencies at a loci between two populations A and B and the average allele frequency $\bar{p}$ at the same loci in the total population comprised of A and B. It is also possible to define it by probability of identity by decent (wiki, Fixation Index). In general it is defined as

$$F_{ST} = \frac{\sigma_S^2}{\sigma_T^2}$$

The extreme value $F_{st} = 0$ means that there are no difference in population structure (allele variation at a loci) between A or B and the total population consisting of A and B combined. Conversely, $F_{st} = 1$ means that the population structure of A or B differs completely from the combined population A and B. In this case, the combined population A and B does not have a uniform structure. It is possible to estimate both the between population variance of an allele σ_S^2 and the total population variance σ_T^2 from the data, in a similar way to how it is done in ANOVA.

In the paper 'Texas Population Substructure and Its Impact on Estimating the Rarity of Y STR Haplotypes from DNA Evidence', the researchers pooled three populations (African American, Caucasian, and Hispanic) into one population, then computed F_{ST} between the full population and one subpopulation at the time. Further, by treating each k Y-STR markers as one allele, they computed the F_{ST} for $k = 1, \ldots, 16$. As we shall see more in section 2.1, there is a clear trend of F_{ST} values to be lower when introducing more Y-STR markers (Budowle et al, 2009). Smaller F_{ST} indicates a reduction in population substructure. This can potentially be exploited to increase the accuracy of models that do not incorporate lineage information into the haplotype estimate.

1.2.3 YHRD database

The Y chromosome haplotype reference database (YHRD) is an open access database of population samples in the form of typed Y chromosome sequences, also called Y haplotypes. According to their website yhrd.org, the project has two main objectives. First, generating reliable estimates for Y-STR haplotype frequencies, in order to compute match probabilities in forensic and kinship cases. Second, assessing male population stratification on a macroscopic scale by using Y-STR and Y-SNP frequency distributions (YHRD, 2015). The first objective is exactly the same problem this thesis is trying to gain some further knowledge on. Although, the latter problem may be of interest to gain more precise estimates for computing match probabilities in forensic cases. Here is a summary over the number of different haplotypes for various number of loci in the YHRD, counted July 2015.

Number of loci	n	Typing kit
9	≈154 000	PowerPlex
17	102 700	Yfiler
23	26 100	PowerPlex Y23
27	19 200	Yfiler

Table 2. Approximate number of haplotypes contained in YHRD, provided the following typing kits.

YHRD has rigorous database sample submission protocols to exclude bad Y-SNP/Y-STR markers. The population samples are submitted by individual laboratories and institutes, then further checked before entering the database. So far YHRD has collected population samples from 129 different countries from 250 different institutes and laboratories (YHRD, 2015).
YHRD offers various tools for gathering information from the database like haplotype search, kinship inference, analysis of molecular variance (AMOVA), and mixture analysis (YHRD, 2015).
YHRD also has a comprehensive list of papers that reports their findings about the submitted population samples.

The phylogeny of the Y chromosome can be mapped by the fact that the Y chromosome is inherited patrilineally without recombination and that all males are ancestors of a Y chromosomal adam. By careful tracking of random mutations along Y chromosomal markers, it is possible to cluster haplotypes into haplogroups and map them to phylogenetic trees (YHRD, 2016). The figure below gives some idea of how population substructure emerges and how populations can remain rather homogeneous.

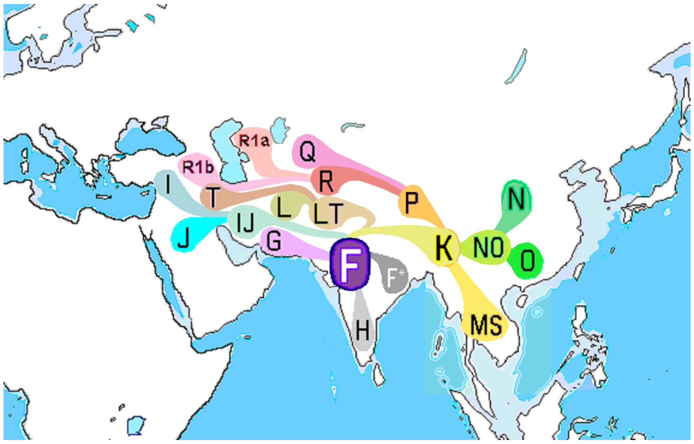


Figure 1. Migration of haplotypes from haplogroup F results in new haplogroups over time. (Source: Wikipedia / Human Y-chromosome DNA haplogroup)

1.3 Formal problem description

The problem I want to address is how can we get an accurate and conservative estimate of the frequency of an previously unseen haplotype. I will address this by using different estimators. More specifically, I want to evaluate how well methods that only uses the count of the different haplotypes perform against methods that incorporate lineage information to improve estimates. My goal is to see how these estimators perform under different degrees of sample size and also when the haplotypes are comprised of different number of loci.

More precisely my first goal is to estimate

$$F_x = Pr(Match|H_2)$$

where the hypothesis H_2 is the same as described in section 1.2 by different methods and compare them with respect to defined estimator properties like MSE. It is also assumed that $\mathbf{h_x}$ is previously unseen, this means that it is not contained in our current data/database. Let the potential different estimates of F_x be indexed by $i = 1, \ldots, k$ So the expression $\hat{F_x}^{(i)}$ denotes an estimate from a particular method i, while $\hat{F_x}$ denotes an arbitrary one. So, for a haplotype $\mathbf{h}_y$, the population frequency is F_y and it's estimated frequency by estimator i is $\hat{F_y}^{(i)}$.
The estimates

$$\hat{F_x}^{(i)}$$

are estimated from haplotypes comprised of different number of loci and with different degree of sample size and population coverage. Population coverage is defined as $\frac{N_s}{N}$, where N_s is the number of distinct haplotypes in a sample $\mathbf{X}$ and N is the number of distinct haplotypes in the population. The performance of the estimators can indicate if some method consistently performs well under diverse conditions and if some methods are preferable given certain limitations.

Now I will state the problem in a more formal manner. Let P be a population of haplotypes and $\mathbf{X}_1, \mathbf{X}_2 \ldots, \mathbf{X}_m$ be m random samples of size n drawn from P, with the restriction of not containing the unseen haplotype $\mathbf{h}_{x,j}$, which is also randomly sampled in each of the m trails, with $1 \leq j \leq m$. Further, let $F_{x,j}$ be the population frequency of our unseen haplotype $\mathbf{h}_{x,j}$ and $\hat{F}_{x,j}$ be the estimate of F_x from the j'th sample augmented with $\mathbf{h}_{x,j}$. Given that the population is simulated, it is possible to compute the mean squared error $MSE(\hat{F}_{x,j})$, since F_x is known. $MSE(\hat{F_x})$ can be interpreted as the prediction error. $MSE(\hat{F_x})$ can then be used to evaluate the different estimators with respect to both bias and accuracy.

Bias is a measure that can characterize if an estimator is conservative given the current data. Having conservative estimates is important in forensic genetics, because we don't want the evidence in favour of H_1 to be stronger than it actually is.

1.4 Current state of Y-haplotype match probability and models

There is a lot of discussion of whether to try to build models that uses the genetic information inherent in the markers in a DNA to improve haplotype frequency estimates (Brenner, 2010). The alternative is to only the use count information in the profile database. This include the count of the distinct haplotypes and the size of the database. These different categories of methods will be referred to as lineage based methods (ABMs) and count based methods (CBMs). The simplest CBM is called the count method (CM) (Section 1.4.1). Some researchers recommend using the CM along with an estimated upper bound and subpopulation correction F_{st} in assessing the rarity of a haplotype. This procedure gives a simple, conservative estimate. (Budowle et al, 2007). Analysis of subpopulation structure in Texas gave this approach further approval, especially when DNA profiles comprised 10-16 or more markers (Budowle et al, 2009). Other researchers have argued that using only the count information in the sample of haplotypes entails a substantial loss of information (Buckleton et al, 2011). This is based on the fact that the evolutionary history of a haplotype is encoded in the profile, and its ancestry can be traced given enough Y-STR data. Their suggested approach is to model allele frequencies in the same way as it is done in autosomal work with some correction procedure, but there is now reason to expect that this approach is hard, but not intractable. However, the frequency surveying method, which is a ABM has later been questioned by some of the same researchers who helped developed it (Caliebe et al, 2015).

Brenner argues that using the genetic information in types only observed once in a sample, can provide very limited amount of information. Therefore, not using this information should not lead to significant loss of estimation accuracy (Brenner, 2010) thereby indirectly arguing that CBMs are sufficient. However, he acknowledges that the Discrete Laplace method (DLM) developed M M. Andersen among others, looks interesting. (Brenner, 2013). The DLM is a ABM that uses the genetic information encoded at each marker in each haplotype to cluster similar haplotypes. After clustering, the allele frequencies at each marker are assumed to follow a discrete Laplace distribution. These distributions can then be used to compute frequency estimates for alleles, then be multiplied across loci to achieve an estimate for a particular haploype. (Andersen et al, 2013).

Other methods have been developed under the following names: C.I from 0, Frequency surveying, Infinite alleles model, Average matching chance, t-model, Coalescent theory based methods (Brenner, 2013).

1.4.1 The Count method

The count method is the simplest model for estimating frequency of a previously unseen haplotype, denoted $\mathbf{h}_x$ (Section 1.3). It makes no assumptions about how the haplotypes are distributed in the population and simply uses the sample frequency (empirical frequency) of $\mathbf{h}_x$ as an estimate for the population frequency. It is presented in 'Fundamental problem of forensic mathematics - The evidential value of a rare haplotype' (Brenner,2010) and 'Basic Priniciples for Estimating the Rarity of Y-STR Haplotypes Derived from Forensic Evidence' (Budowle et al 2007) and other papers addressing the same problem (Andersen et al, 2013). It is obviously a CBM and often called the naive count method due to it's simplicity

Before $\mathbf{h}_x$ is observed, consider a database/sample $\mathbf{X}_{pre}$ consisting of n haplotypes. Now the count of $\mathbf{h}_x$ is zero. An empirical estimate for $\mathbf{h}_x$ based on this sample then becomes

$$\frac{0}{n}$$

, which certainly is wrong, because we have observed it in the population. A more general version of this problem is discussed in Section 2.1.2. One solution to this problem is to add $\mathbf{h}_x$ to $\mathbf{X}_{pre}$, thereby forming $\mathbf{X}$ consisting of $(n + 1)$ haplotypes (Brenner, 2010). Now the count of $\mathbf{h}_x$ is one and the empirical frequency estimate for $\mathbf{h}_x$ is

$$\hat{F}_x = \frac{1}{n+1}$$

The good thing about this estimate is that is found to be very conservative, meaning that

$$\hat{F}_x > F_x$$

in most cases (if not all). Thereby not stating that $\mathbf{h}_x$ is more rare in the population than it actually is (Budowle et al, 2007).

Chapter 2

Methods and data

2.1 General problems in Y-STR frequency estimation

2.1.1 Interdependence between Y-STR markers

The lack of recombination at Y chromosome markers used in forensic science is an indication that a matching probability probably cannot easily be obtained by multiplication of estimated allele frequencies across loci. So it is argued that a Y-STR haplotypes is best represented as indivisible units, akin to alleles (Brenner, 2013). By using this haplotype representation, the only way to distinguish the rarity of different haplotypes is based on their sample frequency. So all haplotypes observed once,twice, etc, in the sample will be treated as equal with respect to their potential population frequency. The interdependence across the Y-STR loci does not mean that it is impossible to decompose a marker set into individual markers (which in my opinion is the ideal), then further estimate allele frequencies at each marker separately, like we do in autosomal practice. However, this process is currently very intractable (Caliebe et al, 2015). Not only are there obvious loci interdependence because of the lack of recombination, but there is also interdependence that is dependent on the sub-population structure (Caliebe et al, 2015). Below I summarize previous findings regarding the problems with Y-STR marker set decomposition.

A paper by Bruce Budowle et al, provides general guidelines for using forensic evidence based on Y-STR haplotype data. They argue that treating that haplotypes should be treated as a unit, like an allele, for various reasons. First failed attempts by others to correct for dependence and using the product rule across loci. Second, the effect population sub-structure (Budowle et al, 2009). The study used data sampled from three of the largest ethnic populations in Texas, African American ($N = 950$), Caucasian ($N = 957$), and Hispanic ($N = 1005$), typed for 16 Y-STR markers by using AmpFlSTR Yfiler kit. Although the dependence due to lack of recombination is inevitable, the dependence due to sub-population structure was not. By an incrementing sub-sets of markers up until max at $r = 16$ loci, the paper found that the effect of sub-population structure gradually vanished when $r \geq 10$ markers were included along with increased discriminatory power (Budowle et al, 2009), measured with fixation index (F_{st}) and PI respectively. These quantities are defined as follows, $F_{st} = \frac{\sigma_S^2}{\bar{p}(1-\bar{p})}$ as mentioned earlier, with $\bar{p}$ as the average haplotype frequency in the total population. $PD = 1 - \sum f_i$ where f_i is the ratio between the number of haplotypes of type i over the total number of haplotypes. Their recommended principles for using Y-STR as forensic evidence is that, first, one should always use autosomal evidence if possible. However, as mentioned in the introduction, Y-STR data is particular powerful in rape cases and cases that spans across multiple generations. Second, great importance should be placed on getting databases that included at least $r = 10$ markers for reducing sub-population effects. Third, use the count estimate with estimated upper bound since no sensible reduction can be made so the product rule can be applied yet (Budowle et al, 2009).

Amke Caliebe, Michael Krawczak, among others, all who have been studying lineage markers and their use in forensic and population genetic studies, sought out to quantify the assumed interdependence across the most common Y-STR loci. They studied haplotype data from 21 markers typed with PowerPlex Y23 set, originating from six different populations (four European, average $N = 1533$, and two Asian, average $N = 650$) (Caliebe et al, 2015). The possible dependence between a set of r markers $X = \{M_1, M_2, \ldots, M_r\}$ was quantified by using Shannon entropy, defined as

$$H(X) = \sum_{i}^{r} -f_i \log(f_i)$$

where f_i haplotype frequency of the marker set under consideration. The Shannon entropy quantifies how much uncertainty there is in the haplotype state. To use a more known example, let us consider a coin. If the coin is fair, then the coin has maximum entropy. So given data from any sequence of previous coin tosses would not yield any extra predictive power for the result of the next coin toss. Further, the Shannon entropy $H(X)$ can be used to define the shared information distance $D(X,Y)$ (Caliebe et al, 2015)

$$D(X,Y) = \frac{[H(X|Y) + H(Y|X)]}{H(X,Y)}$$

where $D(X,Y) \in [0,1]$. So if $D(X,Y) = 1$, then X, Y is conditionally independent. Now it is possible to let X contain one marker M_i, $1 \le i \le r$ and let Y contain one marker M_j with $i \neq j$ up to a total of $r-1$ markers all different from M_i. The study performed pair-wise shared information computation and plotted them. So two markers that are close, based on Euclidean distance metric, have high dependence between them. Here is a excerpt from the study

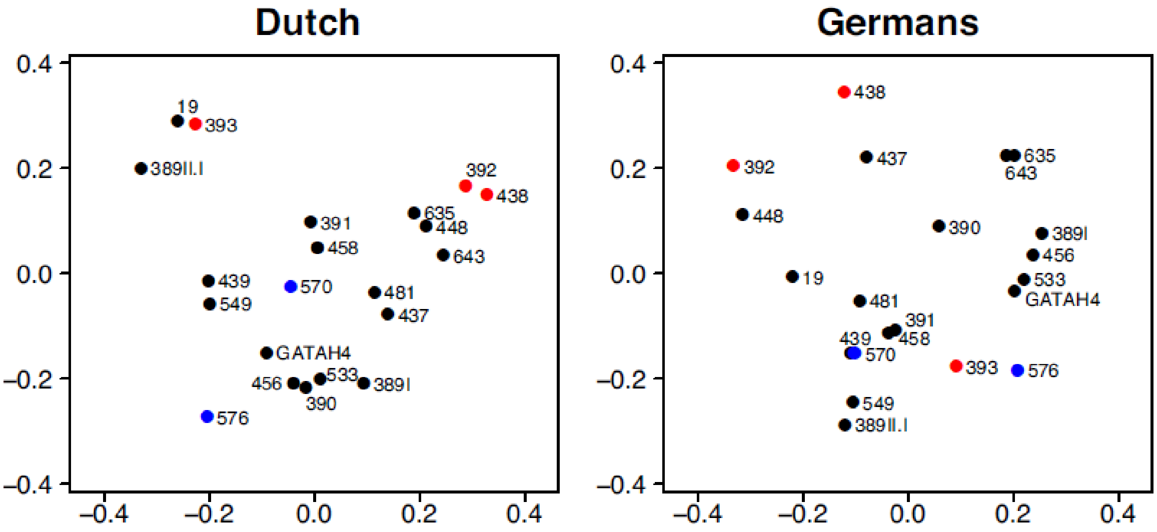


Figure 2. Multidimensional scaling analysis based on shared information distance $[D(X,Y)]$. Colour class is based on marker mutation rates, red for slowly mutating markers and blue for rapidly mutating (Caliebe et al, 2015).

There is a consistent pattern of higher marker dependency between markers with low mutation rates, but even for markers with high mutation rate

(DYS576, DYS578), the dependency is enough to not be neglected. The study concludes that the pattern of interdependence between PPY23 markers is too complex to be broken down into quasi-independent subsets (Caliebe et al, 2015). This means that current models that assume a negligible amount of interdependence and obtain match probability by multiplication across loci are probably invalid.

This means that until more advanced frameworks for modelling this intricate interdependence among common Y-STR markers, a set of Y-STR markers (i.e a haplotype) should be represented as an indivisible unit. An analogy is to think of a set of Y-STR markers as one marker, where a haplotype is akin to an allele variant. By this analogy, different Y-STR haplotypes can be viewed as a allele variants at a highly polymorphic marker.

2.1.2 Implications for modelling

The above results implies that all haplotypes that have been observed equally many times in a database is treated equally. This is however not necessary a fixed limitation, as mentioned in the previous section. Consider some database of haplotypes $\mathbf{X}$ and two new previously unseen haplotypes $\mathbf{y_1}$, $\mathbf{y_2}$. In reality, it is reasonable to think that the actual distance between the two new haplotypes to $\mathbf{X}$ is a quantity that can further improve the frequency estimates. This is ideally achievable some time in the future. I restrict myself to finding good models that does not incorporate this information, hopefully without a dire loss of information. This means that the only feature that can distinguish two haplotypes that is observed one time, is the fact that they are different. What they have in common is that both are observed one time. The same reasoning goes for all haplotypes that are observed k times, where $k \leq n$, where n is the size of the database. If $k = n$, then the database contains one haplotype which is observed n times, this is however a highly unrealistic scenario.

Because of the large number of possible haplotypes in a population, a large part of haplotypes is not represented in the database. As we shall see later, the Good-Turing estimate includes an estimate for the proportion of haplotypes not represented in the database. Assume that some haplotype $\mathbf{h}^*$ exists in some population. Then it follows that the probability of drawing the haplotype at random from this population is not zero. But, a probability estimate for matching $\mathbf{h}^*$ based on sample frequency leads to a contradiction.

$$\hat{p} = \frac{0}{n} = 0$$

Pierre-Simon Laplace faced the same general problem when trying resolve the 'Sunrise problem' in 18th century. Laplace introduced the rule of succession in his treatment of the sunrise problem (see wiki), as technique for assigning non-zero probabilities to empirical events. The rule of secession is stated as the following. If we repeat an experiment that we know can result in either success or failure, n times independently, and get s successes, then what is the probability that the next repetition will succeed? Let $X_1, X_1, \ldots, X_{n+1}$ be conditionally independent rv's given p, that assume either the value 0 or 1. If we have no other information than their counts, then

$$Pr(X_{n+1} = 1 | \sum_i^n X_i = s) = \frac{s+1}{n+2}$$

As an example, consider the sunrise problem, if we have observed that the sun has risen every day, then $\sum_i^n X_i = s = n$, so $Pr(X_{n+1} = 1|\sum_i^n X_i = s = n) = \frac{n+1}{n+2}$.

The above estimator is a less general case of what is called additive smoothing. Additive smoothing is a general version of the estimator Laplace developed for his treatment of the sunrise problem. Additive smoothing has applications in natural language processing (NLP), specifically for developing what is known as naive Bayes classifiers and to develop statistical models for language. The most relevant contribution from NLP is the theory of n-gram models, which is a specific type of language model (LM). In general, a language model is a probability distribution over a sequence of words $w_1, w_2, \ldots, w_m$. So a LM assigns a probability to the sequence $P^* = Pr(w_1, \ldots, w_m)$. So, given a text corpus, and we just randomly select m words, it is intuitive that the randomly selected sequence has a count of zero, therefore we see the need for smoothing techniques in this problem too. The most common approach to deriving an estimate for P^* is the use of n-gram models. This is expressed as the following

$$P^* = Pr(w_1, \ldots, w_m) = \prod_i Pr(w_i|w_1, \ldots, w_{i-1})$$

The conditional probability of a word given $(n-1)$ previous words, can be computed by using frequency counts, represented by the function c, as follows

$$Pr(w_i|w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{c(w_{i-(n-1)}, \ldots, w_i)}{c(w_{i-(n-1)}, \ldots, w_{i-1})}$$

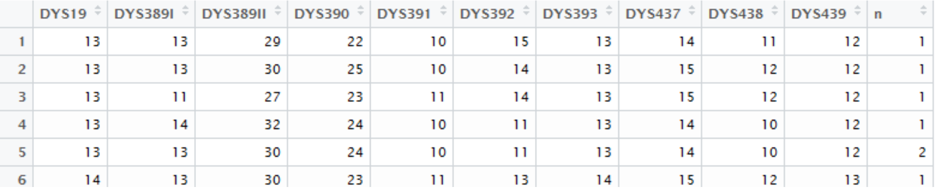
Since in our case (modelling haplotypes), the haplotypes are not ordered, conditioning on a previous haplotype does not make sense. In contrast, modelling the probability of a sentence, i.e P^*, the sequence (w_p, w_q) may

be much more common than (w_q, w_p) and so on. So, the haplotype estimation problem is a unigram in the n-gram framwork where haplotypes are akin to words $w_1, \ldots, w_m$. Since many techniques have been developed too improve the performance of n-gram models, they can also be applied to the case of $n = 1$, i.e unigram models. Examples of this is additive smoothing, Good-Turing smoothing, simple Good-Turing smoothing, etc. The goal of smoothing techniques is to use the count of things we have seen once, twice, ... , to help estimate the counts of things we have not observed before.
To summarize, the haplotypes should (given our current knowledge) be treated as an units, as was argued for in the previous section. This means that we want to use the count information about the haplotypes in a sample/database to best estimate the evidential value of a previously unseen haplotype. Below I will introduce the Good-Turing estimator, abbreviated GTE. The GTE has commonly used in statistical ecology, and has gained more interested branches of machine learning like linguistics, speech recognition and so on. In our context, the GTE is classified as a CBM.

2.2 Data

2.2.1 Y-STR haplotype data, structure and representation

Below we see an excerpt from a Y-STR dataset named danes, that is available through the R package 'disclap', created by Mikkel M. Andersen.



	DYS19	DYS389I	DYS389II	DYS390	DYS391	DYS392	DYS393	DYS437	DYS438	DYS439	n
1	13	13	29	22	10	15	13	14	11	12	1
2	13	13	30	25	10	14	13	15	12	12	1
3	13	11	27	23	11	14	13	15	12	12	1
4	13	14	32	24	10	11	13	14	10	12	1
5	13	13	30	24	10	11	13	14	10	12	2
6	14	13	30	23	11	13	14	15	12	13	1

Table 2.1 A few Y-STR haplotypes based on $r = 10$ loci. Each element in the last column represents the count of the haplotype in corresponding row.

2.2.2 Simulating data

Simulating population dynamics is a very powerful tool for studying population genetics, thus it is also highly relevant in forensic science. The R package *fwsim* makes it possible to generate large haploid datasets that mimics other Y-STR haplotype data to a high degree, thus making it a excellent tool for studying haplotype distributions. (Andersen et al, 2012) Package details can be found at CRAN.

The main reason for the usefulness of simulated data is that if we can develop a model that predict simulated data well, then that model is also a good candidate for predicting real data well too (Andersen et al, 2013). It is important to note that the data is produced by a process that has some similar properties with real evolution natural selection, although omitting many key aspects of evolution, like natural selection. Of course using the same model that generates the data to predict information about process the data is generated from is gives us no new knowledge.

Using *fwsim* to generate data is both easy and computationally fast given the right range of parameters. Examples of how to use *fwsim* is included in the package documentation-file at CRAN. A more analytical assessment of the algorithms asymptotic performance can be found in the paper *'Efficient simulations of Wright-Fisher populations'* (Andersen et al, 2012).

The populations were simulated under a Wright-Fisher model, neutral single-step mutation model using the R-package *'fwsim'.* In total, 2 populations were simulated such that they are as equal as possible with the exception that they differ in number of markers.

- Loci: $r = 8, 11$
- Mutation rate: Empirical (see below)
- Initial population size: $k = 10^5$

The expected population size after g generations and with population growth rate ρ (default $\rho = 1$) is $E(N_g) = \rho^g N_0$ where N_0 is the initial population size (Andersen and Eriksen, 2012). The population growth rate ρ was chosen such that $E(N_g) = \rho^g N_0 = N_{end}$ where $N_{end} = 10^6$. This implies that $\rho = e^{\frac{1}{g}\log(\frac{N_e}{N_0})}$.

In order to make populations more realistic, the markers mutation rates have been parameterized to mimic sets of markers commonly used in Y-STR forensic cases, minimal (8 markers) and Yfiler (11 markers). At each marker, the mutation rate is specified to be equal to the empirical estimate for the marker that are being simulated. After the populations are simulated, datasets of arbitrary size can be sampled. In order to draw realistic samples, the samples are drawn with probability relative to their population frequency. This process usually results in datasets with singleton proportion in the range $(0.6, 1.0 - \epsilon)$ where ϵ is a small real number.

2.2.3 Simulated data

The data used inn all the simulation experiment are sampled from two simulated populations with the expected properties described above. The only differing initial parameter is the number of loci. From these populations, 50 samples of size 500, 2000 and 8000 were drawn, producing 6 different cases in total.

Table 2.2: Summary of the data sets.

sample size n	number of loci r	n.sim
500	8	50
	11	50
2000	8	50
	11	50
8000	8	50
	11	50
All	8	150
	11	150
	All	300

Thus, after the simulation, we have 300 datasets in total, divided into six categories based on sample size and the number of loci in the data.

2.3 Methods

2.3.1 Kappa method

A typical property in Y-STR haplotype data, is the large number of haplotypes observed only once in the sample. We call such haplotypes singletons. The kappa method takes advantage of about the proportion of singletons in the database in the derivation of an estimate for the Y haplotype match probability. The estimation objective is still to estimate F_x in the likelihood ratio $LR = \frac{1}{F_x}$, where F_x is defined in Section 1.3. The derivation of the Kappa method is presented in the next paragraph.

I will first introduce some definitions. Let the crime scene haplotype be denoted by $\mathbf{h}_x$ and consider a sample/database $\mathbf{X}_{pre}$ consisting of $n-1$ haplotypes. Assume that $\mathbf{h}_x \notin \mathbf{X}_{pre}$. Next, let the number of singletons in $\mathbf{X}_{pre}$ be $\alpha - 1$. By setting the size of the database to be $n-1$ and the number of singletons to be $\alpha - 1$ does not change the derived result, it just simplifies computations (Brenner, 2010). Now consider an extended database $\mathbf{X}$, defined as $\mathbf{X} = \mathbf{h}_x \cup \mathbf{X}_{pre}$. Observe that $|\mathbf{X}| = n$ and $|\{\mathbf{h}_x, S_1, \ldots, S_{\alpha-1}\}| = \alpha$. Also by definition, $S_i \neq S_j$ for any $0 \leq i, j \leq \alpha$ and $i \neq j$. The proportion of singletons in $\mathbf{X}$ is denoted by κ, where $\kappa = \frac{\alpha}{n}$.

The singleton proportion $\kappa = \frac{\alpha}{n}$, depends obviously on the sample size n. Also, the number of loci defining the sample haplotypes has a tendency to affect κ. A database composed of 23 loci haplotypes will usually have a larger proportion of singletons than a database composed of 9 loci haplotypes given that the sample size is equal. One reason is because the number of possible haplotypes comprised of 23 loci are much larger than 9 loci (assuming a finite number of variants at each loci).

It is possible to study the growth of κ by treating κ as function of n. However, it is important to note that κ is a fixed constant in haplotype frequency estimation. The growth of κ as a function of the sample size can be fitted to the following function $\kappa(n) = \frac{\theta}{n+\theta}$, where θ is a constant estimated from the sample (Brenner, 2010). Taking the derivative of this function with respect to n yields $\kappa(n)' = -\frac{\theta}{(n+\theta)^2}$. The limit of this function as the sample size goes to infinity is zero, although this extrapolation is probably not valid. It does however indicate that the negative growth of $\kappa(n)$ is very small. Meaning that the number of singletons in a sample is slowly decreasing as a function of n. Note, this last derivation is not a result from Brenner, so should it should be interpreted with caution. Because of this very slow growth of κ, the probability that the next observed haplotype is new, is the same as the probability that the previous haplotype was new. This probability is κ (Brenner, 2010). Brenner justifies this last statement by referring to a Theorem due to Robbins (Estimating the Total Probability of the Unobserved Outcomes of an Experiment, Herbert E. Robbins), but does not explicitly state the Theorem in the paper.

Let T be the haplotype of a suspect unconnected to the crime, i.e, an innocent man. Now, the problem consist of computing $Pr(T = \mathbf{h}_x)$. Let M denote match ($T = \mathbf{h}_x$), O denote observed match (T matches something in $\mathbf{X}$) and S denote singleton match (T matches some singleton in $\mathbf{X}$). Also, if $T = \mathbf{h}_x$, then all the steps in equations 2.1-2.6 must be true (Brenner, 2010; Page 3) First, $Pr(O) = 1 - \kappa$, since it is new type with probability κ. Although, I must admit that I don't understand this step. My guess is that because the probability of not being observed is the same as the probability of being a new haplotype, that is κ, it follows from the inclusion-exclusion principle that $Pr(\neg O) + Pr(O) = 1$, thus $Pr(O) = 1 - \kappa$. Second, $Pr(S|O) \approx \kappa$. Third, $Pr(M|S \cap O) = \frac{1}{\alpha}$. Last, $Pr(M) = Pr(M \cap S \cap O)$. Putting all this together, we find that

$$F_x = Pr(Match) \tag{2.1}$$

$$= Pr(M \cap S \cap O) \tag{2.2}$$

$$= Pr(M|S \cap O)Pr(S|O)Pr(O) \tag{2.3}$$

$$\approx (1 - \kappa)\kappa\frac{1}{\alpha} \tag{2.4}$$

$$= (1 - \kappa)\frac{1}{n} \tag{2.5}$$

therefore

$$LR \approx LR_\kappa = \frac{n}{(1 - \kappa)} \tag{2.6}$$

The factor $\frac{1}{(1-\kappa)}$ is called the inflation factor.

To summarize, the KM gives the following estimate for the frequency of a previously unseen haplotype $\mathbf{h}_x$

$$\hat{F}_x = \frac{1 - \kappa}{|X|} = \frac{1 - \kappa}{n}$$

2.3.2 The discrete Laplace method

In addition to the three CBMs (CM, KM and GTE), the DLM will be used to represent a LBM.

Let $\mathbf{h}_x = (h_1, h_2, \ldots, h_r)$ be our previously unseen haplotype, where $h_k \in \mathbb{Z}$

and $1 \leq k \leq r$. Then the frequency of $\mathbf{h}_x$ can be estimated by

$$\hat{F}_x = \sum_{j=1}^{c} \hat{\tau}_j \prod_{k=1}^{r} f(|h_k - \hat{y}_{jk}|; \hat{p}_{jk}) \tag{2.7}$$

given subpopulation centers $\{\hat{y}_j\}_j$, parameters $\{\hat{p}_{jk}\}_{j,k}$ and prior probabilities $\{\hat{\tau}_j\}$ from a converged run of the EM algorithm (Andersen et al, 2013). The density f is the discrete Laplace distribution, which is a member of the exponential family of distributions. A full derivation of the method is available in the paper 'The discrete Laplace exponential family and estimation of Y-STR haplotype frequencies' (Andersen et al, 2013).

The DLM is implemented in the package 'disclap' on CRAN, written by Mikkel Meyer Andersen and Poul Svante Eriksen. I will use this package to estimate F_x.

2.3.3 Good-Turing frequency estimation

The development of the Good-Turing estimator was driven by trying to provide accurate estimates for the frequency of different types of objects in a population, as the Laplace estimator was inadequate (Good, 1953). The GTE has for example been used widely in statistical ecology, where the classical example of types of objects are distinct species. Let us say that we have observed 2 lions, 4 zebras, 1 hyena, 1 giraffe and so on. This data can then be used to estimate the number of types (number of different species) and the frequencies of the different species. Types observed equally many times has the same frequency estimate (Gale, 1995). In later years this method has gained interest in the field of computational linguistics (Gale, 1995) and in machine learning in general. In computational linguistics, types of objects can be defined as individual words, sequences of words and so on.

The general nature of method makes it suitable as a CBM for estimating the frequency of a previously unseen type. Since the GTE can estimate frequencies for objects observed a particular number of times. The main application of the GTE in this thesis is therefore to estimate the frequency of objects observed one time, i.e, singletons. This frequency estimate will therefore represent the frequency estimate of $\mathbf{h}_x$. In common with all CBMs, objects observed the same number of times will have the same frequency estimate, since the only property that can distinguish types is their count. As far as I can see, no other paper has applied this method to estimating the rarity of $\mathbf{h}_x$ and it would be interesting to see how it compares with other CBMs like Brenner's Kappa model.

The method is described in this paragraph after introducing notation. Let the the distinct species observed be numbered $x = 1, 2, \ldots, X$ with count of each species represented by the vecor $\mathbf{c}_x$ with elements c_x. This means that species number one is observed c_1 times, the first element of $\mathbf{c}_x$. Further the number of species appearing t times is $c_t = |\{x : c_x = t\}|$. The probability estimate for a species observed t times is then defined as

$$p_t = \frac{(t+1)S(c_{t+1})}{N * S(c_t)}$$

where N is the total number of species in the sample and $S()$ is the smoothing function. As you see, this method is very seems very simple to implement, but the difficulties lies in the smoothing step. Another paper by Gale and Church (A comparison of the enhanced GTE and deleted estimation methods for estimating probabilities of English bigrams) published in 1991, showed that for small values of t (in the range $1, \ldots, 8$) we have a good enough approximation to set

$$S(c_t) = c_t$$

I implemented the above estimator in R and tried to implement smoothing, however, I found a R package that contains Good-Turing estimation with smoothing on CRAN, so I used that one instead. The GTE is contained in the R package 'edgeR', which contains a many functions related to analysis of gene expression data. Information about this package can be found on CRAN.

2.4 Simulation experiments

The main purpose of the simulation experiments is to investigate how different estimators behave with respect to their accuracy, bias and under different sampling conditions. Another important goal is to check if certain sampling factors like sample size and number of loci, affect all the estimators in the same way consistently. It may be the case that some estimators only perform well under very limited conditions and other estimators perform well for a wide range of assumptions. Hopefully, this could provide more understanding of where further research into Y-STR haplotype frequency estimation should be focused. It is important to note that the results derived in this thesis are based on simulated data, thus is not empirical by definition. Researchers in this field tend to use simulated Y-STR data to explore and validate their models (Andersen et al, 2013; Brenner, 2010). Results derived from simulated data, even if simulated from very realistic models should be evaluated with a degree of skepticism and ideally be tested against real data. The simulation study consists of two major parts, both described in more detail in section 2.5.1 and 2.5.2. The results are reported in chapter 3. First, a brief overview of the goals of each study.

The simulation study will be devoted to analysing some of the properties of the estimators under varying conditions. A common way of assessing the performance of different estimators in Y-STR simulation studies is to compare estimates $\hat{\theta}$ derived from sampling with the population parameter θ (Andersen et al, 2013; Brenner, 2010). This means that we want to compare how well one estimator does in estimating the population frequency F_x of the previously unseen haplotype $\mathbf{h}_x$ from a population sample $\mathbf{X}_i$.

The second goal, as mentioned in the start of this section, is more exploratory in nature. By this I mean that the behaviours of the estimators are still of interest, but is not the main objective. The objective of this part of the simulation study is to test how sampling factors impacts the estimates, maybe in a general manner. It may also be the case that some estimators only performs well under very limited conditions. This estimator property could be problematic if it were to be used in practice, as it is hard to really know if the required conditions are satisfied.

2.4.1 Computation of method estimates

Before statistical tests are performed, all the estimates must be computed from the simulated datasets. All the estimates are computed in R and further stored in a dataframe of results. For the sake of convenience, all the frequency estimates are then transformed into a dataframe that represents the estimates in terms of log LR of the frequency estimate. In addition to the estimated results, the dataframe contains information of about the true population value of the true population $log(LR)$. The procedure for computing the estimates will be described in general below. Figure 3.2 in the results section displays an excerpt of the results.

The essential R code for computing all the data and estimates can be found in the appendix of this thesis.

In total, we have simulated random samples $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_{300}}$ drawn from the populations described in Section 2.2.2. Each sample $\mathbf{X}_j$ is also augmented with a random sampled haplotype $\mathbf{h}_x$ that serves as our previously unseen haplotype. In mathematical terms, $\mathbf{X}_j = \mathbf{X}^j_{pre} \cup \mathbf{h}_{x,j}$ and $\mathbf{X}^j_{pre} \cap \mathbf{h}_{x,j} = \emptyset$, where $j = 1, 2, \ldots, 300$, so the count of $\mathbf{h}_{x,j}$ in $\mathbf{X}_j$ is one etc. One reason for sampling a new random $\mathbf{h}_{x,j}$ in each sample is that even if the count in sample will always be one, the count of $\mathbf{h}_{x,j}$ in the population is not necessary one. For example, consider two haplotypes with count one some database. Assume that one of the haplotypes belongs to a man with 14 brothers and the belongs to a man with no brothers. Then it is probable that the population frequency of the first haplotype is larger than the latter.

The sample $\mathbf{X}_j$, together with computed estimates, the true population frequency / log(LR) of $\mathbf{h}_{x,j}$, the sample coverage and singleton proportion constitutes one simulation, index under the column sim (Figure 3.2). Each unique combination of the factors sample size n (minimum, medium and maximum) and the number of loci r (minimum and maximum) is simulated

$m = 50$ times according to the scheme above, resulting in 300 simulations in total.

It is important to note that the singleton proportion, sample coverage, true population log(LR) and all the estimates in each row j are computed from the corresponding sample $\mathbf{X}_j$. The experiments in the following section (Section 2.5.2) are all based on these 300 simulations.

In the next section I will use $\log(LR)$ to denote the true population log likelihood ratio for $\mathbf{h}_x$ and $\log(\hat{LR_k})$ to denote the estimated log likelihood ratio of $\mathbf{h}_x$ by method number k, where $k = 1$ is CM, $k = 2$ is BK, and so on. Also, since the count method is only a function of the sample size, it has a constant value in all results with equal sample size. Therefore it will be excluded from tests since it does not have any variance. This implies that he three methods that are going to be tested are the Kappa model (KM) by Brenner, Good-Turing esimation, and discrete Laplace model (DLM).

In order to clarify which subsets of the total 300 simulations that are included in a particular test, I will use the vector $\mathbf{r}$. $\mathbf{r}$ is defined as $\mathbf{r} = (n_i, r_j)$, where $n_0 = 0, n_1 = 500, \ldots, n_3 = 8000$ and $r_0 = 0, n_1 = 7, r_2 = 11$. The value $n_0 = 0$ means all n. For example, $\mathbf{r}^* = (0, 7)$ is all simulations with the number of loci equal to 7 (150 simulations). Another example, in Table 3.3, sim 55,80 and 90 are 3 out of 50 simulations in the subsets $\mathbf{r}^* = (500, 11)$. Also, let $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_6$ represent the standard cases $(500, 7), (500, 11), \ldots, (8000, 11)$.

2.4.2 Simulation experiment, Comparison of estimators

The two most common measures of estimator performance in Y-STR haplotype frequency estimation is accuracy and bias (Andersen et al, 2013; Brener, 2009). These measures are generally quantified by the mean squared error (MSE) of an estimator $\hat{\theta}$ ($\log(\hat{LR})$) in our case). The MSE of an estimator $\hat{\theta}$ is defined as $MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$ (θ is $\log(LR)$ in our case). The MSE is a sum of the variance of $\hat{\theta}$ and the bias $(bias(\hat{\theta}, \theta))^2$ (see wikipedia / MSE for derivation).

If an estimator has a consistent bias, it is desirable that it underestimate the LR instead of overestimating it. This is because overestimation would give more evidential strength in favour of conviction than there actually is (see Section 1.2). A method that consistently overestimates the LR is said to be anti-conservative. To see why this is the case, consider the following general relationship.

$$\hat{LR} - LR > 0$$

$$\hat{LR} > LR$$

$$\frac{1}{\hat{F}_x} > \frac{1}{F_x}$$

$$\hat{F}_x < F_x$$

Which means that we have reported the frequency of $\mathbf{h}_x$ to be lower than it actually is in the population. In a forensic case, this would give more evidential value for conviction than there really is (see Section 1.2). Note that this relationship hold for $\log(LR)$ as well, since the LR is always greater than zero.

Also, if an estimator $\hat{\theta}_1$ has a smaller MSE than another estimator $\hat{\theta}_2$, it means that it's estimates are closer to the true population value of θ, which is a desirable property in general. However, in our case the value of a small MSE is not the only property we try achieve. As we saw in the previous paragraph, we must also see if the bias is consistently skewed in an anti-conservative direction.

To recap, the two properties we want to investigate when comparing estimators are the MSE and bias. Generally this entails a trade-off, we want to minimize the MSE of an estimator, but not at the expense of conservativeness.

Estimator bias, probability of overestimation

A one-sided t-test with assumed unequal variance is performed between the $\log(LR)$ estimate and the true population $\log(LR)$ for all methods and all six scenarios $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_6$. This test can indicate if any of the methods on average overestimates the $log(LR)$ in any of the scenarios. Let μ_k be the true mean $log(LR)$ in arbitrary scenario, and $\hat{\mu}_k$ be the estimated mean $\log(LR)_k$ by method k in the same scenario. We then test

H_0: $\hat{\mu} - \mu_k \leq 0$

H_a: $\hat{\mu} - \mu_k > 0$

at significance level $\alpha = 0.05$ and degrees of freedom $df = n.sim - 1 = 49$. If the alternative hypothesis is true, it means that the method k has overestimated the true population $\log(LR)$ on average in that scenario.

Pairwise comparison of MSE

In this test, a pairwise two-sided t-test with assumed unequal variance of the MSE for all the estimators in all six scenarios $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_6$ were performed. Let $\hat{L}_i$ be the estimated $\log(LR)$ for one of the methods i and $\hat{L}_j$ be the estimated $\log(LR)$ for one of the other methods $j \neq i$ from one of the six scenarios. Also let L be the true population $\log(LR)$ in the same scenario. The following hypothesis will then tested in each scenario for all methods i, j.
H_0: $\text{MSE}(\hat{L}_i) - \text{MSE}(\hat{L}_j) = 0$
H_a: $\text{MSE}(\hat{L}_i) - \text{MSE}(\hat{L}_j) \neq 0$
at significance level $\alpha = 0.05$ and degrees of freedom $df = 2 * (n.sim - 1) = 98$. A rejection of the null hypothesis then means that there is a significant difference in estimator MSE in that scenario. I.e, one method is more accurate on average in that particular scenario.

Chapter 3

Results

This chapter will present the results from the simulation studies. The first section is devoted to a brief summary and exploration of the computed estimates and the data. Further, the next sections reports the results from the experiments designed in the previous chapter. Note, I will also use log LR to represent the estimation results. As explained in Section 2.4.1, this is no problem, as the LR is always (in my restricted case) the inverse of the frequency of the haplotype under consideration. To see this, let $\hat{F}_x$ be the estimated haplotype frequency of x, then $\hat{F}_x \in (0, 1)$ and also by definition $LR_x = \frac{1}{\hat{F}_x} > 0$. Positive numbers are also more convenient to work with, and can be used to avoid some rounding errors when dealing with small numbers. So if the reader is interested in the frequency representation, it is easy to convert back using the inverse relationship just described and the fact that the natural logarithm is always injective on $\mathbb{R}_{>0}$. The $\log(LR)$ will be denoted with L to make the notation more concise.

Table 3.1: Mean sample coverage and singleton proportion in all simulation scenarios.

Notation	Meaning
L	True population $\log(LR)$
$\hat{L}_1$	CM estimated $\log(LR)$
$\hat{L}_2$	KM estimated $\log(LR)$
$\hat{L}_3$	GTE estimated $\log(LR)$
$\hat{L}_4$	DLM estimated $\log(LR)$

3.1 Computed estimates and data characteristics

3.1.1 Sample singleton proportion and coverage

From figures 3.1;3.2, we see there that there is a trend of increased sample coverage when the sample size is larger and a decrease when the haplotypes are composed of more markers. Both these results are expected. Full sample coverage is certainly attained when n is equal to the population size N and possibly much sooner, depending on the haplotype diversity in the population. The same sampling factors also seem to affect the sample singleton proportion κ in the opposite manner.

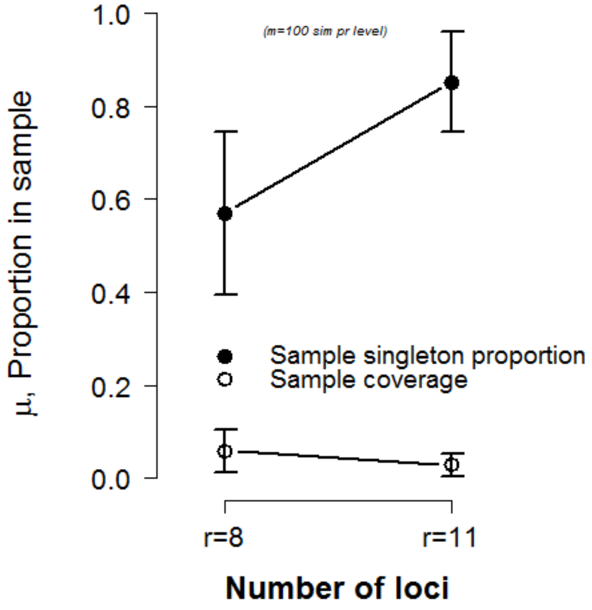


Figure 3.1: Simulations with $n = 500, 2000, 8000$ pooled and grouped under the factor loci.

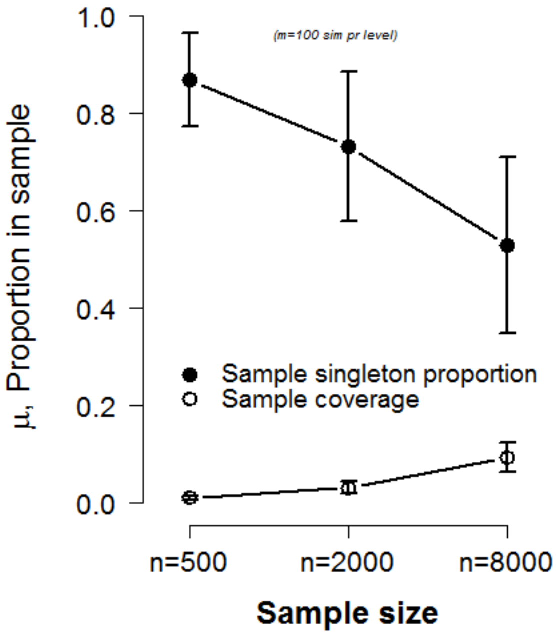


Figure 3.2: Simulations with $r = 8, 11$ pooled and grouped under the factor n.

Table 3.1 gives a more precise representation of what was discussed in the beginning of this section. Given a fixed value of r, mean coverage always increases with larger sample size and the singleton proportion decreases. For $r = 8$, there is a mean increase of 0.029 and 0.079 in coverage when sample size is increased from 500 to 2000 and 2000 to 8000 respectively. For For $r = 11$, there is a mean increase of 0.0133 and 0.045 in coverage when sample size is increased from 500 to 2000 and 2000 to 8000. This indicates that coverage increases faster with sample size when haplotypes consists of fewer markers.

Maximum and minimum mean sample coverage occurs when $n = 8000$ (maximum sample size used), $r = 8$ (minimum number of loci used) and $n = 500$ (minimum sample size used), $r = 11$ (maximum number of loci used) respectively.

Table 3.2: Mean sample coverage and singleton proportion in all simulation scenarios.

			coverage	singleton.prop
sample size n	number of loci r	n.sim	mean	mean
500	8	50	0.013046	0.776972
	11	50	0.004747	0.962709
	All	100	0.008897	0.869841
2000	8	50	0.043041	0.579630
	11	50	0.018087	0.884226
	All	100	0.030564	0.731928
8000	8	50	0.122143	0.350052
	11	50	0.063613	0.708753
	All	100	0.092878	0.529403
All	8	150	0.059410	0.568885
	11	150	0.028816	0.851896
	All	300	0.044113	0.710390

3.1.2 Summary of simulation estimates

Table 3.3: Summary of log-LR estimates for 10 random simulations of a total of 300 simulations with sample size n, number of loci r.

sim	n	r	s	κ	L	$\hat{L_1}$	$\hat{L_2}$	$\hat{L_3}$	$\hat{L_4}$
55	500	11	0.005	0.942	12.40	6.219	9.070	7.221	9.554
80	500	11	0.005	0.964	13.09	6.219	9.547	7.809	11.035
90	500	11	0.005	0.960	12.40	6.219	9.441	7.528	10.918
111	2000	8	0.043	0.579	13.83	7.602	8.468	7.684	9.608
148	2000	8	0.043	0.578	13.83	7.602	8.466	7.656	8.113
175	2000	11	0.018	0.897	13.78	7.602	9.871	8.067	12.485
193	2000	11	0.018	0.894	13.78	7.602	9.843	8.416	12.469
224	8000	8	0.122	0.347	13.83	8.987	9.413	8.896	12.407
228	8000	8	0.122	0.350	13.83	8.987	9.418	8.845	9.694
281	8000	11	0.064	0.715	11.59	8.987	10.243	9.372	14.006

None of the methods appear to overestimate the true population LR consistently and have the following relationship with respect to conservativeness:

$$\text{Count} > \text{Good-Turing} > \text{Brenner} > \text{DiscLap}$$

The same pattern is also appear to be consistent in the simulation cases where $n = 500$ and the number of loci $r = 8, 11$.

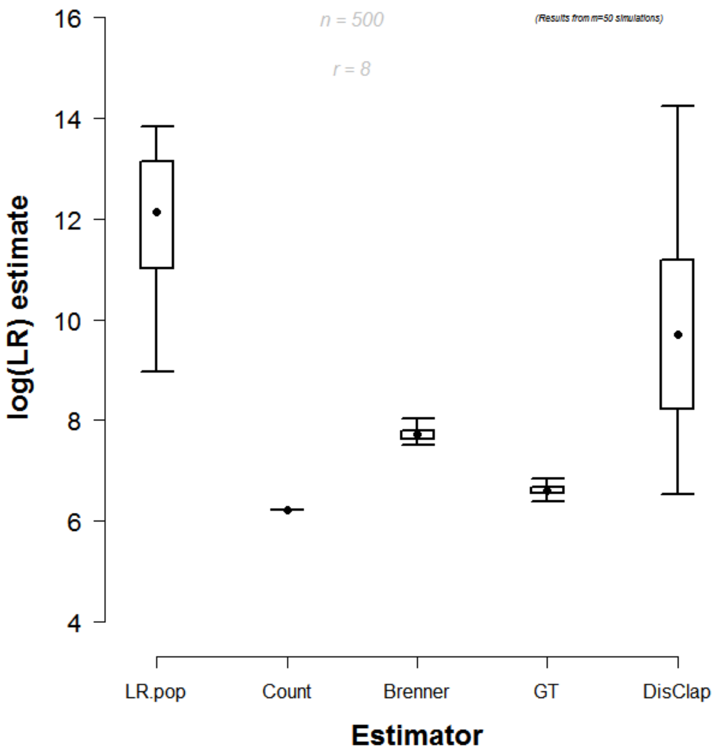


Figure 3.3: Log likelihood ratio estimates for all the methods based on $m = 50$ simulations and the true population log likelihood ratios of the different unseen haplotypes randomly sampled in each simulation.

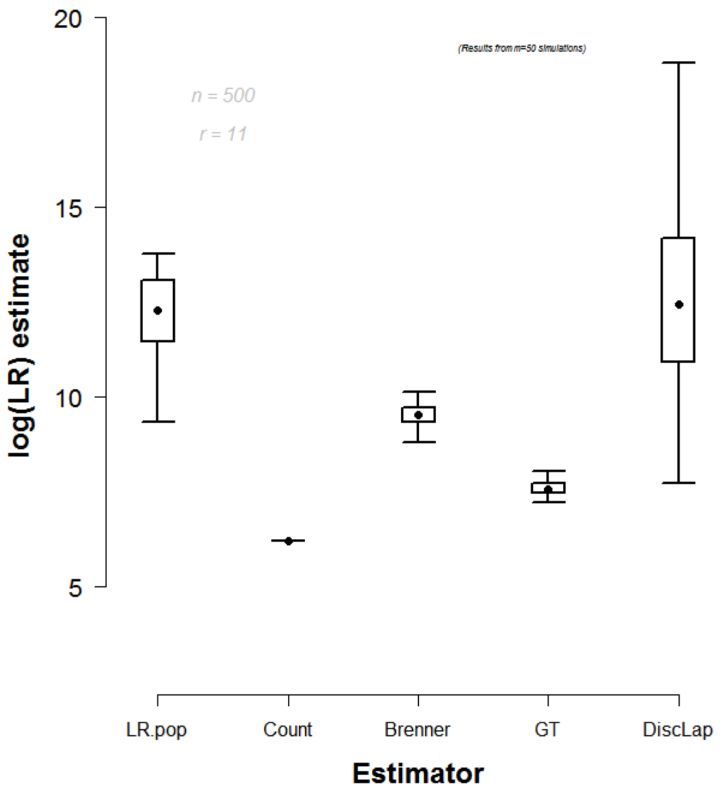


Figure 3.4: Log likelihood ratio estimates with the all the same simulation properties as in Figure 3.3, with the exception of the number of loci $r = 11$.

In the simulation case with $n = 500$ and $r = 8$, the plot (Figure 3.3) indicate that all estimators are conservative. However, in the analogous case with $n = 500$ and $r = 11$ (Figure 3.4), the DLM appears to overestimate the LR. Also for the sample mean values, we have in both of these scenarios the following relationship, $|\hat{L_4} - L| < |\hat{L_j} - L|$ for the other estimators $j \neq 4$, suggesting that the DLM is the most accurate estimator. The figures also suggests that the DLM have higher variance than the other estimators.

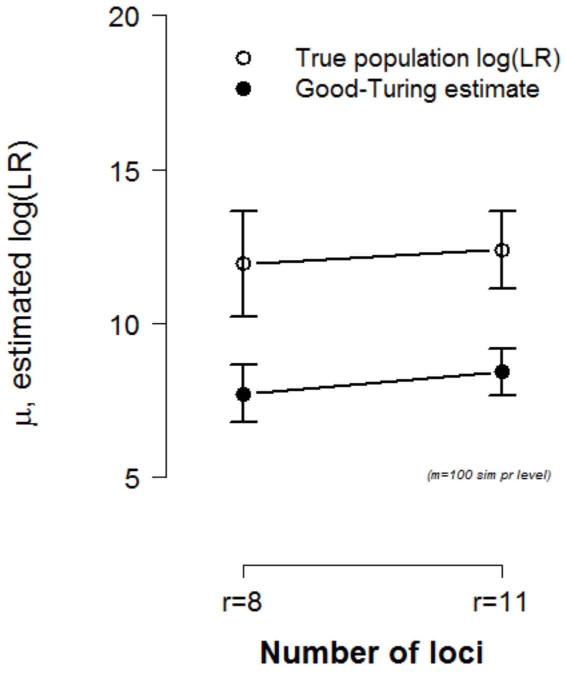


Figure 3.5: Good-Turing log likelihood ratio estimates based on different number of loci.

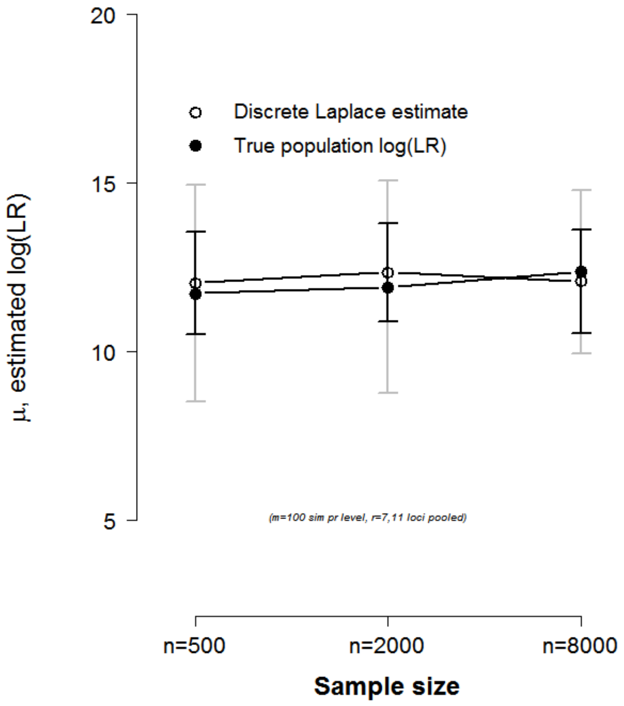


Figure 3.6: Discrete Laplace log likelihood ratio estimates based on different sample size.

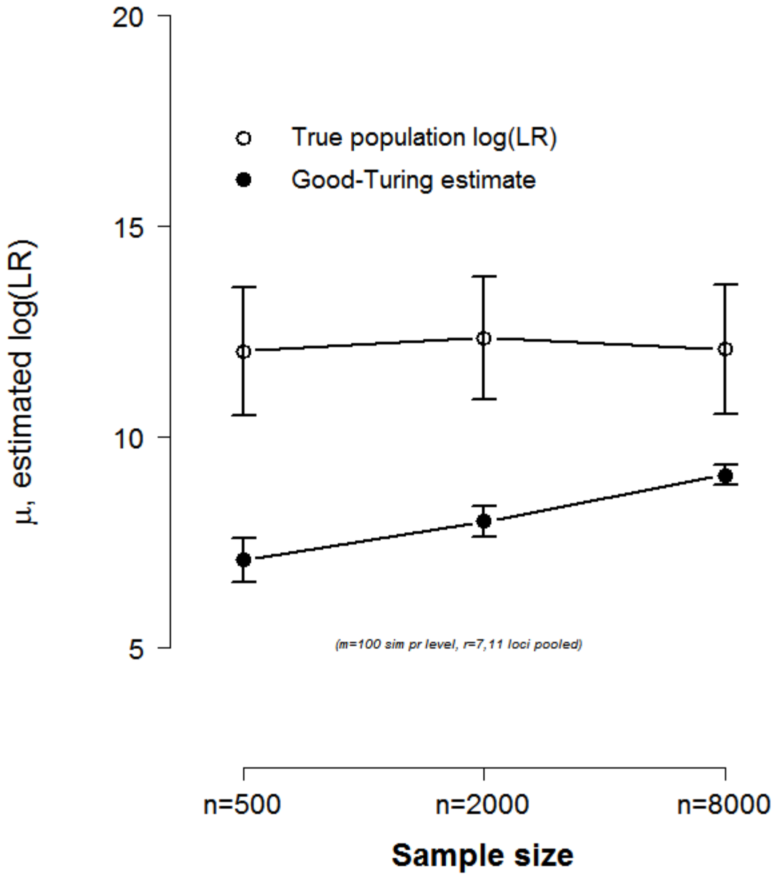


Figure 3.7: Good-Turing log likelihood ratio estimates based on different sample size.

The log LR tend to increase as more loci are included in the frequency estimates (Figure 3.5). The log LR estimates based on DLM is close to the population log LR, but seems to have high variability in the estimates (Figure 3.6). In contrast, log LR estimates based on GTE tends to underestimate the population log LR (conservative) (Figure 3.7). Also, the log LR estimates based on GTE appear to have low variance and slowly approach the population log LR as the sample size increases.

3.2 Experiment results, comparison of estimator performance

3.2.1 MSE differences

Table 3.4: MSE and Variance computed for the KM(2), GTE(3) and DLM(4) based on 50 simulation in each scenario.

n.sim	n.sample	n.loci	MSE.2	var.2	MSE.3	var.3	MSE.4	var.4
50	500	8	20.10	146.62	30.63	253.09	11.83	288.69
50	500	11	8.794	39.73	23.29	120.52	12.558	436.98
50	2000	8	16.796	106.67	23.48	165.91	10.106	110.49
50	2000	11	8.938	32.26	18.65	85.07	12.507	504.59
50	8000	8	8.241	56.36	11.06	86.06	6.138	43.77
50	8000	11	6.397	24.03	11.06	48.16	7.290	92.02

Table 3.5: Results from pairwise two-tailed t-tests with 49 degrees of freedom between the mean log LR for all methods in all simulation scenarios at sign level $\alpha = 0.05$ with $df =$ n.sim $- 1 = 49$. Unequal variance was assumed. First, KM(2) is tested against GTE(3), then KM(2) against DLM(4), ..., until exhaustion.

n.sim	n.sample	n.loci	$p_{2,3}$	$d_{2,3}$	$p_{2,4}$	$d_{2,4}$	$p_{3,4}$	$d_{3,4}$
50	500	8	0	10.523	0.005	8.268	0.000	18.791
50	500	11	0	14.498	0.244	3.764	0.005	10.734
50	2000	8	0	6.682	0.000	6.689	0.000	13.371
50	2000	11	0	9.708	0.289	3.569	0.086	6.139
50	8000	8	0	2.822	0.038	2.103	0.000	4.925
50	8000	11	0	4.664	0.593	0.893	0.049	3.771

In Table 3.5, $d_{i,j}$ is defined as $d_{i,j} = |MSE(L, \hat{Li}) - MSE(L, \hat{Lj})|$, i.e, the absolute MSE difference between method i and j and $p_{i,j}$ is p-value for the corresponding test in difference. Table 3.5 summarizes the results of a test for significant differences in MSE. Differences in MSE was highly significant in many cases, except between KM and DLM in scenarios $\mathbf{r}_{2,4,6}$ and between GTE and DLM in scenario $\mathbf{r}_4$ Increases in sample size seems to decrease the MSE difference of estimators in general.

3.2.2 Probability for overestimation

Pair-wise t-test for overestimation

Table 3.6: Results from pairwise one-tailed t-tests with 49 degrees of freedom between the population log LR and the estimated log LR for all methods in all simulation scenarios at sign level $\alpha = 0.05$ with $df = \text{n.sim} - 1 = 49$. Unequal variance was assumed. The last three columns display the p-values from the experiment.

n.sim	sample size	r	p.KM	p.GTE	p.DLM
50	500	8	1	1	0.999
50	500	11	1	1	**0.040** *
50	2000	8	1	1	1.000
50	2000	11	1	1	**0.014** *
50	8000	8	1	1	0.825
50	8000	11	1	1	**0.008** **

The DLM was the only method that significantly overestimated the population $\log(LR)$. This occurred in scenarios $\mathbf{r}_{2,4,6}$. The tendency of DLM to overestimate appears to increase with haplotypes comprised of more loci. It is encouraged to further investigate if this tendency manifests itself on real Y-STR data and under different simulated conditions.

Chapter 4

Discussion

4.1 Brief summary of the thesis, problem and findings

Accurate estimates of Y-STR haplotypes frequencies is an important problem in forensic genetics, since the estimates are used to compute the likelihood ratio for the evidential weight of a DNA profile found at a crime scene. Estimating Y-STR haplotypes is more intricate than it is for haplotypes found on autosomal DNA due to the lack of recombination on Y-STR markers selected for use in forensic genetics and there is currently no evidence to suggest that there is an easy method for decomposing haplotypes into individual markers (Caliebe et al 2015). This result that motivated me to compare methods that do not decompose haplotypes into individual markers, i.e CBMs, with a method that do (ABM), namely the discrete Laplace method (Andersen et al, 2013). CBMs are based on less assumptions about the underlying distribution of haplotypes than ABMs in general, and therefore requiring less conditions to be satisfied (they are more general) when applying the model to a problem. Also, it is hard to know what conditions like haplotype diversity, population substructure, etc, that affect the estimates the most. Based on this property, a CBM is preferable to an ABM if the difference in accuracy and bias is negligible and are the only properties of interest. CBMs also appear to have less variance in the estimates. However, ABMs can compute different frequency estimates for two haplotypes only observed once, a highly wanted property that CBMs do not have. This is not possible with CBMs since they do not differentiate between types observed equally many times in a sample. The ability to discriminate between types observed equally many times is important in cases where there are multiple possible suspects with haplotypes only observed once in the sample. This property can also be used to identify the most likely contributors to a mixture of Y-STR profiles. The paper 'Identifying the most likely contributors to a Y-STR mixture using the discrete Laplace method', by Andersen et al, provides a framework for this procedure using the DLM. This could be done by any ABM in theory.

To recapitulate, there seems to be a trade-off in selecting either of the two classes of methods, this will be discussed more in Section 4.3. I will now summarize the findings of the simulation study. Regardless of method category (CBM or ABM), the goal of the simulation study was to estimate the haplotype frequencies F_x of haplotypes randomly sampled from a given population, from the corresponding random samples with different sample size and composed of different number of markers (Section 1.3, 2.4.1). This was done in many different scenarios (Section 2.2.3, 2.41) with different methods.

When all the estimates were computed, statistical tests were performed in order to test for significant differences in accuracy (MSE) and bias (overestimation). Note, the results can be reproduced and all the necessary R code are provided in Appendix (Section 5.2). Having said that, simulating the populations is a stochastic process even if the input parameters are constant. The populations used in the study is therefore stored on a hard drive and can provided by request.

The study revealed that the DLM performed best in terms of accuracy in general in all simulation scenarios. However, the DLM also appeared to be the estimator with the highest variance and with more variance when haplotypes comprised $r = 11$ markers than $r = 8$ markers. Given fixed number of loci r, the variance appear to decrease substantially, from 504.59 to 92.02 when sampled size is increased from 2000 to 8000 (Table 3.4, column 8). This can indicate that the DLM require relatively large samples when dealing with a sufficient number of markers. It is also important to point out that the DLM was the only method that overestimated the LR significantly (Table 3.6). This happend only when haplotypes comprised $r = 11$ markers. More rigorous testing of this phenomenon is suggested, as modern Y-STR kits can provide haplotypes consisting of up $r = 27$ markers (Yfiler plus). Overall, the DLM appears to be the superior method with respect to all measures of performance with the exception of a potential anti-conservative tendency for haplotypes composed of $r = 11$ markers.

With regard to the CBMs, the KM and GTE, there are no results that favour GTE over KM. Both methods are conservative across all scenarios (Table 3.6), but the KM is more accurate and has less variance on average in all scenarios (Table 3.4).

4.2 Study limitations

The major limitation of the study is that all results were derived from simulation. It is assumed that the simulated populations approximate real populations to a sufficient degree. This requires empirical knowledge about the properties of the system to be simulated. So the derived results are completely dependent on how well simulated Wright-Fisher populations maps on to real populations. Simulated Wright-Fisher populations are generally thought to be very powerful for studying population dynamics and that key insights can be derived from them (Andersen et al, 2012). More research into how to parameterize the simulations in order to achieve good approximations could be interesting. I tried to mimic the mutation rates by using empirical mutation rates, hopefully yielding more realism. Another approach is to diversify the populations. The DLM is for example verified using 12 different simulated populations (Andersen et al, 2013). However, all the twelve populations was defined by $r = 7$ markers, thus not addressing the accuracy and bias for haplotypes composed of more markers. This was shown to have a significant impact on the estimates in this paper.

Also, to further test the effect of different sampling factors like sample size and number of markers comprising the haplotypes, a factorial experiment could have been implemented. By doing this, we could statistically evaluate the effect of each factor on the frequency estimates. This would especially be interesting with respect to the factor number of markers, since the DLM appears to be sensitive to this factor. Including a population with haplotypes comprised of markers of size $r \geq 17$ could challenge the methods even further. This could be defined as a third level(maximum) the factor r (number of loci). Estimates derived from samples from this population would then indicate if the DLM have problems with overestimation given larger r or if this is just a coincidence. The reason for not implementing the last suggestion, is that the computation of haplotype estimates from this type o population never stopped. No asymptotic upper bound on the running time of the DLM are provided in the paper deriving it. It is important to establish the expected running time of the DLM in terms of asymptotic bounds if the method should be used in practice. If this is hard to derive analytically, then estimates should be made. Haplotype frequency estimation based on coalescent theory is a real example of this problem, as the running time increases exponentially with sample size and number of loci (Andersen, Caliebe, et al, 2013). Hopefully, the running time of DLM do not increase exponentially, which would limit the practical applications substantially.

In my opinion, continuing to use simulation studies requires a more rigorous assessment of their validity in order to judge whether a simulated distribution is a good approximation of some empirical distribution. If this is found to be the case (by some method), I speculate that a huge number of possibilities are made available by the amount of data that can be generated. Such tests can be performed on one-dimensional data, like the counts of the various haplotypes in real Y-STR data and the counts in simulated data. For illustration, consider the KolmogorovSmirnov test (wiki, Kolmogorov Smirnov test), which is one such test. The test can be done with the following test statistic

$$D_{n,n'} = \sup_x |F_{1,n}(x) - F_{2,n'}(x)|$$

and by defining $F_{1,n}(x)$ as the empirical count distribution and $F_{12,n'}(x)$ as the simulated count distribution, with rejection region specified in the wiki article.

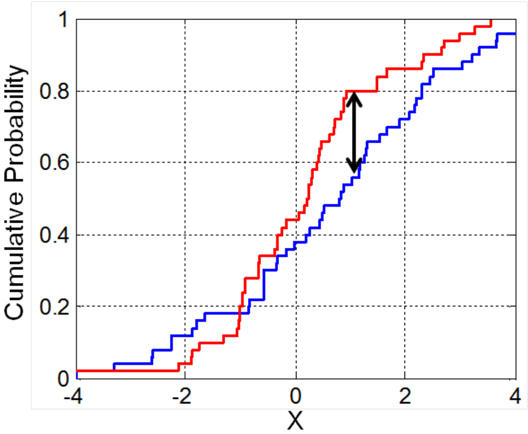


Figure 4.1: K-S test with two empirical distributions. (wiki, Kolmogorov Smirnov test)

However, the similarities between the empirical and simulated count distributions do not necessarily (probably not) entail that the haplotype distributions are the same, so some multi-variable alternative would be needed.

Because of the possible limits with simulation studies, estimator properties should also be tested on real Y-STR with theoretical expectation of behaviour as it is done in 'Estimating Haplotype ...' (Egeland and Salas, 2008). This paper also proposes some pragmatic solutions to haplotype frequency estimation by highlighting the importance of sample coverage in this problem.

4.3 Related and future research

Another method that has been proposed for haplotype frequency estimation is the λ model (Tillmar et al, 2011). The model was developed for estimating haplotype frequencies on X-chromosomal markers, making it a plausible candidate for Y-STR haplotype frequency estimation as well. The frequency of a haplotype i according to this model is

$$F_i = \frac{c_i + \lambda p_i}{C + \lambda}$$

where c_i is the count of the haplotype in the database and p_i is the expected frequency of the haplotype. In the problem of estimating the probability of observing an previously unseen haplotype, the count of the haplotype is one. Further, the total count is defined as the following, $C = \sum\limits_j c_j$, where c_j is the count of some haplotype j. Both the expected frequency and lambda can be estimated from the data. The latter estimate is based on a procedure suggested in (Egeland et al, 2016), that resembles techniques for estimating θ, used in theta correction problems. A major advantage of this model is that is accounts linkage and linkage disequilibrium (LD). Considerable differences in the LR were observed depending on whether linkage and LD were taken into account. It would be very interesting to study if these results would transfer to Y-haplotype frequencies, since linkage and LD is deemed to complex to correct for in Y-STR markers (Caliebe et al, 2015).

4.4 Final remarks

More research into the validity of simulation studies in evaluating methods for Y-STR haplotype frequencies is needed. Simulation studies should best be used as a powerful exploratory tool in my opinion, due to the uncertainty in transferring conclusions to real data. Having said that, the results based on this simulation study revealed some interesting findings. Most noteworthy is the exceptional performance of the DLM relative to the other methods. Also, the relative poor performance of the GTE is interesting. The GTE is known to perform well on other problems with similar count data, like word frequency estimation (Gale, 1995). The GTE also provides an estimate for the sample coverage (Good, 1953). This may be a more proper use of GTE in haplotype frequency estimation problems in general.

Bibliography

[1] Mikkel M. Andersen, Poul S. Eriksen *The discrete Efficient Forward Simulation of Fisher-Wright Populations with Stochastic Population Size and Neutral Single Step Mutations in Haplotypes.* arXiv: 1210.1773 (2012).

[2] Mikkel M. Andersen, Amke Caliebe, Arne Jochens, Sascha Willuweit, Michael Krawczak *Estimating trace-suspect match probabilities for singleton Y-STR haplotypes using coalescent theory..* Forensic Sci. Int. Genet. 7(2):264-271 (2013).

[3] Mikkel M. Andersen, Poul S. Eriksen, Niels Morling *The discrete Laplace exponential family and estimation of Y-STR haplotype frequencies.* Journal of Theoretical Biology; 329:39-51 (2013).

[4] Charles H. Brenner. *Fundamental problem of forensic mathematics The evidential value of a rare haplotype.* Forensic Sci. Int. Genet. 4 (2010).

[5] Charles H. Brenner. *Understanding Y haplotype matching probability.* Forensic Sci. Int. Genet. 8 (2013).

[6] Bruce Budowle, Jianye Ge, Ranajit Chakraborty *Basic Priniciples for Estimating the Rarity of Y-STR Haplotypes Derived from Forensic*

Evidence. Proceedings of the Eighteenth International Symposium on Human Identification, 2007.

[7] Bruce Budowle, Jianye Ge, XG Aranda , JV Planz, AJ Eisenberg and R Chakraborty . *Texas population substructure and its impact on estimating the rarity of Y STR haplotypes from DNA evidence.* J Forensic Sci. 2009 Sep;54(5):1016-21.

[8] J. Buckleton, C. Triggs, and S. Walsh, editors. *Forensic DNA Evidence Interpretation.* CRC Press, Florida, USA, 2005.

[9] Amke Caliebe, Arne Jochens , Sascha Willuweit , Lutz Roewer , Michael Krawczak *No shortcut solution to the problem of Y-STR match probability calculationn.* Forensic Sci. Int. Genet. 15, 2015.

[10] Thore Egeland, Antonio Salas *Estimating Haplotype Frequency and Coverage of Databases.* PLoS ONE 3(12): e3988. doi:10.1371/journal.pone.0003988.

[11] Thore Egeland, Daniel Kling, Petter Mostad *Relationship Inference with Familias and R.* Elsevier Inc, 2016.

[12] William A. Gale *Good-Turing Smoothing Without Tears.* Bell Laboratories, 1995, DOI:10.1080/09296179508590051.

[13] Irving J. Good *The population frequencies of species and the estimation of population parameters.* Biometrika 40 (34): 237264. (1953).

[14] Andreas O. Tillmar, Thore Egeland, Bertil Lindblom, Gunilla Holmlund, Petter Mostad *Using X-chromosomal markers in relationship test-*

ing: Calculation of likelihood ratios taking both linkage and linkage disequilibrium into account. Forensic Sci Int Genet. 2011 Nov;5(5):506-11.

[15] S. Willuweit , L. Roewer. *The new Y Chromosome Haplotype Reference Database., Forensic Sci Int Genet 15, 43-8.* www.yhrd.org 2016.

Chapter 5

Appendix

5.1 Notation

5.1.1 Mathematical Notation

Symbolic	**Conceptual**
$\mathbf{h_x}$	Crime scene haplotype (prev unseen haplotype).
F_x	Frequency of previously unseen haplotype.
$\hat{F}_x$	Estimated frequency of previously unseen haplotype.
$\mathbf{X}$	Data/database augmented with prev unseen haploype (size n+1).
$\mathbf{c}$	Vector of counts of the haplotypes in database.

5.1.2 Abbreviations

Abbreviation	**Concept**
CM	Count method
KM	Brenner’s Kappa model
GTE	Good-Turing estimator
DLM	Mikkel M. Andersen’s Discrete Laplace Method

5.2 R code

5.2.1 Methods for estimating haplotype frequencies

```
ComputeCountEstimate <- function(x) {
  x <- x$N
  N <- sum(x)
  f.hat <- (1/(N+1))
}
ComputeKappaEstimate <- function(x) {
  x <- x$N

  N <- sum(x)
  alpha <- sum(x == 1L)
  kappa <- (alpha+1)/(N+1)

  brenner <- list()
  f.hat <- (1-kappa)/(N+1)

  brenner[[1]] <- f.hat
  brenner[[2]] <- kappa
  return(brenner)
}
require(disclapmix)
# Input: Haplotype data x
# Output: Estimate of singleton of interest frequency
```

```
ComputeDisclapEstimate <- function(x) {
  k <- 1
  x.mat <- as.matrix(x[rep(1L:nrow(x), x$N),
                       1L:(ncol(x)-k)])
  possible.fits <- disclapmix(x.mat, cluster = 1L)
  #(ncol(x.mat)
  possible.fits <- lapply(1L:5L, function(clusters) {
    fit <- disclapmix(x.mat, clusters = clusters, verbose = 0L,
                      iterations = 200L)
    return(fit)
  })
  BICs <- unlist(lapply(possible.fits, function(fit) fit$BIC_margina
  best.fit.BIC <- possible.fits[[which.min(BICs)]]
  disclap.match.prob <- predict(best.fit.BIC, newdata = as.matrix(su


  f.hat <- disclap.match.prob[length(disclap.match.prob)]
}
# Haplotype frequency estimates based on Good-Turing estimation
# with non-parametric bootstrap (based on function from packege edge
# This is implemented by rewriting the C code for simple good-turing
# http://www.grsampson.net/RGoodTur.html
library(edgeR)
ComputeGoodTuringEstimate <- function(x) {
  x <- x$N
  freq <- goodTuring(x,5)
  f.hat <- freq$proportion[2]
}
```

5.2.2 Simulate data

```
library(fwsim)
require(disclap)
require(disclapmix)
# Functions for generating populations and samples simulated under F
# by using the package 'fwsim'
SimulatePopulation <- function(r, mut) {
  r <- as.integer(r)
  g <- 200L # default
  n.null <- 100000L
  n.end <- 1000000L

  prop.end <- n.end/n.null
  rho <- exp((1/g)*log(prop.end))

  # Founder haplotype median of Y-STR data danes
  temp.data <- data(danes)

  hap.mean <- round(as.vector((apply(danes[,1:10], 2, mean))))
  #H.null <- as.integer(sample(hap.median,r, replace=TRUE, prob = (h
  H.null <- as.integer(sample(hap.mean,r, replace=TRUE))
  rm(temp.data)

  mutation.rate <- rep(0.003,r)
  sim <- fwsim(G = g, H.null, N0 = n.null, mutmodel = mutation.rate,
  pop <- sim$population



  return(pop)
}

# Generate all 12 populations required for simulation experiments
SimulatePopulations <- function() {
  populations <- list()

  mut.models <- list()
  mut1 <- c(0.001,0.00186, 0.00242, 0.00311, 0.00265, 0.00052, 0.00076, 0.0
  mut2 <- c(0.00265, 0.00186, 0.00242, 0.00477, 0.00055, 0.00099, 0.001, 0.
  mut.models[[1]] <- mut1
  mut.models[[2]] <- mut2

  loci <- c(length(mut1), length(mut2))

  idx <- 1
  for (i in 1:2) {
    pop <- SimulatePopulation(loci[i],mut.models[[i]]) # replace with pop g
    populations[[idx]] <- pop
    idx <- idx+1
  }
  return(populations)
}
# Generating samples from a population
# types <- sample(x = 1:nrow(pop), size = n.sample, replace = TRUE,
GenerateSample <- function(x, n.sample) {
  pop <- x

  types <- sample(x = 1:nrow(pop), size = n.sample, replace = TRUE,
  types.table <- table(types)

  sample.dataset <- pop[as.integer(names(types.table)),]
  sample.dataset$N <- as.integer(types.table)
  #sample.dataset <- cbind(sample.dataset,as.integer(types.table))
  #colnames(sample.dataset)[n.col] <- "N"
  #transform(sample.dataset,)
  #sample.dataset$N <- as.numeric(as.vector(types.table))

  s <- list()
  s[[1]] <- sample.dataset
  s[[2]] <- types

  return(s)
}
```

5.2.3 Compute haplotype frequencies based on population

```
require(disclap)
require(disclapmix)
library(fwsim)
require(tables)
require(ggplot2)
# Function for computng haplotype frequency estimates based on various met
# Input: Population P and sample coverage coverage(sample size)
# Output: Results as described in header
# NOTE, CHANGE FUNCTION $ IT TAKES coverage LOW or HIGH instead of n.size
ComputeResults <- function(P, n.size) {
  result <- data.frame(hap.idx=numeric(), coverage=numeric(), p=numeric(),
                       count=numeric(), brenner=numeric(), gt=numeric(), d
  n.types.pop <- dim(P)[1]
  r <- dim(P)[2]-1
  n.sim <- 50
  for(i in 1:n.sim) {
    sample.computed <- GenerateSample(P,n.size)



    s <- sample.computed[[1]] # sample as data.frame
    t <- sample.computed[[2]] # indices of sampled haplotypes
```

```
unseen.hap <- GenerateSample(P[-t,],1)
h <- unseen.hap[[1]]
unseen.idx <- unseen.hap[[2]]

# add unseen hap to database
s <- rbind(s, h)
unseen.idx.sample <- dim(s)[1] # last row

n.types.sample <- dim(s)[1]
sample.coverage <- n.types.sample/n.types.pop

# singleton proportion in sample
brenner <- ComputeKappaEstimate(s)
kappa <- brenner[[2]]

while(kappa == 1) {
  sample.computed <- GenerateSample(P,n.size)
  s <- sample.computed[[1]] # sample as data.frame
  t <- sample.computed[[2]] # indices of sampled haplotypes

  unseen.hap <- GenerateSample(P[-t,],1)
  h <- unseen.hap[[1]]
  unseen.idx <- unseen.hap[[2]]

  # add unseen hap to database
```

```
      s <- rbind(s, h)
      unseen.idx.sample <- dim(s)[1] # last row

      n.types.sample <- dim(s)[1]
      sample.coverage <- n.types.sample/n.types.pop

      # singleton proportion in sample
      brenner <- ComputeKappaEstimate(s)
      kappa <- brenner[[2]]

    }
    # real population frequency of unseen haplotype x
    p.x <- P$N[unseen.idx]/sum(P$N)

    p.hat.gt <- ComputeGoodTuringEstimate(s)
    p.hat.count <- ComputeCountEstimate(s)
    p.hat.kappa <- brenner[[1]]
    p.hat.disclap <- ComputeDisclapEstimate(s)
    #p.hat.disclap <- 1
    result <- rbind(result, data.frame(hap.idx=unseen.idx, coverage=sample.
                                       brenner=p.hat.kappa, gt=p.hat.gt, d

  }
  return(result)
}
```

5.2.4 Statistical tests

```
OverestimationTest <- function(res) {
  # Test for overestmation
  R.all.LR <- res
  t.overest <- list()
  t.brenner <- list()
  t.gt <- list()
  t.disclap <- list()

  c1 <- 0
  c2 <- 0
  for(i in 1:6) {
    c1 <- (i-1)*50 + 1
    c2 <- c2 + 50
    LR <- R.all.LR$LR.pop[c1:c2]
    LR.brenner <- R.all.LR$LR.brenner[c1:c2]
    tt.brenner <- t.test(LR.brenner, LR, paired = TRUE, alternative =
    t.brenner[[i]] <- tt.brenner

    LR.gt <- R.all.LR$LR.gt[c1:c2]
    tt.gt <- t.test(LR.gt, LR, paired = TRUE, alternative = ”greater
    t.gt[[i]] <- tt.gt



    LR.disclap <- R.all.LR$LR.disclap[c1:c2]
    tt.disclap <- t.test(LR.disclap, LR, paired = TRUE, alternative = "grea
    t.disclap[[i]] <- tt.disclap
  }
  t.overest[[1]] <- t.brenner
  t.overest[[2]] <- t.gt
  t.overest[[3]] <- t.disclap

  test.results <- OverestimationTestExtract(t.overest)
  return(test.results)
}



estimatorMSE <- function(res) {
  test.results <- data.frame(MSE.KM=numeric(), var.KM=numeric(),
                             MSE.GTE=numeric(), var.GTE=numeric(), MSE.DLM=

  R.all.LR <- res

  c1 <- 0
  c2 <- 0
  for(i in 1:6) {
    c1 <- (i-1)*50 + 1
    c2 <- c2 + 50
    LR <- R.all.LR$LR.pop[c1:c2]
    LR.brenner <- R.all.LR$LR.brenner[c1:c2]
    SE.brenner <- (LR.brenner-LR)^2
    MSE.brenner <- mean(SE.brenner)
    var.brenner <- var(SE.brenner)

    LR.gt <- R.all.LR$LR.gt[c1:c2]
    SE.gt <- (LR.gt-LR)^2
    MSE.gt <- mean(SE.gt)
    var.gt <- var(SE.gt)

    LR.disclap <- R.all.LR$LR.disclap[c1:c2]
    SE.disclap <- (LR.disclap-LR)^2
    MSE.disclap <- mean(SE.disclap)
    var.disclap <- var(SE.disclap)
    test.results <- rbind(test.results, data.frame(MSE.KM=MSE.brenne
                         MSE.GTE=MSE.gt, var.GTE=var.gt, MSE.DLM=MS
  }

  n.s <- as.factor(c(500,500,2000,2000,8000,8000))
  n.r <- as.factor(c(8,11,8,11,8,11))
  test.results <- cbind(test.results, n.sample=n.s, n.loci=n.r)
  test.results <- test.results[c(7,8,1:6)]
  rownames(test.results) <- 1:6
  return(test.results)
}
```

```
# Two sided t-test to test for sign differences between the estimators in t
SignDiffMSETest <- function(res) {
  R.all.LR <- res
  t.MSE <- list()
  MSE.brenner.gt <- list()
  MSE.brenner.disclap <- list()
  MSE.gt.disclap <- list()

  c1 <- 0
  c2 <- 0
  for(i in 1:6) {
    c1 <- (i-1)*50 + 1
    c2 <- c2 + 50
    LR <- R.all.LR$LR.pop[c1:c2]

    LR.brenner <- R.all.LR$LR.brenner[c1:c2]
    SE.brenner <- (LR.brenner-LR)^2

    LR.gt <- R.all.LR$LR.gt[c1:c2]
    SE.gt <- (LR.gt-LR)^2

    LR.disclap <- R.all.LR$LR.disclap[c1:c2]
    SE.disclap <- (LR.disclap-LR)^2



    tMSE.brenner.gt <- t.test(SE.brenner, SE.gt, paired = TRUE, mu=0
    tMSE.brenner.disclap <- t.test(SE.brenner, SE.disclap, paired =
    MSE.brenner.gt[[i]] <- tMSE.brenner.gt
    MSE.brenner.disclap[[i]] <- tMSE.brenner.disclap

    tMSE.gt <- t.test(SE.gt, SE.disclap, paired = TRUE, mu=0)
    MSE.gt.disclap[[i]] <- tMSE.gt
  }
  t.MSE[[1]] <- MSE.brenner.gt
  t.MSE[[2]] <- MSE.brenner.disclap
  t.MSE[[3]] <- MSE.gt.disclap
  test.results <- SignDiffMSETestExtract(t.MSE)

}
# Investigating factors important for MSE differences
OneFactorMSETest <- function(res) {
  t.MSE <- list()
  MSE.brenner <- list()
  MSE.gt <- list()
  MSE.disclap <- list()
  c1 <- 0
  c2 <- 0
  for(i in 1:6) {
    c1 <- (i-1)*50 + 1
    c2 <- c2 + 50
    LR <- R.all.LR$LR.pop[c1:c2]

    LR.brenner <- R.all.LR$LR.brenner[c1:c2]
    SE.brenner <- (LR.brenner-LR)^2

    LR.gt <- R.all.LR$LR.gt[c1:c2]
    SE.gt <- (LR.gt-LR)^2

    LR.disclap <- R.all.LR$LR.disclap[c1:c2]
    SE.disclap <- (LR.disclap-LR)^2

  }
  t.MSE[[1]] <- MSE.brenner.gt
  t.MSE[[2]] <- MSE.brenner.disclap
  t.MSE[[3]] <- MSE.gt.disclap
  return(t.MSE)
}
```

[language=R]

Norges miljø- og biovitenskapelig universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway