

To the Graduate Council:

I am submitting herewith a thesis written by Alex Brian Haun entitled "Automated Discovery of Pedigrees and Their Structures in Collections of STR DNA Specimens Using a Link Discovery Tool." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Engineering.

J. D. Birdwell, Major Professor

We have read this thesis
and recommend its acceptance:

Tsewei Wang

David Icove

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

**Automated Discovery of Pedigrees
and Their Structures in
Collections of STR DNA Specimens
Using a Link Discovery Tool**

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee

Alex Brian Haun
May 2010

Dedication

This thesis is dedicated to my wife

Sarah Haun

and my parents

Michael Haun

and

Sharon Baroody

who have given me unconditional and unwavering love and support.

Acknowledgements

My time at the University of Tennessee has been very challenging and very rewarding, and many people have been vital to my success, both academic and personal, while in Knoxville. I would like to thank my advisor Dr. J. Douglas Birdwell who gave me the opportunity to work on challenging and inspiring projects in the Laboratory for Information Technologies, for his guidance and advice. I would like to thank Dr. Tsewei Wang, one of my thesis committee members, for always being enthusiastic about providing opportunities to learn. I would also like to thank Dr. David Icove, the other member of my thesis committee, for his advice and support.

I would like to thank my brother Evan Haun, my sister Tessa Lott, my step-parents Janet Haun and Theodore Barody, my grand-parents Robert and Nell Faust, my in-laws Allen and Ellen Johnston, all of my wonderful family and friends, and especially my little boy Michael Allen Haun, for all of your support and encouragement.

Abstract

In instances of mass fatality, such as plane crashes, natural disasters, or terrorist attacks, investigators may encounter hundreds or thousands of DNA specimens representing victims. For example, during the January 2010 Haiti earthquake, entire communities were destroyed, resulting in the loss of thousands of lives. With such a large number of victims the discovery of family pedigrees is possible, but often requires the manual application of analytical methods, which are tedious, time-consuming, and expensive. The method presented in this thesis allows for automated pedigree discovery by extending Link Discovery Tool (LDT), a graph visualization tool designed for discovering linkages in large criminal networks. The proposed algorithm takes advantage of spatial clustering of graphs of DNA specimens to discover pedigree structures in large collections of specimens, saving both time and money in the identification process.

Table of Contents

Chapter 1: Introduction	1
1.1 Summary of the Findings of This Work	1
1.2 Identification of Human Remains using DNA.....	1
1.3 STR Data: Loci and Alleles	2
1.4 Specimen Relationships and Likelihood Ratios.....	2
1.5 Link Discovery Tool.....	3
1.6 Need for an Automated Approach for Discovering Pedigrees	4
1.7 Statement of Research Work.....	5
Chapter 2: Background Review.....	6
2.1 Genetic Pedigrees.....	6
2.2 Likelihood Ratios for Relatedness.....	8
2.3 Basic Graph Terminology	11
Chapter 3: Problem Formulation and Solution Implementation	13
3.1 Extension of Link Discovery Tool to Accommodate Pedigree Discovery	13
3.2 Expected Difficulties.....	14
3.3 Link Discovery Tool Physics Model.....	15
3.4 Pedigree Structures.....	19
3.5 Generation of Data Sets.....	20
3.5.1 Preparation of Allele Frequency Distributions	20
3.5.2 Generation of Independent Specimens.....	23
3.5.3 Generation of Pedigree Specimens.....	24
3.6 Study: Feasibility of Relationship Discovery Using STR Data	25
3.6.1 Objective	28
3.6.2 Formulation	28
3.6.3 Implementation	29
3.6.4 Interpretation of Results.....	29
3.7 Study: Feasibility of Relationship Discovery Using Likelihood Ratios.....	32
3.7.1 Objective	32
3.7.2 Formulation	32
3.7.3 Implementation	33
3.7.4 Interpretation of Results.....	33
3.8 Pedigree Discovery Algorithm	37
3.8.1 Overview.....	37
3.8.2 Snip Algorithm.....	38
3.8.3 Analysis Algorithm	39
3.8.4 Discovery Algorithm	42
Chapter 4: Results and Discussion	46
4.1 Experiment: Discovery of Individuals	46
4.1.1 Objective	46
4.1.2 Results.....	46
4.2 Experiment: Discovery of Pedigrees.....	47
4.2.1 Objective	47

4.2.2 Results.....	49
4.3 Experiment: Influence of Sibling Edges	50
4.3.1 Objective	50
4.3.2 Results.....	54
4.4 Distribution of Likelihood Ratio Values	57
4.5 Experiment: Influence of Edge Weight Threshold Value	60
4.5.1 Objective	60
4.5.2 Results.....	62
Chapter 5: Conclusion and Future Work.....	70
5.1 Conclusion	70
5.2 Future Work.....	71
References.....	73
Appendix A: Tables and Derivations for Likelihood Ratios	76
A.1 Probability of Sharing IBD Alleles	77
A.2 Probabilities and Likelihood Ratios for a Parent-Child Relationship	78
A.3 Probabilities and Likelihood Ratios for a Full-Sibling Relationship	80
Appendix B: Pedigree Classification Decision Rules.....	83
Vita.....	89

List of Tables

Table 1: Descriptions and labels of target pedigrees.	21
Table 2: Algorithm for pseudo-randomly generating specimens.	25
Table 3: Algorithm for forming an offspring specimen by mating parent specimens.	26
Table 4: Algorithm for generating pedigrees of a specified type.	27
Table 5: Snip algorithm for forming viable candidate clusters.....	40
Table 6: Pedigree characteristic for decision rules in the analysis algorithm.	43
Table 7: Analysis algorithm for pedigree classification.	44
Table 8: Pedigree discovery algorithm.	45
Table 9: Coefficients of polynomial regressions for the number of PC-LR and Sib-LR edges.....	49
Table 10: Percentage of Type I and Type II error as Log-LR threshold value varies between -2.0 and 2.0.....	60
Table 11: Comparison of LR distributions from Specimen Generator and Buckleton [6].	61
Table 12: Probabilities that two individuals share a number of IBD alleles given a relationship [6].	77
Table 13: Conditional probabilities for the parent-child relationship from Buckleton [6].	79
Table 14: Likelihood ratios for the parent-child relationship from Buckleton [6].	80
Table 15: Conditional probabilities for the full-sibling relationship from Buckleton [6].	81
Table 16: Likelihood ratios for the full-sibling relationship from Buckleton [6].	82
Table 17: Pedigree characteristic for decision rules in the analysis algorithm.	86
Table 18: Pedigree characteristics for classifying clusters with additional or missing edges.	87

List of Figures

Figure 1: 2D graphical representation of a pedigree.....	8
Figure 2: 3D representation of two pedigree trio clusters.	9
Figure 3: Graphical representations of the types of pedigrees used.....	22
Figure 4: Graph of one specimen and associated STR profile.....	30
Figure 5: Graph of PPC trio from a distance (left) and close up of cluster of related specimens (right).....	30
Figure 6: Graph of 10 PPC trios from a distance (left) and close up of potentially related specimens (right).	31
Figure 7: Graph of 100 PPC trios using LR link weights.....	34
Figure 8: Graph of 100 PPC trios using Log-LR link weights.....	35
Figure 9: Graphs of pedigree trio collections of various sizes.....	36
Figure 10: Discovery accuracy of individuals as collection size increases.	48
Figure 11: Log of number of PC-LR and Sib-LR edges as collection size increases.	48
Figure 12: Quadratic polynomial regression of number of Sib-LR edges as collection size increases.	48
Figure 13: Quadratic polynomial regression of number of PC-LR edges as collection size increases.	49
Figure 14: Results of discovery with collections of 10 pedigrees.	50
Figure 15: Results of discovery with collections of 50 pedigrees.	51
Figure 16: Results of discovery with collections of 100 pedigrees.....	51
Figure 17: Results of discovery with collections of 200 pedigrees.....	52
Figure 18: Results of discovery with collections of 300 pedigrees.....	52
Figure 19: Results of discovery with collections of 400 pedigrees.....	53
Figure 20: Results of discovery with collections of 500 pedigrees.....	53
Figure 21: Accuracy of discovering true pedigree as pedigree collection size increases.	54
Figure 22: Number of PC and Sib edges in graph versus the maximum number of true PC and Sib edges.	55
Figure 23: Graph composition as percentage of PC-LR and Sib-LR edges.	56

Figure 24: Distributions of the log-likelihood ratio for pairs of unrelated individuals and pairs of individuals related as full-siblings.	58
Figure 25: Comparison of Type I and Type II error for the full-sibling relationship.....	59
Figure 26: Close-up comparison of Type I and Type II error for the Log-LR range of -2.5 to 2.5.	59
Figure 27: Distributions of the log-likelihood ratio for pairs of unrelated individuals and pairs of individuals related as full-siblings from Buckleton et al. [6].....	61
Figure 28: Number of PC and Sib edges in graph versus the maximum number of true PC and Sib edges with threshold = 1.0.	64
Figure 29: Graph composition with threshold = 1.0.	65
Figure 30: Accuracy of discovering true pedigree with threshold = 1.0.....	66
Figure 31: Number of PC and Sib edges in graph versus the maximum number of true PC and Sib edges with threshold = 2.0.	67
Figure 32: Graph composition with threshold = 2.0	68
Figure 33: Accuracy of discovering true pedigree with threshold = 2.0.....	69

List of Abbreviations and Symbols

DNA	Deoxyribonucleic Acid
IBD	Identical By Descent
LDT	Link Discovery Tool
Log-LR	Log-Likelihood Ratio
LR	Likelihood Ratio
PC	Parent-Child Relationship, Parent + Child Pedigree Duo
PCC	Parent + 2 Children Pedigree Trio
PPCC	Parent + 3 Children Pedigree Quad
PC-LR	Parent-Child Likelihood Ratio
PPC	2 Parent + Child Pedigree Trio
PPCC	2 Parent + 2 Children Pedigree Quad
PPCCC	2 Parent + 3 Children Pedigree Quint
Sib-LR	Full-Sibling Likelihood Ratio
SS	2 Full-Sibling Pedigree Duo
SSS	3 Full-Sibling Pedigree Trio
STR	Short Tandem Repeat

Chapter 1: Introduction

1.1 Summary of the Findings of This Work

The pedigree discovery algorithm presented in this work was designed as an extension of the Link Discovery Tool (LDT) [1], making use of LDT's physics-based graph clustering capabilities. The primary contribution of this work is the ability to automate the discovery of pedigrees using the snip and analysis algorithms. The automated pedigree discovery algorithm was tested using pseudo-randomly generated data collections consisting of individual specimens and pedigrees of various types and was shown to be able to discover each type of pedigree. The choice of an edge weight threshold had a drastic effect on the number of full-sibling edges in the graph, which impact the performance of the algorithm. Increasing the threshold value reduced the number of false full-sibling edges and increased the performance of the algorithm.

1.2 Identification of Human Remains using DNA

Over the last decade DNA typing has become the cornerstone of forensic genetics, providing investigators with an increasingly reliable tool to accurately identify human biological remains [2,3]. While humans share around 99.9% of their genetic code with each other, the remaining 0.1% of the genetic code is highly variable between individuals [2]. It is this variability that provides us with the discriminatory power necessary to distinguish between the profiles of two or more

individuals and allows us to genetically identify otherwise unrecognizable human remains [3].

1.3 STR Data: Loci and Alleles

Most human cells contain a complete copy of the human genome, which is organized into twenty-three pairs of chromosomes; each parent provides one chromosome of each pair. Each chromosome is a chain of base pairs (composed of adenine, cytosine, guanine, and thymine) arranged in a sequence [3, 4]. Several regions (loci, plural, and locus, singular) of each chromosome contain sequences of repeated patterns, which are known as alleles. Variations (or polymorphisms) in alleles are due to different numbers of repeated units composed of between one and six base pairs [2,4] and give an allele its name; for example, an allele containing eight repeated units is called “allele 8”. A genetic profile contains two alleles at each locus; one allele is identical by descent (IBD) from each parent. These regions are known as short tandem repeats (STR) and are currently the most commonly analyzed polymorphisms in forensic genetics [2]. Profiles studied in this work consist of alleles from the FBI’s thirteen core STR loci, which are used in the national DNA database and are permissible to be presented in both state and federal courts [5].

1.4 Specimen Relationships and Likelihood Ratios

The principle of Mendelian inheritance specifies that offspring receive half of their genetic material from each parent [6]. We expect that each offspring shares one allele IBD per locus with each parent (assuming no mutation has occurred); this information can be used to formulate a likelihood ratio (LR) that provides a measure

of the strength of the hypothesis that two specimens are related by a specified relationship (such as parent-child or full-sibling) against the hypothesis that two specimens are unrelated [6]. For example, the larger the likelihood ratio for a parent-child relationship between two specimens, the more likely it is that the specimens are related as a parent and child. The conditional probabilities representing the hypotheses in the ratio can be derived using the methods of Balding and Nichols [7] or Weir [8] for a given relationship [6]. Profiles in this work are compared using likelihood ratios for parent-child and full-sibling relationships. The specimens and likelihood ratios provide the input for using Link Discovery Tool.

1.5 Link Discovery Tool

The Link Discovery Tool (LDT) is a visualization tool that is designed to display data links between information entities as a graph in three dimensions. The tool was tailored and optimized for use as a criminal investigation tool using entities such as persons, pieces of evidence, businesses, etc. and links such as telephone calls, or bank records [1]. Entities are treated as particles that are naturally repulsive, and links are treated as attractive forces that tend to pull related entities together. The result is a three-dimensional graph in which unrelated entities have been pushed apart, while related entities have been arranged into clusters. These clusters become the focus of link analysis, and graph algorithms are used to discover paths between entities within a cluster or between entities in different clusters. The Link Discovery Tool was re-implemented in the present work and extended to include the features and algorithms needed for pedigree discovery.

As an example application of this tool, consider a data set consisting of criminal associates and known communications between them. The resulting clusters can represent criminal groups or organizations. An outlier of the clusters

may represent a gang member involved in a crime, whereas the interior entities may represent the organization's leaders or inner circle. By discovering the clusters of entities in the information, law enforcement agencies may be able to more efficiently target the leaders of criminal organizations rather than spending time chasing lower-ranking criminals [1].

The specimens and likelihood ratios generated in this work provide the input for Link Discovery Tool, where the specimens are the LDT entities and the likelihood ratios are the weights of the links connecting the entities. Specimens are associated with STR DNA profiles with data at the 13 core loci. Amelogenin, Y-STR DNA, Mitochondrial DNA, and metadata are not included in the specimen information.

1.6 Need for an Automated Approach for Discovering Pedigrees

In instances of mass fatality, such as a plane crash or a tsunami, investigators may encounter a large collection of unidentified DNA specimens representing hundreds or thousands of victims. Many of the specimens in the collection will be unrelated, but a small number of them may be related; for example, families tend to fly together while on vacation. In the case of a major natural disaster, such as the recent earthquake in Haiti where entire communities were destroyed, the profile collection may contain many small groups of genetic profiles from related individuals (the missing families). Software tools are available to determine the likelihood or possibility that two or more people share given relationships, but these methods are tedious on a large scale (for example, LeClair, et al. have described a method of discovering pedigree trios that requires exhaustive search of the solution space [9]). The ability to automatically identify the relationships among the pedigrees in a collection could save both time and money in the identification process. The pedigree

discovery algorithm presented in this thesis outlines an automated method of discovering pedigrees and their structures in collections of DNA specimens.

1.7 Statement of Research Work

Link Discovery Tool has been useful in discovering links between members of a criminal network [1]. It is natural to desire a straightforward extension of LDT that provides the ability to discover pedigree structures in a large collection of specimens. This thesis documents the results of researching and implementing the functionality to automatically discover pedigree structures in large mixtures of specimens, based on the original physics model and graph algorithms available in LDT. Difficulties, both expected and discovered, in this extension of functionality are discussed. The modifications and algorithms added to LDT are explored along with recommendations for the development of future versions of Link Discovery Tool.

In Chapter 2, a brief overview of the related literature is presented. In Chapter 3, a formulation of the research problem is presented along with a solution implementation. Chapter 4 presents and discusses the results of the experiments and the effectiveness of the pedigree discovery algorithm. Finally, Chapter 5 presents the conclusion of the research work along with recommendations for future research.

Chapter 2: Background Review

2.1 Genetic Pedigrees

A genetic pedigree describes a group of individuals that are biologically related, such as a set of two individuals that are related as full-siblings (the two individuals share both parents) [10]. When represented as a directed graph, the pedigree can be defined as a non-rooted tree that contains member nodes, mating nodes, and edges [10]. The member nodes are the DNA specimens, and the mating nodes are associated with a mating pair between a male member node and a female member node. Each edge links either a member node *to* a mating node, or a mating node *to* a member node. Edges connecting two member nodes or two mating nodes are not valid. Pedigrees can have certain characteristics that differentiate them from trees in graph theory. For example, pedigrees may contain cycles, which can occur due to inbreeding, marriage/divorce, polygamy, etc. Cannings et al. have described many of the cycles that can occur in natural pedigrees, such as mating cycles, inbreeding cycles, and exchange cycles, which can add considerable complexity to the analysis of the pedigree [10].

Pedigree graphs have two graphical representations in this thesis. The first representation shows the graph drawn in two-dimensions. This 2D representation follows the pedigree directed graph and outlined above and contains member nodes, mating nodes, and edges. In two-generation pedigrees parent member nodes have an in-degree of zero and an out-degree of one, where the out-going edge goes from the member node to a mating node. Offspring member nodes have an in-degree of one and an out-degree of zero, where the in-coming edge goes from the mating node to the member node. Mating nodes have an in-degree of zero, one, or two (when the

pedigree contains zero, one, or two parents, respectively) and an out-degree of one, two, or three (when the pedigree contains one, two, or three offspring, respectively). The in-coming edges go from parent member nodes to the mating node and the out-going edges go from the mating node to the offspring member nodes. The parent nodes are drawn on a higher level than the mating nodes, and the offspring nodes are drawn on a lower level. Sets of edges in this type of representation using either one in-coming mating node edge and one out-going mating node edge or two out-going mating node edges represent *actual* biological relationships. For example, a set of edges with an edge going from the parent to the mating node and an edge going from the same mating node to an offspring node imply a parent-child relationship between the parent and offspring nodes; a set of edges with an edge going from a mating node to an offspring node and an edge going from the same mating node to a different offspring node imply a full-sibling relationship between the offspring nodes. This representation provides an intuitive view of the pedigree and is the representation used by Cannings [10]. Figure 1 shows an example 2D representation of a pedigree containing two parents and two children.

The second pedigree representation shows the pedigree graph in three-dimensions. This is the default representation in Link Discovery Tool and contains only the member nodes and edges between member nodes. The edges in this representation are different than the edges in the 2D representation, which does not allow edges between member nodes. In this representation, the edges depict a *potential* biological relationship between the member nodes at the ends of the edge. The edges are undirected and have an associated type, which represents the type of the potential relationship, such as a parent-child relationship or a full-sibling relationship. The edges also have an associated weight, which represents the strength of the potential relationship. This representation is useful in LDT because LDT can use the edge weights to cluster related members together in the 3D space. Figure 2 shows a 3D graph that contains two clusters. Each cluster represents one

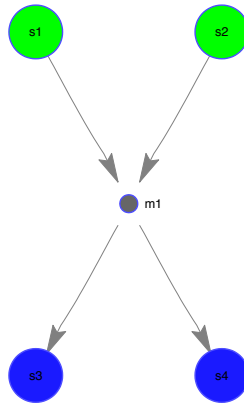


Figure 1: 2D graphical representation of a pedigree containing two parents and two children.

pedigree trio. The Euclidean distance between members of a trio is small, while the distance between the trio clusters is relatively large.

2.2 Likelihood Ratios for Relatedness

Likelihood ratios are used in this work as weights applied to edges between vertices in the graph. LDT treats these weights as spring constants used to pull linked vertices closer together. LDT treats un-linked vertices as though they were mutually repulsive particles, using a negative gravity to push the vertices further apart. Since the vertices represent DNA specimens, the presence of an edge between two vertices represents a hypothetical biological relationship between the vertices, and the weight of the edge represents the strength of the hypothesized relationship. The absence of an edge between two vertices implies that there is no biological relationship between the two vertices (or at least that the strength of the relationship is not strong enough to exceed a preset threshold value).

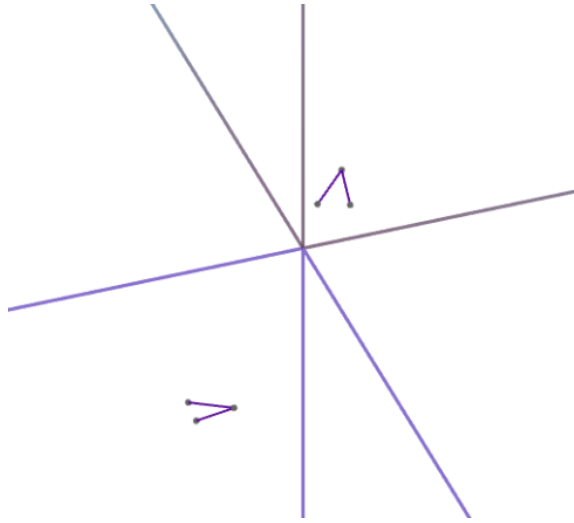


Figure 2: 3D representation of two pedigree trio clusters.

Likelihood ratios are used to assess the weight of the DNA evidence under two mutually exclusive hypotheses [3,6]. Here, the evidence comes from the specimen profiles, and the hypotheses are:

- H_1 : the two individuals being compared are related by
the stated relationship (parent-child or full-sibling)
- H_0 : the two individuals are unrelated

The likelihood ratio comes from Bayes' Theorem, which states:

posterior odds = likelihood ratio x prior odds

$$\frac{\Pr(H_1 | E)}{\Pr(H_0 | E)} = \frac{\Pr(E | H_1) \Pr(H_1)}{\Pr(E | H_0) \Pr(H_0)}$$

where H_0 and H_1 are the hypotheses stated above, and E is the evidence under scrutiny [6]. The likelihood ratios in this work are pair-wise comparisons of two individuals from the collection and can be expressed as follows:

$$LR = \frac{\Pr(E | H_1)}{\Pr(E | H_0)} = \frac{\Pr(G_{S1}, G_{S2} | H_1)}{\Pr(G_{S1}, G_{S2} | H_0)}$$

where G_{S1} and G_{S2} are the genotypes of specimens S_1 and S_2 , H_1 is the hypothesis that S_1 and S_2 are related by either a parent-child relationship (H_{PC}) or a full-sibling relationship (H_S), and H_0 is the hypothesis that S_1 and S_2 are not related. The likelihood ratio can be simplified as follows:

$$LR = \frac{\Pr(G_{S1}, G_{S2} | H_1)}{\Pr(G_{S1}, G_{S2} | H_0)}$$

$$LR = \frac{\Pr(G_{S1} | G_{S2}, H_1) \Pr(G_{S2} | H_1)}{\Pr(G_{S1} | G_{S2}, H_0) \Pr(G_{S2} | H_0)}$$

The second equation above is a result of applying the definition of conditional probability, $P(A, B) = P(A | B)P(B)$, to the likelihood ratio [3]. The right-hand term in the equation can be removed because the genotype of specimen S_2 does not depend on whether or not S_1 and S_2 are related, so both probabilities are the same. The denominator of the left-hand term can be simplified by recognizing that the genotype of specimen S_1 is independent of specimen S_2 's genotype when the two are not related [6]. These simplifications result in:

$$LR = \frac{\Pr(G_{S1} | G_{S2}, H_1)}{\Pr(G_{S1} | H_0)}$$

The denominator in the equation above is the probability of observing the genotype G_{S1} at random and is a straightforward calculation. The numerator can be

evaluated by using the Method of Balding and Nichols (see section A.1) and assuming Mendelian inheritance, no mutation, and no inbreeding:

$$\begin{aligned}\Pr(G_{S_1} | G_{S_2}, H_1) &= \Pr(G_{S_1} | G_{S_2}, Z_2, H_1) \Pr(Z_2 | H_1) + \Pr(G_{S_1} | G_{S_2}, Z_1, H_1) \frac{1}{2} \Pr(Z_1 | H_1) \\ &\quad + \Pr(G_{S_1} | G_{S_2}, Z_1, H_1) \frac{1}{2} \Pr(Z_1 | H_1) + \Pr(G_{S_1} | G_{S_2}, Z_0, H_1) \Pr(Z_0 | H_1)\end{aligned}$$

Buckleton et al. have presented the conditional probabilities in the equation above for all possible genotype combinations at a single locus [6]. As an example, consider the genotypes $G_{S_1} = aa$ and $G_{S_2} = ab$. The probability of specimens S_1 and S_2 being related as a parent and child can be shown as:

$$\Pr(G_{S_1} | G_{S_2}, H_{PC}) = \frac{1}{2} \Pr(Z_1 | H_{PC}) p_b + \frac{1}{2} \Pr(Z_1 | H_{PC}) p_b = p_b$$

and the likelihood ratio follows as:

$$LR = \frac{\Pr(G_{S_1} | G_{S_2}, H_{PC})}{\Pr(G_{S_1} | H_0)} = \frac{p_a}{p_a^2} = \frac{1}{p_a}$$

Appendix A shows the conditional probabilities and likelihood ratio calculations presented by Buckleton for all possible genotype combinations for the parent-child and full-sibling relationships [6].

2.3 Basic Graph Terminology

Several fundamental terms and concepts of graph theory are relied upon in this work and are noted below. A graph $G = \{V, E\}$ is defined as an object consisting of two sets; the set V is the vertex set, which is a non-empty set of elements and the set E is the edge set, which may be empty or may contain elements that are two-

element subsets of the vertex set [11]. The vertex set used in Link Discovery Tool represents STR DNA specimens, while the edge set represents potential biological relationships between the pair of specimens connected by an edge.

A graph in which there is a path of edges between each pair of vertices is said to be connected, and a disconnected graph consists of a set of connected sub-graphs, called components [11]. Components are useful in the algorithm presented in the next chapter because the edges represent potential biological relationships and the absence of an edge between two specimens implies that they are not biologically related as either parent-child or as full-siblings. Therefore, the graph can be divided into components and each component can be analyzed independently.

An edge that, if removed, would divide a component into two components is called a cut edge [11]. The ability to search a component and eliminate cut edges is a key concept in the pedigree discovery algorithm. Combined with graph clustering, the elimination of cut edges allows weak relationships in large components to be removed, dividing the larger component into smaller, more strongly related components.

A cluster in a graph is a subset of vertices that are related by a measure, usually connectivity (the number of edges between the vertices in the cluster) or distance. Clusters in LDT are defined as a set of vertices in which the distance between a pair of vertices in the cluster is less than the distance between a vertex in the cluster and any vertex outside of the cluster. Link Discovery Tool clusters vertices such that vertices inside the cluster have strong potential biological relationships, and vertices outside of the cluster have weak (or no) potential biological relationships to vertices inside the cluster. This behavior tends to push specimens with weak potential relationships away from clusters of specimens containing relatively strong potential relationships.

Chapter 3: Problem Formulation and Solution Implementation

3.1 Extension of Link Discovery Tool to Accommodate Pedigree Discovery

Link Discovery Tool is a visualization tool that is designed to display data links between information entities as a three-dimensional graph. The tool was designed as a criminal investigation tool using information entities such as persons of interest, businesses, or pieces of evidence, and links such as telephone calls, or bank records [1]. Entities in the data set are treated as vertices in the graph and links as edges between vertices. A physics-based model is applied to the graph in which vertices are naturally repulsive towards each other and links act as an attractive force between vertices. The result is a graph containing clusters of strongly related vertices. The clusters are the focus of link analysis and graph algorithms can be applied to discover intra- and inter-cluster paths.

The work in this thesis re-implements and extends Link Discovery Tool by adding automated pedigree discovery capability. The motivation of the pedigree discovery functionality is a mass disaster, such as a tsunami or plane crash, in which hundreds or thousands of DNA specimens may be recovered and investigators want to discover potential biological relationships among the victims. In the case of a plane crash, families may be included in the recovered remains since families often travel together on vacation. A single family doesn't typically fill up an entire airplane, so there are likely to be other individuals or families among the victims. Analysis by hand is both time consuming and expensive. An automated approach to pedigree discovery would save both time and money, allowing investigators to focus on other efforts.

The algorithm presented here attempts to provide a fast method of discovering and reporting pedigree structures in collections of specimens. It is difficult to be certain of the results when a pedigree has been discovered, but in cases of ambiguity and high-degrees of uncertainty, the algorithm attempts to produce a pedigree containing most of the true members and reduce the set of other potential members for further investigation.

3.2 Expected Difficulties

Several obstacles were encountered during the development of the pedigree discovery algorithm. The first obstacle stems from the nature of Link Discovery Tool, namely that it relies upon a physics model to cluster potentially related vertices together while pushing unrelated vertices farther apart. The difficulty lies in the fact that a spatial clustering algorithm relies on the availability of a sufficient amount of time to determine the locations of the graph vertices. For optimal results, locations at or near a steady-state configuration are desired. Using an incremental model (such as a physics model) can require a great deal of time with a very large graph. This obstacle predominantly inhibits the performance in terms of execution time of the algorithm, although the performance in terms of discovery accuracy can be degraded if the physics aren't sufficiently applied to the graph prior to discovery.

The second obstacle stems from the nature of full-sibling relationships, in which two individuals can share zero, one, or two alleles IBD. This means that a large data set is likely to contain a large number of full-sibling links that represent false sibling relationships. For example, one test data set contains 1,000 un-related specimens but 9,525 full-sibling links. Some of these links have weights that are quite high (such as a log-likelihood ratio value of 4.0, which corresponds to the probability of a full-sibling relationship between the individuals being 10,000 times more likely

than if the individuals are unrelated), but are incorrect since no individual in the collection has a relative that is also present in the collection. These edges, therefore, make it difficult to differentiate between true relationships and false relationships. This obstacle dominates among the various impacts upon accuracy of the pedigree discovery process and manifests as a false positive error rate that is proportional to the number of individual nodes in the graph.

3.3 Link Discovery Tool Physics Model

The physics model used by Link Discovery Tool treats information entities (vertices in the graph) as particles that are naturally repulsive to each other. Links (edges in the graph) between pairs of information entities are treated as attractive forces that tend to pull linked entities together. The result is a clustered graphical representation of the data in which related entities (those connected by links) are located closer together in space and un-related entities (those not connected by links, or those connected by weak links) are located farther apart.

For pedigree discovery, the vertices are the DNA specimens and the edges are potential biological relationships (either parent-child or full-sibling) between pairs of specimens. The edges are weighted using the value of the log-likelihood ratio for the specified relationship. The effect of a Log-LR weight is to strengthen (if the value is greater than 1) or weaken (if the value is less than 1) the type of relationship described by the edge associated with the weight, which tends to pull more strongly related specimens closer together. The application of the physics model on a collection of specimens results in one or more clusters containing specimens that are more likely to be related to each other than to other specimens outside of the cluster. The snip algorithm uses this information to reduce large clusters into smaller clusters that can be classified as a specific type of pedigree.

The physics calculations are described as follows. Each vertex is independently assigned a random initial position, P_0^i :

$$P_0^i = \begin{bmatrix} x_0^i \\ y_0^i \\ z_0^i \end{bmatrix}$$

The position update can be calculated by summing the forces acting on vertex i :

$$P_{t+1}^i = P_t^i + \frac{1}{2} \Delta T^2 \sum_{j \neq i} \bar{F}_T^{ij}$$

$$\Delta p_{t+1}^i = \frac{1}{2} \Delta T^2 \sum_{j \neq i} \bar{F}_T^{ij}$$

where $\bar{F}_T^{i,j}$ is the sum of the forces acting on vertex i from vertex j :

$$\bar{F}_T^{ij} = \bar{F}_C^{ij} + \bar{F}_A^{ij} - \bar{F}_R^{ij}$$

where F_C is a weighted connective force (occurring between two connected vertices) that acts like a spring, F_A is a small attractive force that acts like weak gravity, and F_R is a repulsive force that acts as anti-gravity. The connective force is calculated by:

$$\bar{F}_C^{ij} = \bar{u}_{ij} w_{ij} d_{ij} C_C$$

where $\bar{u}_{ij}$ is a unit vector in the direction of vertex i to vertex j , w_{ij} is the weight of the edge between i and j , d_{ij} is the distance between i and j , and C_C is a positive constant strength value of the connective force. The attractive force is calculated by:

$$\bar{F}_A^{ij} = \bar{u}_{ij} d_{ij} C_A$$

where C_A is a small positive constant strength value of the attractive force. The repulsive force is calculated by:

$$\bar{F}_{ij}^R = \frac{\bar{u}_{ij} C_R}{d_{ij}^2}$$

where C_R is a positive constant strength value of the repulsive force. The connective force was chosen to be proportional to the distance between two vertices so that the force would pull the vertices together strongly when they are far apart and weakly when they are closer together. The repulsive force was chosen to be inversely proportional to the distance squared so that the force would push the vertices apart strongly when they are closer together and more when they are farther apart. The attractive force was chosen to be proportional to the distance with a small strength to keep graph components from “flying apart” indefinitely.

To show how the configuration of the graph approaches a steady state, consider two states of connectivity for a graph containing two vertices, i and j . In one state, the vertices are not connected. The steady-state configuration of the graph occurs when the changes in position for both vertices due to the forces between i and j are zero. This state occurs when i and j are separated by a certain finite distance, d_{ij} , which can be found as follows:

$$\begin{aligned} \Delta p_{ij} &= \Delta p_{ji} = 0 \\ \frac{1}{2} \Delta T^2 (\bar{F}_{ij}^A - \bar{F}_{ij}^R) &= \frac{1}{2} \Delta T^2 (\bar{F}_{ji}^A - \bar{F}_{ji}^R) \\ \bar{u}_{ij} d_{ij} C_A - \frac{\bar{u}_{ij} C_R}{d_{ij}^2} &= \bar{u}_{ji} d_{ji} C_A - \frac{\bar{u}_{ji} C_R}{d_{ji}^2} \\ \bar{u}_{ij} (d_{ij} C_A - \frac{C_R}{d_{ij}^2}) &= \bar{u}_{ji} (d_{ji} C_A - \frac{C_R}{d_{ji}^2}) \end{aligned}$$

The expression above can be simplified by noting that $d_{ij} = d_{ji}$ and $\bar{u}_{ji} = -\bar{u}_{ij}$:

$$\begin{aligned}
d_{ij}C_A - \frac{C_R}{d_{ij}^2} &= -d_{ij}C_A + \frac{C_R}{d_{ij}^2} \\
2d_{ij}C_A &= 2\frac{C_R}{d_{ij}^2} \\
d_{ij}C_A &= \frac{C_R}{d_{ij}^2} \\
d_{ij}^3C_A &= C_R \\
d_{ij}^3 &= \frac{C_R}{C_A} \\
d_{ij} &= \sqrt[3]{\frac{C_R}{C_A}}
\end{aligned}$$

In the second state, the vertices are connected by an edge with weight w_{ij} . The steady-state configuration of the graph again occurs when the changes in position for both vertices due to the forces between i and j are zero. This state occurs when i and j are separated by a certain finite distance, d_{ij} , which can be found as follows:

$$\begin{aligned}
\Delta p_{ij} &= \Delta p_{ji} = 0 \\
\frac{1}{2} \Delta T^2 (\bar{F}_{ij}^C + \bar{F}_{ij}^A - \bar{F}_{ij}^R) &= \frac{1}{2} \Delta T^2 (\bar{F}_{ji}^C + \bar{F}_{ji}^A - \bar{F}_{ji}^R) \\
\bar{u}_{ij}w_{ij}d_{ij}C_C + \bar{u}_{ij}d_{ij}C_A - \frac{\bar{u}_{ij}C_R}{d_{ij}^2} &= \bar{u}_{ji}w_{ji}d_{ji}C_C + \bar{u}_{ji}d_{ji}C_A - \frac{\bar{u}_{ji}C_R}{d_{ji}^2} \\
\bar{u}_{ij}(w_{ij}d_{ij}C_C + d_{ij}C_A - \frac{C_R}{d_{ij}^2}) &= \bar{u}_{ji}(w_{ji}d_{ji}C_C + d_{ji}C_A - \frac{C_R}{d_{ji}^2})
\end{aligned}$$

The expression above can be simplified by noting that $d_{ij} = d_{ji}$, $w_{ij} = w_{ji}$, and $\bar{u}_{ji} = -\bar{u}_{ij}$:

$$w_{ij}d_{ij}C_C + d_{ij}C_A - \frac{C_R}{d_{ij}^2} = -(w_{ij}d_{ij}C_C + d_{ij}C_A - \frac{C_R}{d_{ij}^2})$$

$$\begin{aligned}
w_{ij}d_{ij}C_C + d_{ij}C_A - \frac{C_R}{d_{ij}^2} &= -w_{ij}d_{ij}C_C - d_{ij}C_A + \frac{C_R}{d_{ij}^2} \\
2w_{ij}d_{ij}C_C + 2d_{ij}C_A &= 2\frac{C_R}{d_{ij}^2} \\
w_{ij}d_{ij}C_C + d_{ij}C_A &= \frac{C_R}{d_{ij}^2} \\
w_{ij}d_{ij}^3C_C + d_{ij}^3C_A &= C_R \\
d_{ij}^3(w_{ij}C_C + C_A) &= C_R \\
d_{ij}^3 &= \frac{C_R}{w_{ij}C_C + C_A} \\
d_{ij} &= \sqrt[3]{\frac{C_R}{w_{ij}C_C + C_A}}
\end{aligned}$$

The difference between the distances in the two cases above is the term $w_{ij}C_C$, which reduces the distance and pulls connected vertices together, providing the intended behavior. Setting the weight w_{ij} to zero is equivalent to the case when two vertices are not connected by an edge and the equilibrium distance is the same with both equations.

3.4 Pedigree Structures

The pedigrees used in this work range in size from two specimens to five specimens; they are designed to provide a varying degree of difficulty in discovery. Many of the pedigree types contain parent-child relationships (PC), which we expect to be relatively easy to detect in Link Discovery Tool since the child shares one allele IBD with the parent at each locus according to the principle of Mendelian inheritance [6]. Many of the pedigrees also contain sibling relationships, which according to the same principle are more difficult to detect since siblings may share zero, one, or two alleles IBD at a given locus [6]. The pedigrees used are described and labeled in Table

1. In the pedigree tree graphs shown in Figure 3, the nodes drawn in green are the parent specimens, the small gray nodes are the mating nodes, and the blue nodes are the offspring specimens.

3.5 Generation of Data Sets

The research presented in this thesis is dependent upon data collections of various size and composition. The data collections range in size from a single specimen to thousands of specimens and can contain pedigrees of a single type or pedigrees of various types. There are two types of specimens in the data sets: independent specimens and pedigree specimens. The specimens in the data sets contain STR data from the thirteen core loci. Independent specimens were generated independently using the NIST Caucasian American allele frequency distributions [12]. Pedigree specimens were generated in ensemble by creating independent specimens (the founders) and simulating their matings to form offspring specimens according to the principle of Mendelian inheritance [6].

3.5.1 Preparation of Allele Frequency Distributions

The NIST allele frequency distributions of alleles for each of the core loci used in this work were compiled using several hundred random DNA samples from U.S. Caucasian males [5, 12]. The allele frequencies at a specified locus were calculated by counting the number of times an allele was observed and dividing it by the total number of alleles at this locus in the sample population. For example, the frequency of allele A for a given locus is calculated as: $F_A = N_A/2N$, where N_A is the number of occurrences of allele A in the sample population and N is the number of individuals in the sample population [2].

Table 1: Descriptions and labels of target pedigrees.

Type	Label	Structure	Description
0	I	1 Individual	Single specimen with no relatives present in the collection.
1	PC	1 Parent, 1 Child	Pedigree duo consisting of one parent and one biological offspring. The other parent is not included in the pedigree.
2	PCC	1 Parent, 2 Children	Pedigree trio consisting of one parent and two biological offspring. The other parent is not included in the pedigree.
3	PCCC	1 Parent, 3 Children	Pedigree quad consisting of one parent and three biological offspring. The other parent is not included in the pedigree.
4	PPC	2 Parents, 1 Child	Pedigree trio consisting of two parents and one biological offspring.
5	PPCC	2 Parents, 2 Children	Pedigree quad consisting of two parents and two biological offspring.
6	PPCCC	2 Parents, 3 Children	Pedigree quint consisting of two parents and three biological offspring.
7	SS	2 Siblings	Pedigree duo consisting of two full-siblings. The parents are not included in the pedigree.
8	SSS	3 Siblings	Pedigree trio consisting of three full-siblings. The parents are not included in the pedigree.

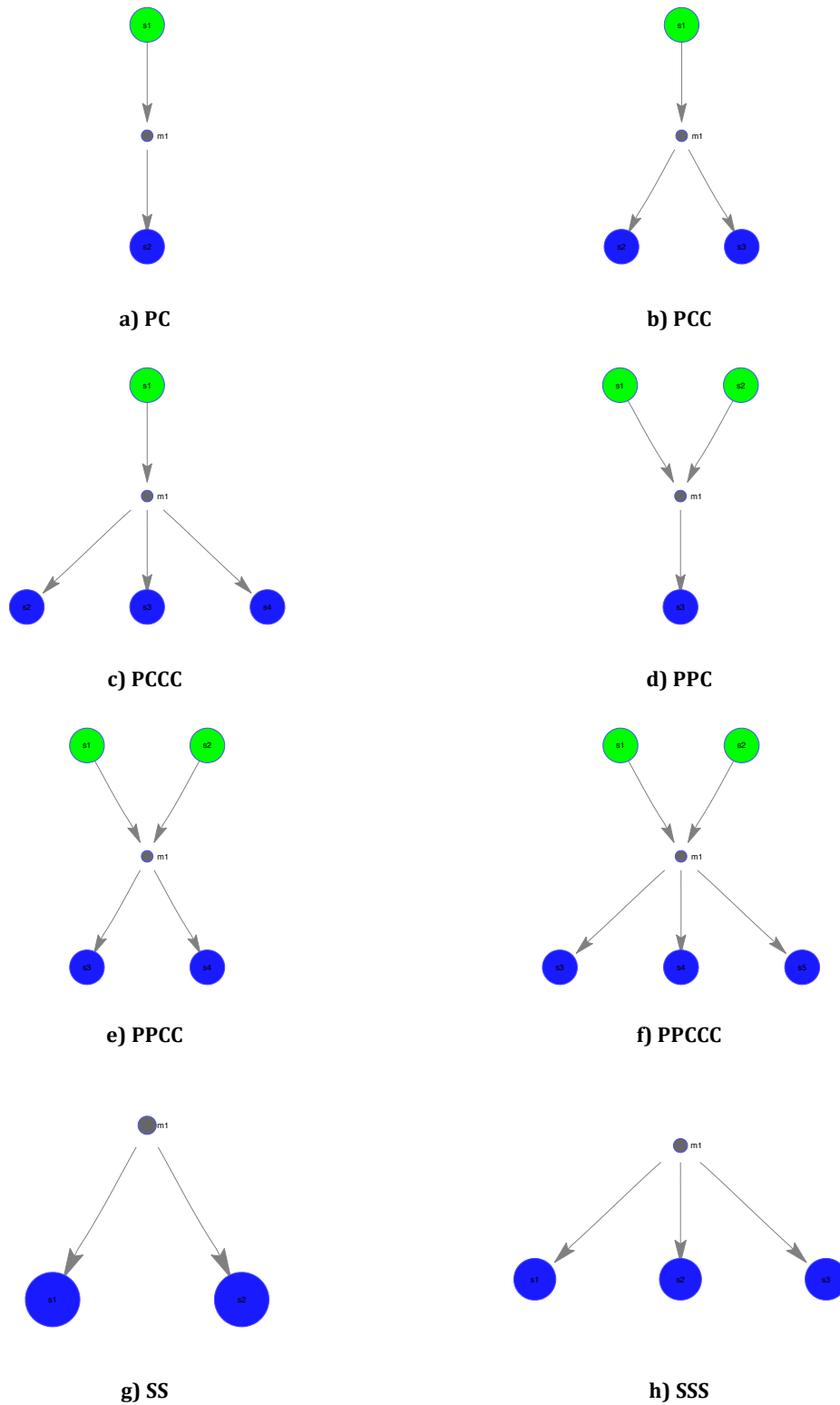


Figure 3: Graphical representations of the types of pedigrees used.

Since the size of the sample population is much smaller than size of the overall population, some allele values are rarely observed. In order to account for unobserved alleles, the frequency calculations were corrected by including a minimum allele frequency, which is calculated with the formula $5/2N$, where 5 is considered the smallest number of alleles counted, and N is the number of individuals in the sample population [2]. The minimum allele frequency is calculated for each locus and any allele frequency falling below that value is set to the minimum allele frequency. The allele frequencies are then normalized such that the allele frequencies at a given locus sum to one.

3.5.2 Generation of Independent Specimens

Large sets of specimens were pseudo-randomly generated by independently creating a DNA profile for each specimen. The algorithm is similar to random sequence generators that use a probability distribution to guide the selection of symbols in the sequence [13] and uses the normalized allele frequency distributions to determine the probability of independently selecting alleles for each locus. Several simplifying assumptions were made during this process: 1) the population is in Hardy-Weinberg equilibrium, which states that the alleles at a given locus are statistically independent [2-4,6], 2) the loci are in linkage equilibrium, which states that the alleles at different loci are statistically independent [3,6], and 3) all of the specimen profiles contain the same number of loci, each with a complete genotype. The first two assumptions allow the use of the product rule to calculate the estimated joint probabilities for the likelihood ratios and the third assumption allows ignoring phenomena such as allele dropout and degraded samples.

Specimens are generated locus by locus independently of each other. A genotype is generated at each locus by selecting two allele values according to the locus' normalized allele frequency distribution from NIST. The allele values may be

the same, in which case the genotype is homozygous, or different, in which case the genotype is heterozygous [2-4,6-8,14]. The genotypes of each locus are combined to form the specimen profile. The algorithm for pseudo-randomly generating the independent specimens is shown in Table 2.

3.5.3 Generation of Pedigree Specimens

Large sets of pedigrees were pseudo-randomly created by generating a pedigree trio consisting of two parent specimens and one offspring specimen and modifying the trio appropriately to form the target pedigree (the pedigrees of interest are discussed in detail in the next section). There were some simplifying assumptions made during this process: 1) the population mates randomly, and no inbreeding occurs, 2) mating specimens take only one partner so there are no half-relatives (such as half-siblings), and 3) no mutation occurs during the mating process.

Pedigrees of a specific type are generated by first creating a pedigree trio consisting of two parent specimens and one offspring specimen. The two parent specimens are generated independently using the algorithm for pseudo-randomly generating specimens in Table 2. Mating the two parent specimens produces the offspring profile that contains two alleles at each locus, where one of the alleles is IBD from the first parent and the other allele is IBD from the other parent. The trio is then modified by some combination of independently mating the parents to add offspring and removing one or both of the parents. The algorithm used to mate two specimens to form an offspring specimen is shown in Table 3. The algorithm used to generate and modify pedigree trios to create pedigrees of a specified type is shown in Table 4.

Table 2: Algorithm for pseudo-randomly generating specimens.

<i>Number of Specimens to Generate: n</i>
<i>Number of Loci in Each Profile: l, where $1 \leq l \leq 13$</i>
<i>Set of Allele Frequencies at Each Locus: $\{F_1^x, F_2^x, \dots, F_m^x\}$, where $1 \leq x \leq 13$ is the locus number, and $1, 2, \dots, m$ are arbitrary allele labels at locus x.</i>

- For each specimen $s = 1 : n$
 - For each locus $i = 1 : l$
 - i. Independently generate two pseudo-random numbers, $0 \leq r_1 < 1$ and $0 \leq r_2 < 1$
 - ii. Assign the first allele as allele p , where p is determined by:
 $G_{p-1} \leq r_1 < G_p$, where $G_p = F_p^i + G_{p-1}$, $G_0 = 0$, and locus i has m alleles
 - iii. Assign the second allele as allele p , where p is determined by:
 $G_{p-1} \leq r_2 < G_p$, where $G_p = F_p^i + G_{p-1}$, $G_0 = 0$, and locus i has m alleles
 - iv. Store the genotype for locus i in the profile for specimen s
 - Store the profile for specimen s in the collection

3.6 Study: Feasibility of Relationship Discovery Using STR Data

Given that the specimen generation process provides the full STR DNA profile of every individual in a collection, it seems plausible that the STR profiles may be useful in determining whether or not two individuals are related.

Table 3: Algorithm for forming an offspring specimen by mating parent specimens.

Specimen Profile of the Father Specimen: S_F

Specimen Profile of the Mother Specimen: S_M

Number of Loci in Each Profile: l , where $1 \leq l \leq 13$

Set of Allele Frequencies at Each Locus: $\{F_1^x, F_2^x, \dots, F_m^x\}$, where $1 \leq x \leq 13$ is the locus number, and $1, 2, \dots, m$ are arbitrary allele labels at locus x .

- For each locus $i = 1 : l$
 1. Pseudo-randomly generate a number $0 \leq r < 1$
 2. Let the alleles at locus i for S_F be labeled AB
 3. Let the alleles at locus i for S_M be labeled CD
 4. If $0 \leq r < 0.25$ then let the offspring alleles be AC
 - a. Set offspring allele 1 as allele A from S_F
 - b. Set offspring allele 2 as allele C from S_M
 5. Else If $0.25 \leq r < 0.5$ then let the offspring alleles be AD
 - a. Set offspring allele 1 as allele A from S_F
 - b. Set offspring allele 2 as allele D from S_M
 6. Else If $0.5 \leq r < 0.75$ then let the offspring alleles be BC
 - a. Set offspring allele 1 as allele B from S_F
 - b. Set offspring allele 2 as allele C from S_M
 7. Else If $0.75 \leq r < 1.0$ then let the offspring alleles be BD
 - a. Set offspring allele 1 as allele B from S_F
 - b. Set offspring allele 2 as allele D from S_M
 8. Store the genotype for locus i in the profile for the offspring specimen
 - Store the profile for the offspring specimen
-

Table 4: Algorithm for generating pedigrees of a specified type.

Number of Pedigrees to Generate: n

Type of Pedigrees to Generate: $t \in \{0, \dots, 8\}$

Number of Loci in Each Profile: l , where $1 \leq l \leq 13$

Set of Allele Frequencies at Each Locus: $\{F_1^x, F_2^x, \dots, F_m^x\}$, where $1 \leq x \leq 13$ is the locus number, and $1, 2, \dots, m$ are arbitrary allele labels at locus x .

- For each pedigree $i = 1 : n$
 1. Generate a pedigree trio
 - a. Independently generate two parent specimen profiles containing l loci using the algorithm in Table 2.
 - b. Generate an offspring specimen profile from the parent profiles using the mating algorithm in Table 3
 2. Modify the pedigree tree into type specified by t
 - a. If $t = 0$ then randomly select with equal probability a pedigree type by setting t to an integer in the range $t \in [1, 2, \dots, 8]$
 - b. If $t = 1$ then modify the pedigree to form a PC pedigree
 - i. Randomly select with equal probability one of the parent specimens and remove it from the pedigree
 - c. Else if $t = 2$ then modify the pedigree to form a PCC pedigree
 - i. Generate an additional offspring specimen profile from the parent profiles using the mating algorithm
 - ii. Randomly select with equal probability one of the parent specimens and remove it from the pedigree
 - d. Else if $t = 3$ then modify the pedigree to form a PCCC pedigree
 - i. Generate two additional offspring specimen profiles from the parent profiles using the mating algorithm
 - ii. Randomly select with equal probability one of the parent specimens and remove it from the pedigree
 - e. Else if $t = 4$ then modify the pedigree to form a PPC pedigree
-

-
- i. Since the trio is a PPC pedigree, there is no modification
 - f. Else if $t = 5$ then modify the pedigree to form a PPCC pedigree
 - i. Generate an additional offspring specimen profile from the parent profiles using the mating algorithm
 - g. Else if $t = 6$ then modify the pedigree to form a PCCC pedigree
 - i. Independently generate two additional offspring specimen profiles from the parent profiles using the mating algorithm
 - h. Else if $t = 7$ then modify the pedigree to form an SS pedigree
 - i. Generate an additional offspring specimen profile from the parent profiles using the mating algorithm
 - ii. Remove both of the parent specimens from the pedigree
 - i. Else if $t = 8$ then modify the pedigree to form an SSS pedigree
 - i. Independently generate two additional offspring specimen profiles from the parent profiles using the mating algorithm
 - ii. Remove both of the parent specimens from the pedigree
3. Store the profiles for the pedigree specimens in the collection
-

3.6.1 Objective

The purpose of this study was to determine the feasibility of using STR profiles to discover specimen relationships that may exist between specimens in a collection.

3.6.2 Formulation

When two individuals are related by a parent-child relationship, the individuals will share one allele at each locus. In terms of link analysis, this means that for each locus X , individual A will have a link to allele Y at locus X and individual B

will also have a link to allele *Y* at locus *X*. Thus, if the two individuals share an allele at a given locus there will be a path between them that goes through the shared allele. This idea is the basis for this feasibility study as it is expected that there would be no paths connecting two completely unrelated individuals (who share no alleles at any locus) and that the individuals would be separated in the graph space. Two potentially related individuals would be connected by a path and would be closer to each other in the graph space.

3.6.3 Implementation

The input to LDT was a small set of collections containing between one and thirty specimens. Each collection was generated using the pedigree generation algorithm. Each links in the data was recorded as an edge between a specimen and a coded allele value. The allele values were coded in such a way as to allow LDT to differentiate between allele values at different loci.

3.6.4 Interpretation of Results

The first collection tested consisted only of a single specimen with STR data at each of the thirteen core loci. Figure 4 shows a graph of the specimen linked to the associated STR alleles. Note that the specimen and profile form a star shape in which the specimen identifier is at the center of the star and the allele values are on the perimeter. The second collection tested consisted of a pedigree trio containing two parents and one offspring. Figure 5 shows two views of the pedigree. The first view (left) is from a distance and shows a small cluster of three specimens in the center. The shared alleles can be observed by noting that some of the perimeter vertices (the allele values) have a degree of more than one. The second view (right) shows a close-up of the pedigree cluster, which appears as a group of three stars. As expected, the specimens in the trio are close to each other and paths through shared alleles can be

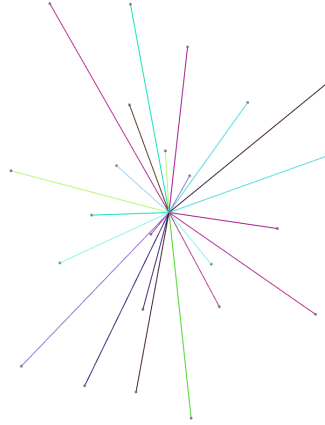


Figure 4: Graph of one specimen and associated STR profile.

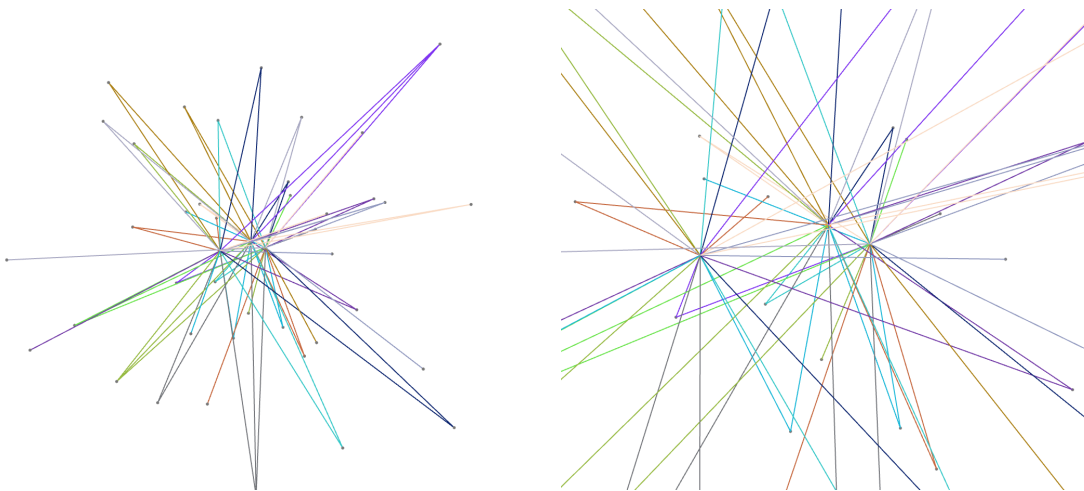


Figure 5: Graph of PPC trio from a distance (left) and close up of cluster of related specimens (right).

discovered between specimens. The third collection tested consisted of ten pedigree trios (PPC type). Figure 6 shows two views of the graph. The first view (left) is from a distance and shows the entire graph. The second view (right) shows a close up of the interior of the graph. It is easy to see that it has become very difficult to observe a potential relationship between any pair of individuals in the collection.

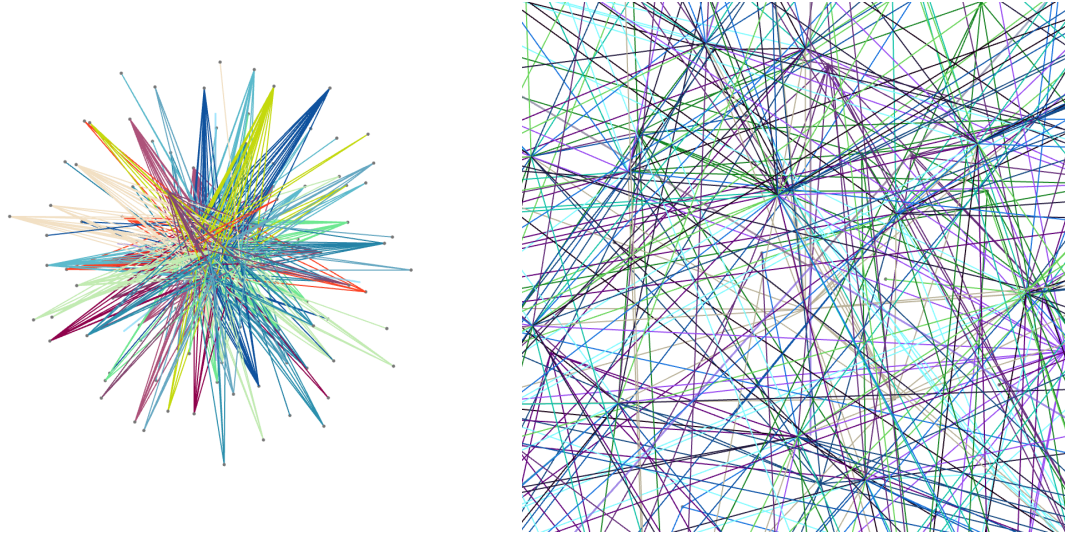


Figure 6: Graph of 10 PPC trios from a distance (left) and close up of potentially related specimens (right).

There are several problems with the use of STR profiles to discover relationships between individuals. 1) The representation is not scalable to hundreds or thousands of specimens because the increase in the number of edges begins to produce a spaghetti-like graph structure that is very difficult to interpret (Figure 6). 2) It is not immediately apparent as to the type of relationship being suggested by the graph (or even *if* a relationship is being suggested). For example, a parent and a child share one allele IBD at each locus, but siblings may also by chance share one allele at each locus. 3) There are many edges in the graph that are meaningless in terms of a given biological relationship. For example, consider a true parent-child relationship between two individuals. The two will share an allele IBD at each locus and so a path will be available from the parent to the child going through the shared allele node. However, the non-shared alleles in each profile will still have representative edges in the graph because each specimen node is connected to each of its 26 allele value nodes.

Visualizing hundreds or thousands of pedigrees with this method would be nearly impossible; therefore, this method of representing the specimen collections was abandoned.

3.7 Study: Feasibility of Relationship Discovery Using Likelihood Ratios

Given that the specimen generator provides the full STR DNA profile of every individual in a collection and that likelihood ratios provide a measure of relatedness between individuals under the stated relationship, it seems plausible that the likelihood ratios are useful as links between nodes representing STR specimens. This is one of the approaches disclosed in [15].

3.7.1 Objective

The purpose of this study was to determine the feasibility of using likelihood ratios as weights for edges in Link Discovery Tool, where the vertices represent a specimen profile containing thirteen loci and an edge between two vertices has a weight value that represents the value of the likelihood ratio and a label that represents the relationship measured by the ratio (for example, a parent-child relationship or a full-sibling relationship).

3.7.2 Formulation

The use of STR profiles in LDT to discover relationships between individuals was shown to be a poor choice with several inherent problems. Likelihood ratios have the potential to alleviate those concerns because: 1) a likelihood ratio representation is more scalable since there is at most one edge between two specimens for each type of relationship under consideration, 2) the type of

relationship being proposed is immediately apparent in the graph by simply labeling the LR edges representing each type of relationship, and 3) there are no meaningless edges in the graph in terms of a biological relationship since the presence of an edge signifies the potential for a relationship to exist. The use of LR values as weights on edges is a natural choice for LDT since we expect LDT to pull linked vertices together and push un-linked vertices apart. The addition of a weight representing the strength of the proposed relationship will tend to pull more strongly related vertices closer together than weakly related vertices.

3.7.3 Implementation

The input to LDT was a small set of collections containing between one and one hundred pedigree trios (PPC type). Each vertex in the collection represents a specimen and each edge in the collection represents a potential parent-child relationship. The weight of an edge represents the strength of the potential relationship and comes from the value of the corresponding parent-child likelihood ratio. The likelihood ratios were calculated for each pair of specimens in the collection.

3.7.4 Interpretation of Results

The first test was to establish whether or not the value of a likelihood ratio was viable as a link weight in LDT. The value of a likelihood ratio can range from zero (not related) to the tens (weak potential relationship) to thousands and above (strong potential relationship). This variability in link weight produced a very unstable graph structure in LDT in which the forces acting on a vertex would grow too large and cause the vertex to move too far away to be viewed, or the graph itself would oscillate. Figure 7 show a graph of a collection containing one hundred PPC pedigrees. The LR link weights vary considerably in value and cause the graph to

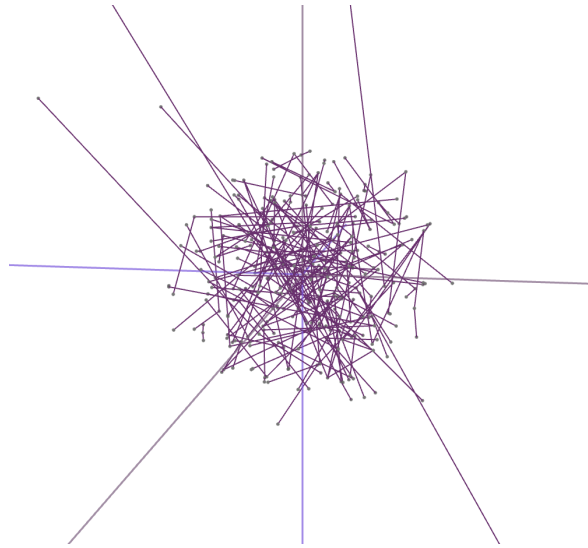


Figure 7: Graph of 100 PPC trios using LR link weights.

oscillate, forming the unappealing structure shown when the physics are frozen. To alleviate this problem the log-likelihood ratios (Log-LR) were used for the link weights and were computed by simply calculating the base-10 logarithms of the likelihood ratios. The use of Log-LR link weights produced a much more stable graph structure, as shown in Figure 8, which does not need to be frozen and clearly shows the pedigree clusters.

The use of Log-LR link weights provides a much more aesthetically appealing graph that also happens to be much more easily interpreted. The graphs shown in Figure 9 illustrate that the Log-LR link weights are also much more scalable in terms of the number of specimens contained in a collection. The collections used to produce the graphs in the figure consisted of PPC pedigrees. The first graph shows a collection containing two trios, the second shows a collection containing ten trios, the third shows a collection containing one hundred trios, and the fourth shows a collection containing two hundred trios. The long edges are weak parent-child relationships (the strong parent-child relationships have pulled the vertices together to form small

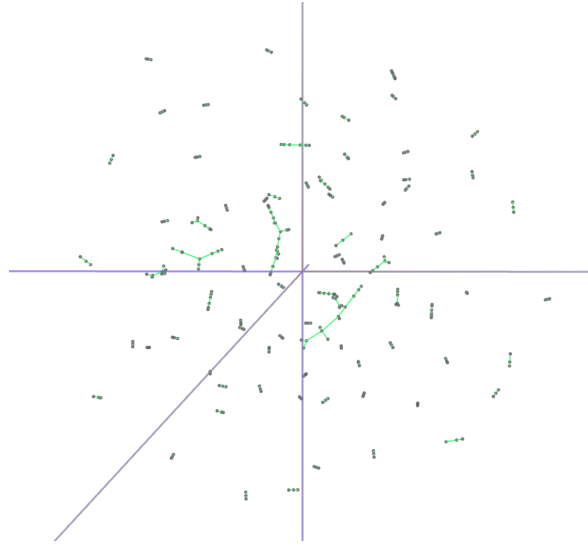


Figure 8: Graph of 100 PPC trios using Log-LR link weights.

clusters). Note that as the size of the collection increases, the number of potential edges in the collection increases quadratically [11]. However, since the Log-LR edges represent only viable potential relationships (and are otherwise not present), the complexity of the graph, in terms of the number of edges drawn, does not seem to increase at the same rate, which results in aesthetically pleasing graphs that remain relatively easy to interpret.

One issue that arises from the use of likelihood ratios as edge weights is that likelihood ratios indicate probability, not certainty, and can be present in the graph even when there is no true biological relationship between the specimens connected by the edge. This is especially true for likelihood ratios of full-sibling relationships, which allow for zero, one, or two IBD alleles at any locus, resulting in the potential for a sibling edge between any two specimens in the graph. Since most of these edges will represent false relationships, the use of a threshold value for accepting an edge into the graph becomes important. For example, Buckleton, et al. have shown that a

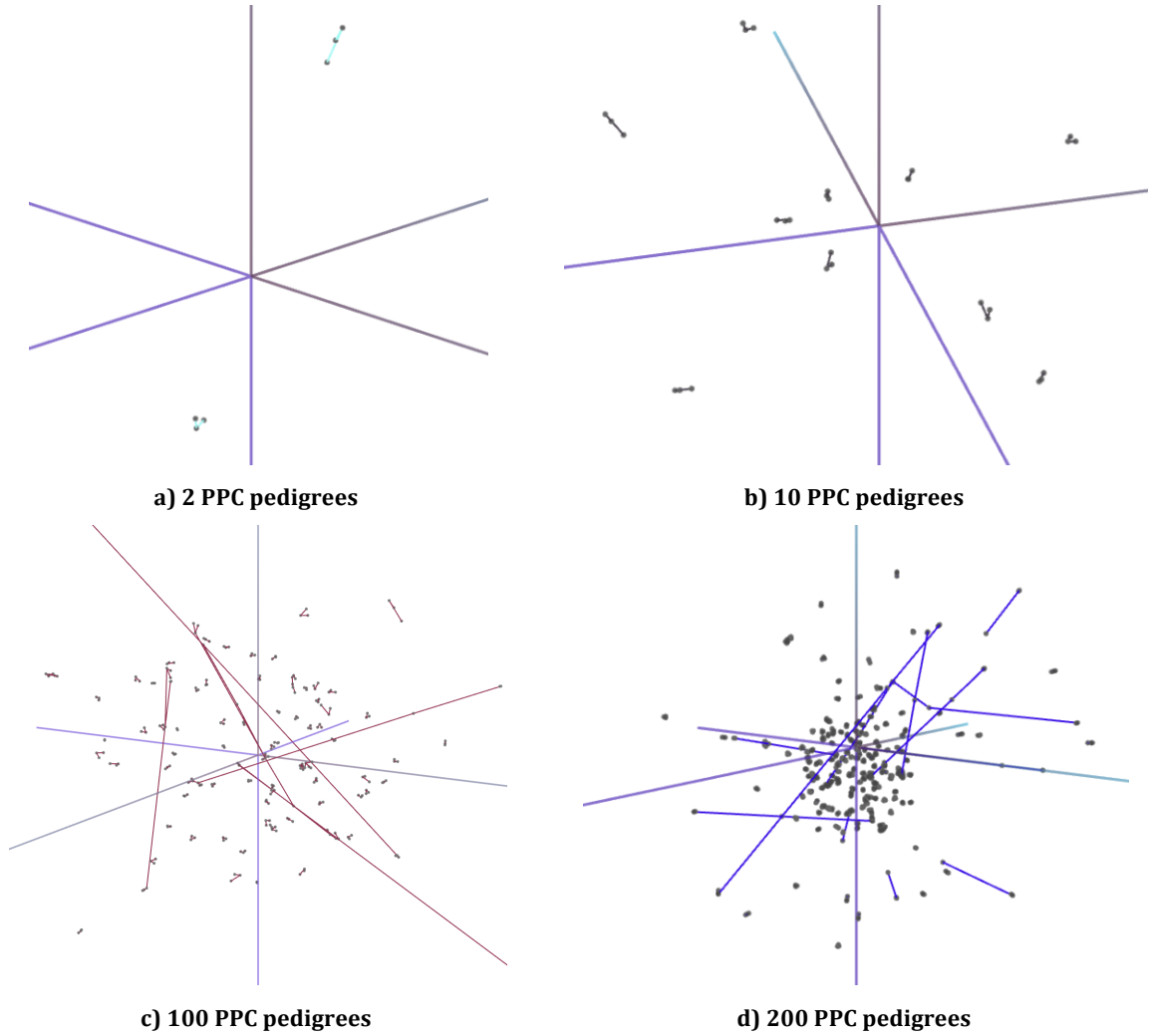


Figure 9: Graphs of pedigree trio collections of various sizes.

full-sibling log-likelihood ratio less than zero tends to indicate that the specimens are not related [6]. Parent-child relationships are similarly indicated by a log-likelihood ratio greater than zero (a log-likelihood ratio less than zero indicates that the case of observing the two profiles at random is more likely than the case in which they are related by the parent-child relationship). Therefore, an initial edge weight threshold value of zero will be used to decide whether or not to accept an edge into the graph.

The influence of the selection of threshold value on the performance of the algorithm is explored in the next chapter.

3.8 Pedigree Discovery Algorithm

3.8.1 Overview

The foundation of the approach to pedigree discovery explored and developed in this work lies in graph clustering. Link Discovery Tool provides a reliable physics-based clustering functionality, so the extension of LDT to allow for pedigree discovery requires a method of recognizing and analyzing the resulting spatial clusters. Based on the results of the feasibility studies discussed earlier, the data set should consist of vertices representing DNA specimens and weighted edges representing the strength of hypothesized parent-child and full-sibling relationships; the pedigree generator discussed in Section 3.5 provides data collections conforming to these constraints using log-likelihood ratios as edge weights.

Link Discovery Tool clusters the specimens in the data set using a physics-based algorithm (Section 3.3). This provides a solid first step to discovering pedigrees, however, the clusters provided by LDT are dependent upon the initial (random) graph configuration so results may vary between runs. However, several runs suggest that the variance in the clustering manifests as translations and rotations of the clusters; membership of the resulting clusters seems to be quite stable. Analyzing the clusters manually is very tedious and time consuming and may be difficult to do in the presence of large quantities of false full-sibling edges. These edges often form clusters containing hundreds of specimens and thousands of edges. Clusters of this magnitude cannot possibly be one of the eight types of pedigrees studied in this work, so the pedigree discovery algorithm presented here seeks to

reduce the size of large clusters by incrementally removing (“snipping”) edges. The process results in several smaller clusters that can then be analyzed, classified, and reported.

The following sections present the snip algorithm, which incrementally removes the weakest edge from a cluster, the analysis algorithm, which processes snipped clusters and classifies them, and the discovery algorithm, which combines snipping and analysis and outlines the process of pedigree discovery using Link Discovery Tool.

3.8.2 Snip Algorithm

The snip algorithm is based on simple observations of how vertex clusters form and appear in LDT. The first observation is that pedigrees contain related individuals, and with the exception of two parents (if both are present in the pedigree) every member of the pedigree has a valid biological relationship that is expressed as a weighted edge in the graph. Therefore, the pedigree will appear as a near-clique in the graph and the strength of the relationships among pedigree members will tend to pull the members closer to each other. The second observation is that an individual that appears to be related to a pedigree member will be attracted to the hypothetically related member, but repulsive towards the remaining pedigree members. The combined repulsion will tend to push the individual away from the pedigree. In this example, an observation that considers only edge weights would have difficulty distinguishing between a false relationship that has a high likelihood and a true relationship, but the clustering provided by LDT should expose such an edge as a potentially false relationship.

The edges of the greatest length in the graph correspond to relationships of the weakest strength. The snip algorithm incrementally removes these edges by first

separating the graph into components, which are labeled as candidate pedigree clusters. The components are considered in turn and the size of each determines whether or not it requires snipping. The largest pedigree under consideration is the PPCCC pedigree, which contains five members (two parent specimens and three offspring specimens). Any candidate cluster with more than five members cannot match the largest pedigree (or any of the smaller pedigrees) and requires snipping. The process of snipping is fairly simple: the length of each edge in the component is computed and the edge with the greatest length is removed from the component, which may split the component into two smaller components if the removed edge was a cut edge. Once every cluster has been examined and snipped if necessary, the graph is again separated into components, which are labeled as candidate clusters. This process is repeated until every graph component contains five or fewer vertices. The resulting candidate clusters are now considered viable candidate clusters and are ready to be analyzed and classified. The snip algorithm is shown in Table 5 and assumes that physics have been sufficiently applied.

3.8.3 Analysis Algorithm

The analysis algorithm is a decision model that analyzes a cluster for specific characteristics that can be used to classify the cluster as a pedigree type. The characteristics used include: the size of the cluster (the number of vertices in the cluster), the number of parent-child (PC) edges in the cluster, the number of full-sibling (Sib) edges in the cluster, the PC-degree of a vertex, and the Sib-degree of a vertex. For example, consider a simple two-member parent-child pedigree. The members of the pedigree share a PC edge describing the parent-child relationship. However, since the parent-child relationship requires one allele IBD at each locus, this relationship will be accompanied by a Sib edge describing a potential sibling relationship (full-siblings can share zero, one, or two IBD alleles at each locus). Therefore, a cluster representing a parent-child pedigree will contain one PC edge

Table 5: Snip algorithm for forming viable candidate clusters.

Graph Separated into N Components: $G = C_1 \cup C_2 \cup \dots \cup C_N$ where $C_i \cap C_j = \emptyset$ for $i \neq j$

- Set *finished* = *false*
- Loop until *finished*==*true*
 1. Set *finished* = *true*
 2. For each component C_i in G :
 - a. If component size ≥ 6 , then
 - i. Set *finished* = *false*
 - ii. Determine the longest edge in the component and remove (snip) it from the component
 - b. Else, ignore the component
 3. Separate the graph into components $G = C_1 \cup C_2 \cup \dots \cup C_M$ where $C_i \cap C_j = \emptyset$ for $i \neq j$ and $M > N$ if a cut edge was snipped in one of the previous clusters
- Resulting components $C_1, C_2, \dots, C_M$ are viable candidate clusters

and one Sib edge. Compare this to a two-member sibling pedigree (SS) whose members share a Sib edge, but are not likely to also share a PC edge. The decision between these two pedigrees would simply check for the presence of a sibling edge (its absence indicates two un-related individuals) and then a parent-child edge. A pedigree containing both edge types would be declared a parent-child pedigree, while a pedigree missing the PC edge would be declared an SS pedigree.

The decision process is often complicated by full-sibling edges. Occasionally, two individuals that are true full-siblings will share an allele IBD at each locus, adding a PC edge to the Sib edge describing the true relationship. This situation can make it

difficult to correctly classify a pedigree. For example, consider a PCC pedigree (one parent and two children). Ideally there would be two PC edges, one between the parent and each child, and three Sib edges, one between the parent and each child, and one between the two children. However, if the sibling relationship includes a PC edge then the cluster contains three PC edges, which cannot happen (since this would require one child to be the parent of the other, which is not possible since inbreeding is neither allowed nor represented in the data set). These situations cause ambiguity in the clusters, making it difficult to determine the true pedigree type. For example, consider a PPC pedigree, which should contain two PC edges and two Sib edges, one each between each parent and the child. Since the child is complete on Sib edges (it cannot have another Sib edge), the addition of a false Sib edge must go between the two parents. The cluster then contains two PC edges and three Sib edges, which looks exactly like the PCC pedigree mentioned above. In this situation the cluster would be incorrectly classified as a PCC pedigree since one would not expect two unrelated parent specimens to share a Sib edge, but one would expect two siblings to share a Sib edge.

Despite this ambiguity, there are some cases in which the inclusion of a PC edge in a true sibling relationship can be detected, allowing the pedigree to still be correctly classified. For example, consider a PPCCC pedigree (two parents and three children). This type of pedigree would ideally contain six PC edges, one between each parent-child pair, and nine Sib edges, one between each parent-child pair, and one between each sibling pair. If the addition of a false PC edge occurred between a pair of siblings, then the cluster would contain seven PC edges, the parents would have a PC degree of three, and one child one have a PC degree of three. If the false PC edge were to occur between the parents (which would result in an invalid pedigree due to inbreeding) then the parents would have a PC degree of four and the children would have a PC degree of two.

The analysis algorithm classifies a cluster as a particular type of pedigree by measuring the characteristics mentioned above and selecting a type using the pedigree characteristics shown in Table 6. The algorithm is shown in Table 7, and assumes that the graph components have been properly constructed (e.g., each vertex in the component has at least one edge, otherwise the vertex would be in a component by itself). A discussion of the decision rules and pedigree characteristics used for matching clusters to pedigree types is provided in Appendix B.

3.8.4 Discovery Algorithm

The pedigree discovery algorithm combines the snip algorithm and the analysis algorithm to reduce large components into smaller components that represent classifiable candidate clusters. Large components may contain one hundred or more specimens when the data set contains large numbers of specimens. Since the pedigrees generated by the pedigree generator contain at most five members, specimens in large clusters cannot all be related. The snip algorithm makes use of the lengths of edges within the cluster to incrementally remove the longest edge, corresponding to the weakest potential relationship. Eventually a cut edge is snipped, dividing the large cluster into two smaller clusters. The members of one smaller cluster are more likely to be related to other members within the cluster than to members of the other cluster. This process is repeated until the cluster size has become small enough to be classified.

The analysis algorithm classifies the candidate clusters by measuring certain characteristics of the component and matching them to characteristics of various types of pedigrees. If the candidate cluster cannot be classified, due to confusion caused by additional or missing edges, or due to confusion caused by additional or missing vertices, it is marked as an unknown type. Clusters of unknown type are input to the snip algorithm to remove the weakest (longest) edge. The cluster

Table 6: Pedigree characteristic for decision rules in the analysis algorithm.

Pedigree	
Type	Relevant Pedigree Characteristics
Individual (I)	Size = 1
PC	Size = 2, Number of PC edges = 1, Number of Sib edges = 1
PCC	Size = 3, Number of PC edges = 2, Number of Sib edges = 3, parent-member PC degree = 2, child-member PC degree = 1, parent-member Sib degree = 2, child-member Sib degree = 2
PPCC	Size = 4, Number of PC edges = 3, Number of Sib edges = 6, parent-member PC degree = 3, child-member PC degree = 1, parent-member Sib degree = 3, child-member Sib degree = 3
PPC	Size = 3, Number of PC edges = 2, Number of Sib edges = 2, parent-member PC degree = 1, child-member PC degree = 2 parent-member Sib degree = 1, child-member Sib degree = 2
PPCC	Size = 4, Number of PC edges = 4, Number of Sib edges = 5, parent-member PC degree = 2, child-member PC degree = 2, parent-member Sib degree = 2, child-member Sib degree = 3
PPCCC	Size = 5, Number of PC edges = 6, Number of Sib edges = 9, parent-member PC degree = 3, child-member PC degree = 2, parent-member Sib degree = 3, child-member Sib degree = 4
SS	Size = 2, Number of PC edges = 0, Number of Sib edges = 1
SSS	Size = 3, Number of PC edges = 0, Number of Sib edges = 3, sibling-member PC degree = 0, sibling-member Sib degree = 2

Table 7: Analysis algorithm for pedigree classification.

<i>Graph Separated into N Components: $G = C_1 \cup C_2 \cup \dots \cup C_N$ where $C_i \cap C_j = \emptyset$ for $i \neq j$</i>	
<i>Each C_i is a viable candidate cluster formed by the snip algorithm</i>	
•	For the specified component C_i , $i \in \{1, 2, \dots, N\}$
1.	Measure the <i>size</i> of the component, the number N_{PC} of PC edges, the number N_S of Sib edges
2.	If <i>size</i> == 1, then classify C_i as an individual
3.	Else if <i>size</i> == 2, then
a.	Classify C_i as either PC or SS type using a decision rule
b.	If C_i couldn't be classified with a decision rule, then classify it as unknown
4.	Else if <i>size</i> == 3, then
a.	Classify C_i as either PCC, PPC, or SSS type using a decision rule
b.	If C_i couldn't be classified with a decision rule, then classify it as unknown
5.	Else if <i>size</i> == 4, then
a.	Classify C_i as either PCCC or PPCC type using a decision rule
b.	If C_i couldn't be classified with a decision rule, then classify it as unknown
6.	Else if <i>size</i> == 5, then
a.	Classify C_i as PPCCC type using a decision rule
b.	If C_i couldn't be classified with a decision rule, then classify it as unknown
7.	Else
a.	C_i was not sufficiently snipped, so classify it as unknown

component is then checked to see if the removed edge was a cut edge, and if so the cluster is split into two smaller components. The resulting component is then re-analyzed. This process is repeated on each candidate cluster until it has been classified (which, at worst case results in individuals). The algorithm for pedigree discovery is shown in Table 8.

Table 8: Pedigree discovery algorithm.

<i>Graph G containing V vertices, E_{PC} weighted Log-LR PC edges, E_S weighted Log-LR Sib edges</i>
1. Enable every vertex and edge in the graph 2. Build a list L of components $C_1, C_2, \dots, C_N$, where $G = C_1 \cup C_2 \cup \dots \cup C_N$, $C_i \cap C_j = \emptyset$ for $i \neq j$, and sort the vertices V and edges E_{PC} and E_S into the appropriate components. The components are candidate clusters. 3. Execute the <i>Snip Algorithm</i> on the graph 4. Loop through each cluster $C_1, C_2, \dots, C_M$ resulting from 3. a. Loop while the cluster C_i is classified as <i>unknown</i> i. Execute the <i>Analysis Algorithm</i> on cluster C_i ii. If cluster C_i could not be classified, execute the <i>Snip Algorithm</i> on C_i 5. Report the classification of each pedigree cluster

Chapter 4: Results and Discussion

4.1 Experiment: Discovery of Individuals

4.1.1 Objective

The purpose of this experiment was to establish a base-line figure for the performance of the pedigree discovery algorithm on individual specimens and how individual specimens may introduce error into the classification results. The data sets used in this experiment consisted of collections of individual specimens and edges representing log-likelihood ratios for potential parent-child and full-sibling relationships.

4.1.2 Results

The test was executed using collections of specimens containing between 10 and 2000 individuals using an edge weight threshold value of zero. The graph in Figure 10 shows the accuracy of discovering individuals as the size of the data set increases. The discovery algorithm correctly recognizes individuals when the data set is small (and trivial), but the classification accuracy quickly drops to between forty-five and fifty percent with larger data sets. The reason for the reduction in accuracy is the increase in parent-child and full-sibling edges in larger data sets. For a collection of size N , there are $\frac{1}{2}N(N-1)$ potential edges for each type of relationship [11], and there is a non-zero fixed probability that an edge between unrelated individuals will match.

The parent-child relationship is the more differentiable relationship of the two and is much less likely to occur between two non-related individuals. For example, the collection of size 2,000 used in this experiment contained 465 of a possible 1,999,000 PC-LR edges. The full-sibling relationship is a much less differentiable since the individuals can share zero, one, or two alleles at a locus. Therefore, a full-sibling edge is much more likely to occur between two non-related individuals. The data set in the previous example contained 37,679 of a possible 1,999,000 Sib-LR edges. The number of full-sibling edges grows at a much higher rate than the number of parent-child edges, as shown in Figure 11. Figure 12 and Figure 13 show that the growth is quadratic for both PC-LR and Sib-LR edges (see Table 9 for the polynomial coefficients for each edge type), but confirms that the number of Sib-LR edges grows much more rapidly. Since the collections used in this experiment contained only individuals, every edge in the graph represents a false relationship. These false relationships are very difficult to differentiate from true relationships using only likelihood ratios and are the source of classification error.

4.2 Experiment: Discovery of Pedigrees

4.2.1 Objective

The purpose of this experiment was to establish a base-line figure for the performance of the pedigree discovery algorithm on various types of pedigrees. The data sets used in this experiment consisted of collections containing varying numbers of pedigrees of uniform type and edges representing log-likelihood ratios for parent-child and full-sibling relationships.

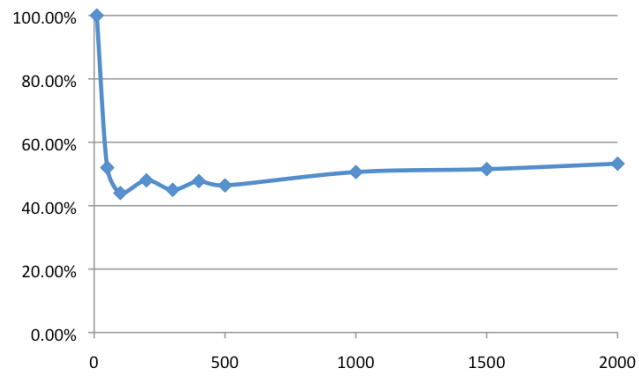


Figure 10: Discovery accuracy of individuals as collection size increases.

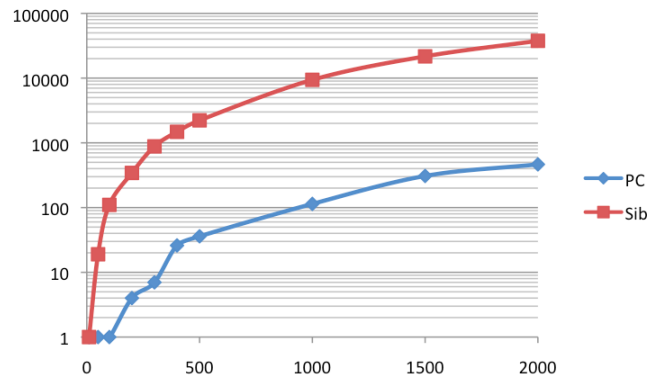


Figure 11: Log of number of PC-LR and Sib-LR edges as collection size increases.

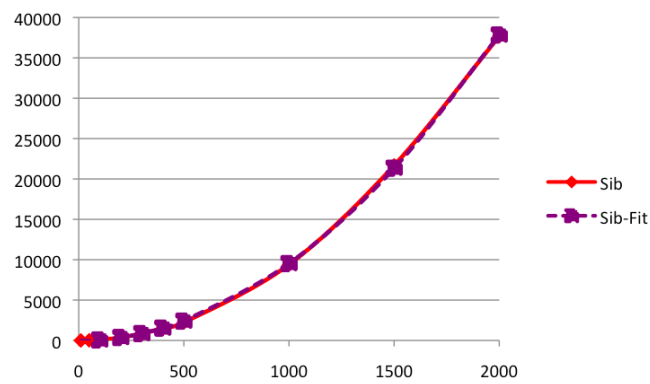


Figure 12: Quadratic polynomial regression of number of Sib-LR edges as collection size increases.

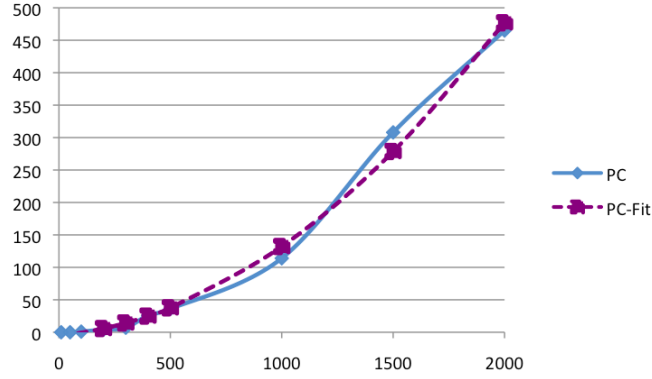


Figure 13: Quadratic polynomial regression of number of PC-LR edges as collection size increases.

Table 9: Coefficients of polynomial regressions for the number of PC-LR and Sib-LR edges.

#PC or Sib Edges = An^2+Bn+C			
where $n = \#$ of profiles in collection			
Edge Type	A	B	C
PC	0.0001	0.0336	-5.1621
Sib	0.0094	0.2089	-52.9398

4.2.2 Results

The test was executed using collections of pedigrees containing 10, 50, 100, 200, 300, 400, and 500 pedigrees of the same type using an edge weight threshold of zero. The results of the test are shown in Figure 14 through Figure 20, which show the hypothesized pedigree type (provided by the discovery algorithm) versus the true pedigree type. Ideally, the graphs would show tall spikes along the diagonal and no spikes in the individual-type column (I) since there are no individuals in the collection. The figures show that discovery accuracy is high for small pedigrees and slightly lower for larger pedigrees. As the size of the collection increases, the accuracy of discovery degrades quickly for the larger pedigrees, as indicated in Figure

21, which shows discovery accuracy versus collection size for each pedigree type. The cause of the degradation is due to the increased presence of Sib-LR edges and the choice of edge weight threshold value, which are both explored in the following experiments.

4.3 Experiment: Influence of Sibling Edges

4.3.1 Objective

The purpose of this experiment was to determine how much influence the number of Sib-LR edges has on the performance of the discovery algorithm. The data sets used in this experiment consisted of collections containing varying numbers of pedigrees of uniform type and edges representing log-likelihood ratios for parent-child and full-sibling relationships.

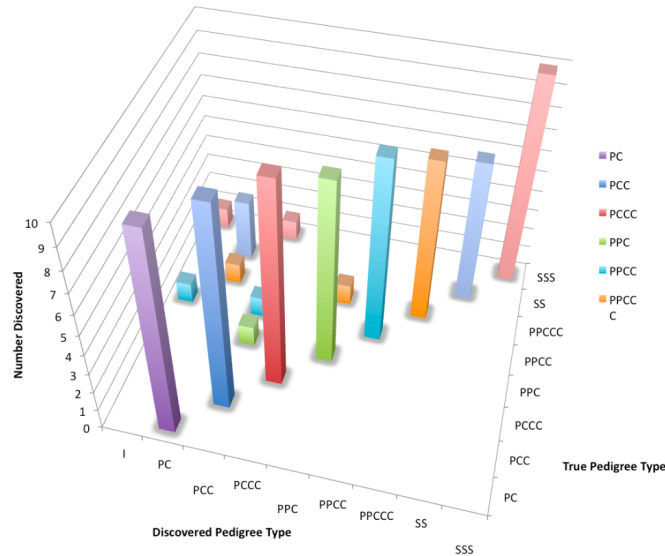


Figure 14: Results of discovery with collections of 10 pedigrees.

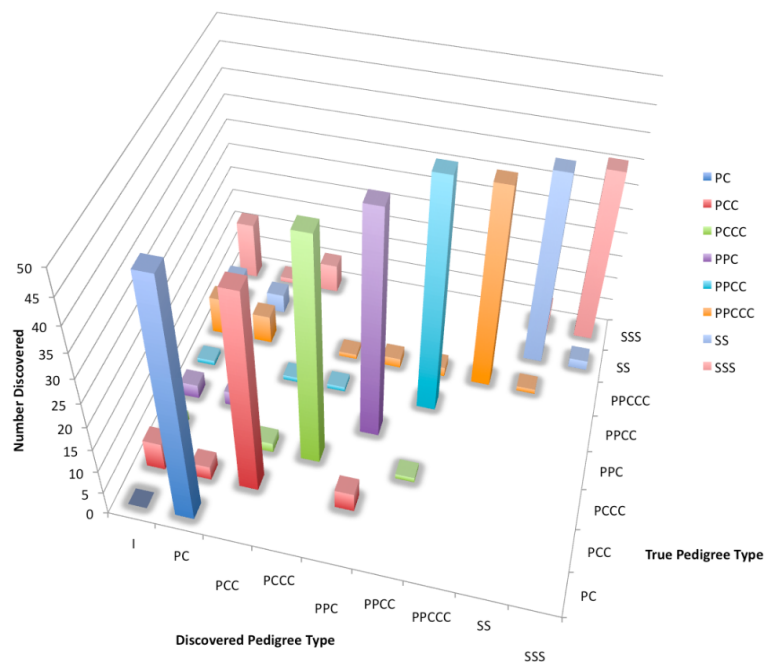


Figure 15: Results of discovery with collections of 50 pedigrees.

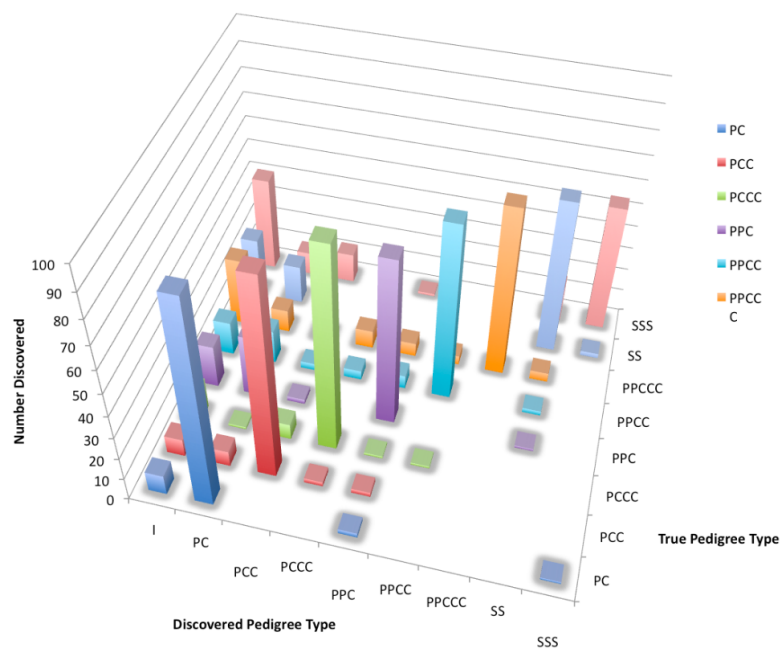


Figure 16: Results of discovery with collections of 100 pedigrees.

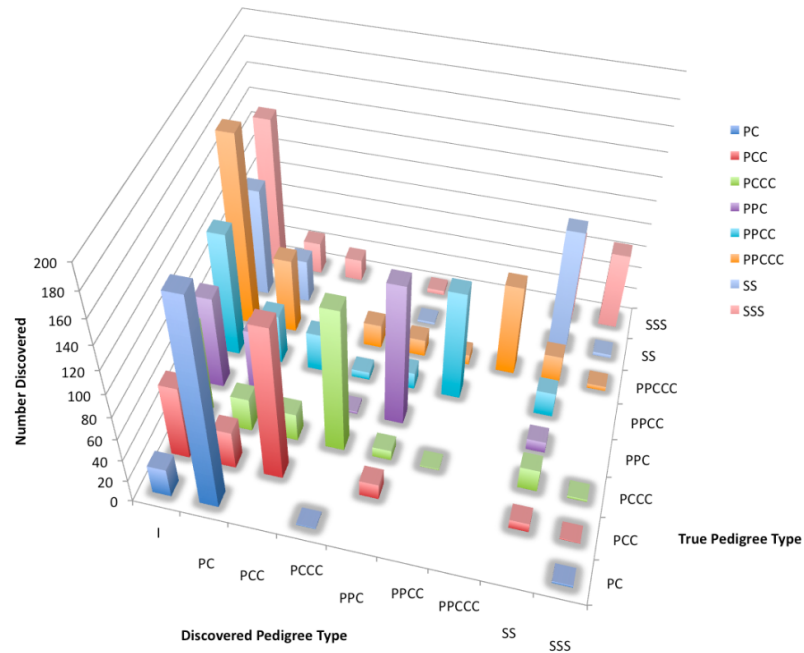


Figure 17: Results of discovery with collections of 200 pedigrees.

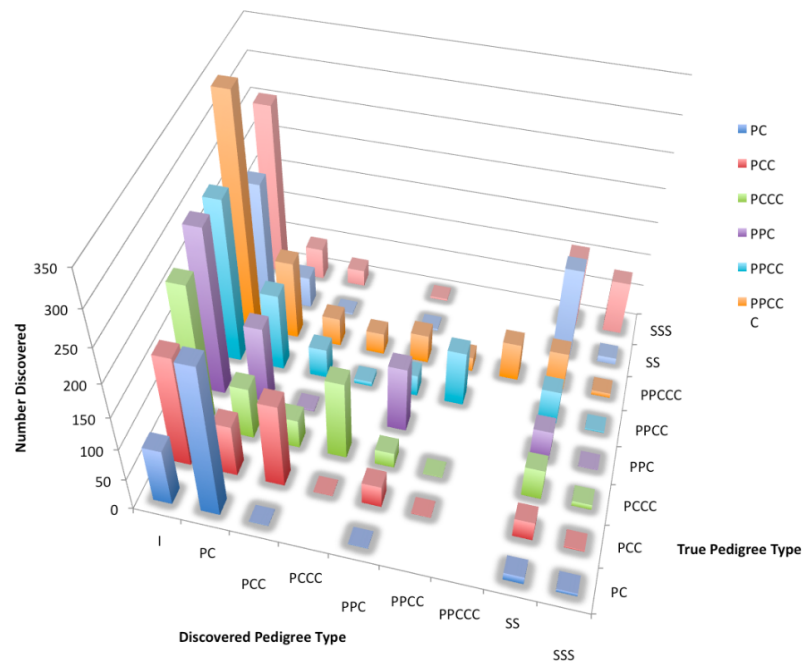


Figure 18: Results of discovery with collections of 300 pedigrees.

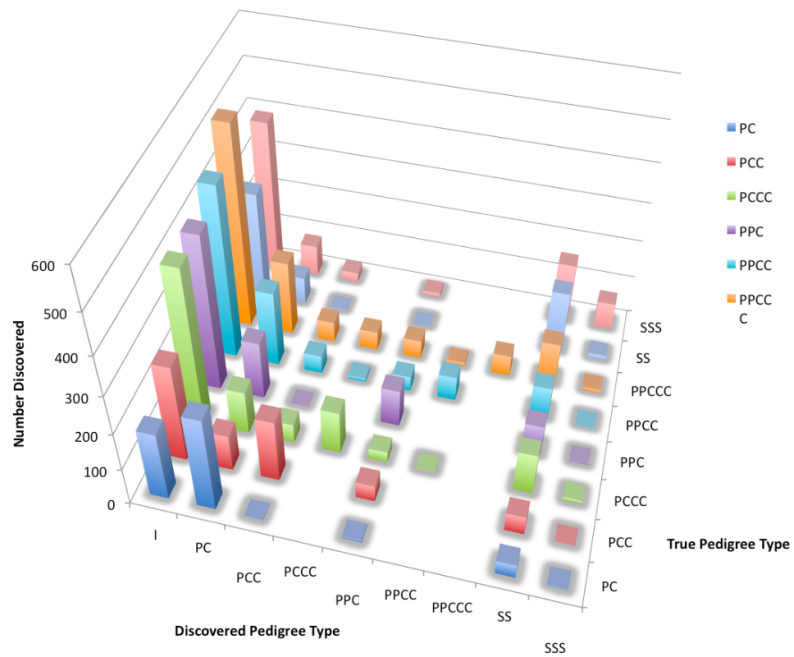


Figure 19: Results of discovery with collections of 400 pedigrees.

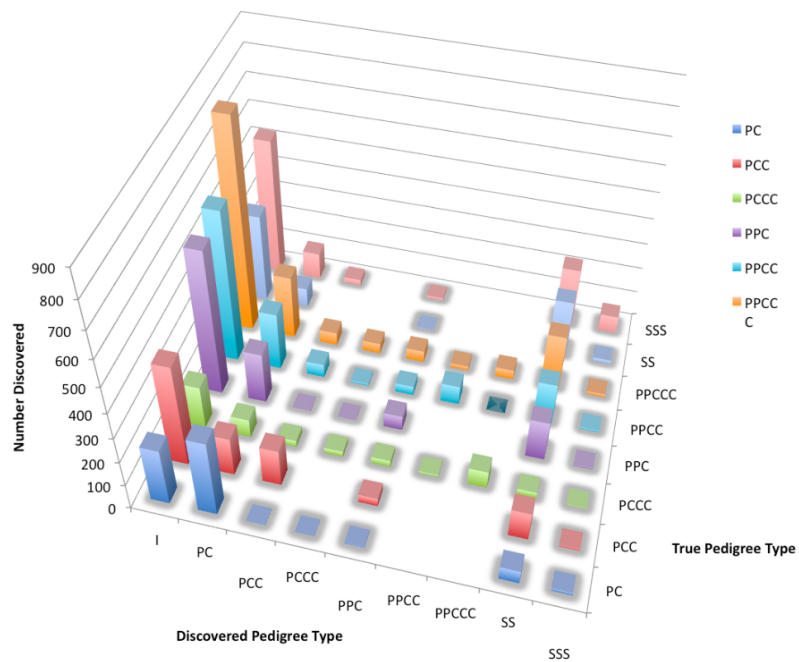


Figure 20: Results of discovery with collections of 500 pedigrees.

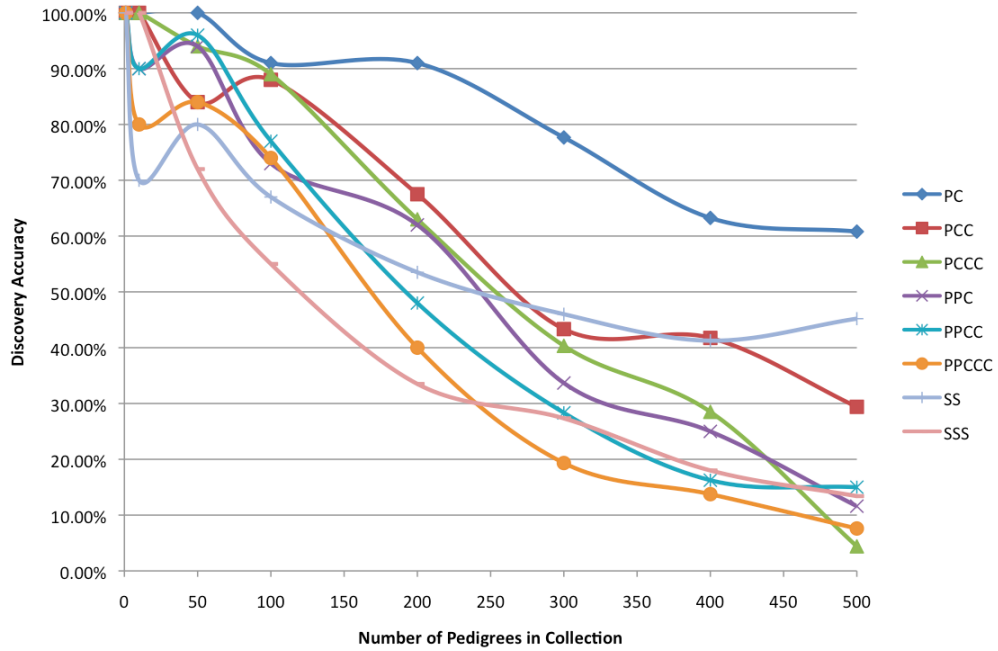
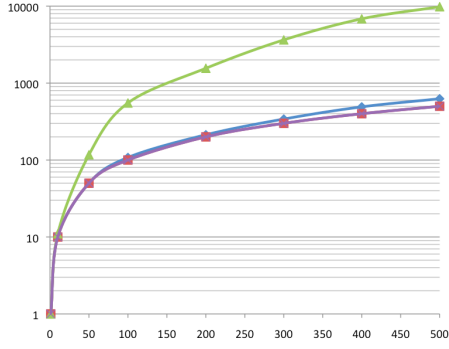


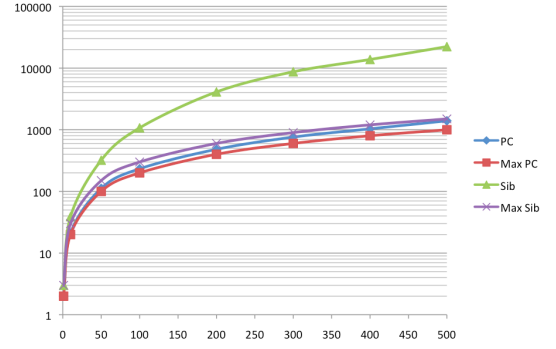
Figure 21: Accuracy of discovering true pedigree as pedigree collection size increases.

4.3.2 Results

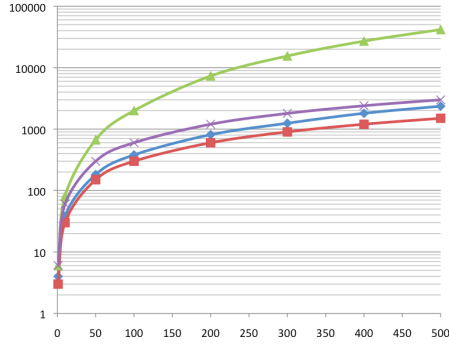
As shown by the results in the previous section, the ability to discover the true structures of pedigrees in a collection degrades rapidly as the collection size increases. The largest pedigree, the PPCCC pedigree, contains nine true Sib-LR edges (three that represent the full-sibling relationships and six that accompany the parent-child relationships). As the number of pedigrees p in a collection increases, the number of true parent-child and full-sibling edges increases linearly with p . However, the plots in Figure 22 show that while the numbers of both edge types grow quadratically, the number of Sib-LR edges in the graph grows at a much higher rate than the number of PC-LR edges. For larger data sets, this causes the number of full-sibling edges to vastly out-grow the number of parent-child edges, as can be seen in the plots of Figure 23 which show the composition of the graph in terms of the percentages of parent-child edges (blue) and full-sibling edges (red) as the collection



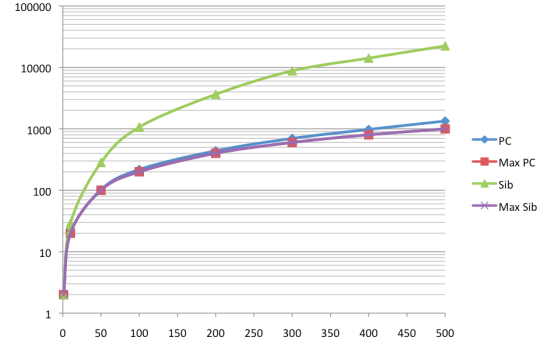
a) PC Type



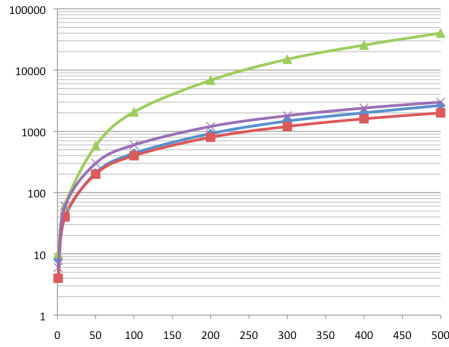
b) PCC Type



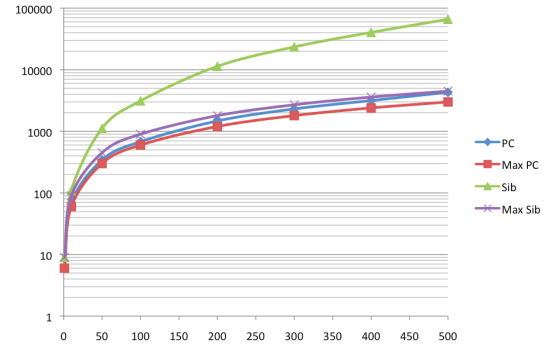
c) PCCC Type



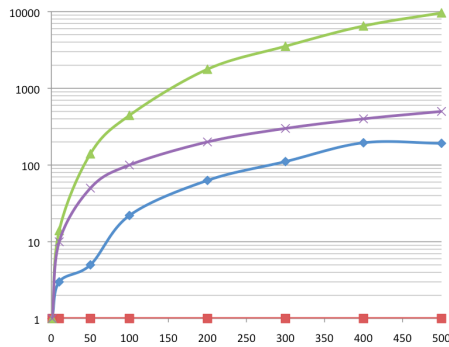
d) PPC Type



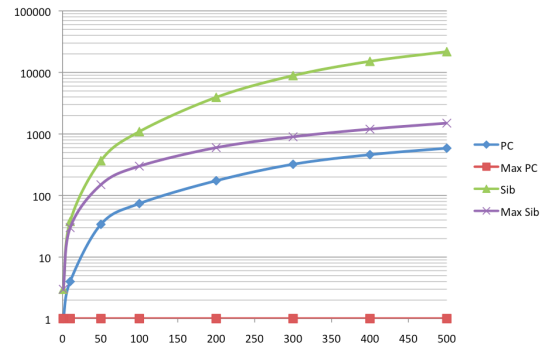
e) PPCC Type



f) PPCCC Type

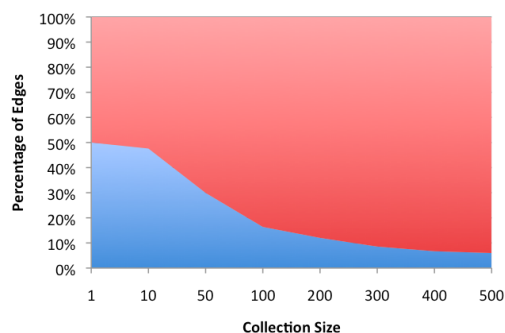


g) SS Type

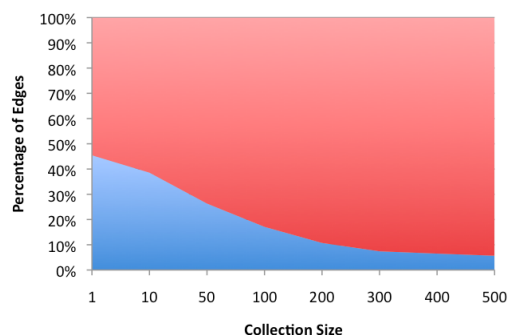


h) SSS Type

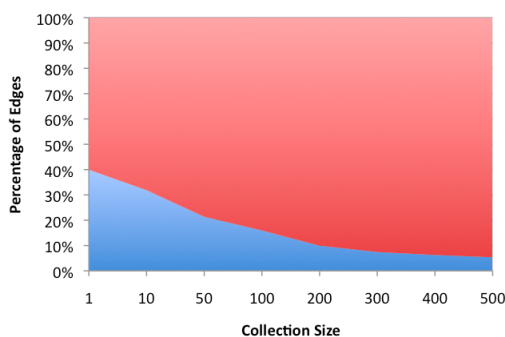
Figure 22: Number of PC and Sib edges in graph versus the maximum number of true PC and Sib edges.



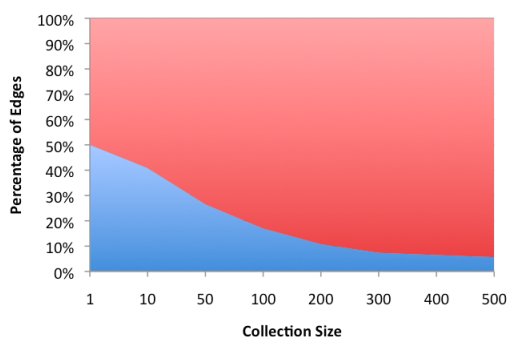
a) PC Type



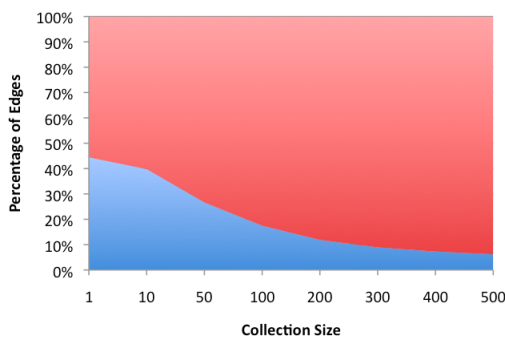
b) PCC Type



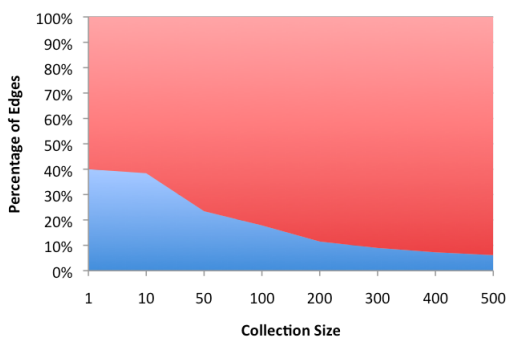
c) PCCC Type



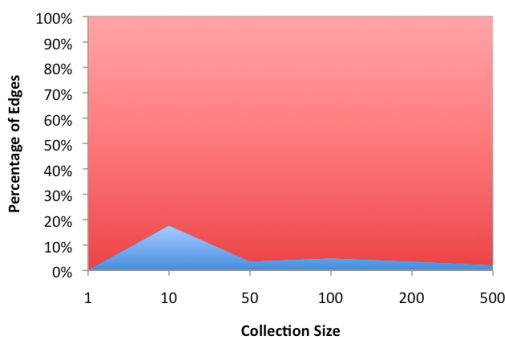
d) PPC Type



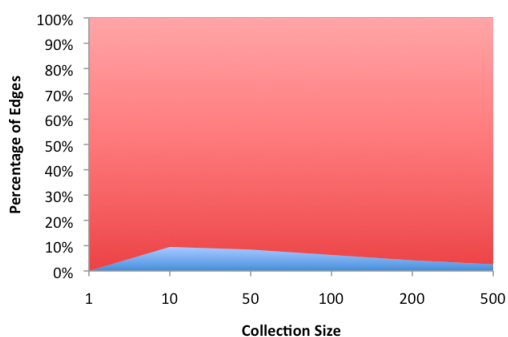
e) PPCC Type



f) PPCCC Type



g) SS Type



h) SSS Type

Figure 23: Graph composition as percentage of PC-LR and Sib-LR edges.

size increases. The majority of the full-sibling edges in a large collection will represent false sibling relationships and will distort the clusters in the graph, reducing the accuracy of the discovery algorithm. The experiments in the following sections test a method of reducing the number of false full-sibling edges in the graph.

4.4 Distribution of Likelihood Ratio Values

As indicated by the results of the previous experiments, the number of full-sibling edges in the graph degrades the performance of the discovery algorithm, especially with large collections. The use of a likelihood ratio threshold value, which rejects any potential edge associated with an LR value below the threshold, can improve the performance of the algorithm by removing edges representing false relationship (e.g., edges with a weak LR value).

In order to measure the performance of the likelihood ratio as a measure of the strength of a potential biological relationship and gain insight into the appropriate threshold value, likelihood ratio distribution curves were generated for the full-sibling relationship type. The curves were generated using two million pairs of unrelated individuals (hypothesis H_0) and two million pairs of individuals related as full-siblings (hypothesis H_1) and are shown in Figure 24. The curve on the left represents the distribution of full-sibling log-likelihood ratio values for individuals known to be unrelated. The curve on the right represents the distribution of full-sibling log-likelihood ratio values for individuals known to be related as full-siblings. A clear separation exists between the two curves, indicating that the value of the log-likelihood ratio is a good indicator of whether or not two individuals are related as full-siblings. For example, any pair of individuals with a full-sibling Log-LR value of -2 or less are almost certainly unrelated, which any pair of individuals with a full-sibling Log-LR value of 2 or higher are almost certainly related as full-siblings.

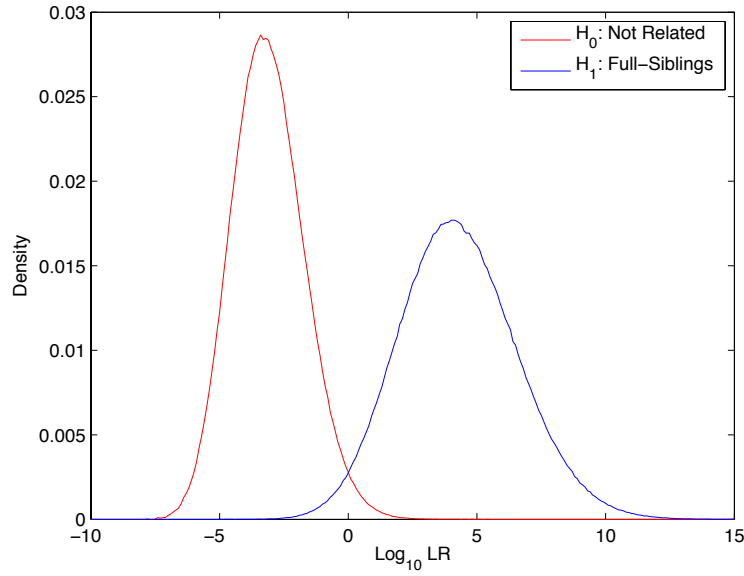


Figure 24: Distributions of the log-likelihood ratio for pairs of unrelated individuals and pairs of individuals related as full-siblings.

The region between -2 and 2 is where most of the error occurs, which manifests as false positive classifications (Type I error) and false negative classifications (Type II error). Figure 25 shows the frequency of each type of error as a function of the Log-LR threshold value and shows that the minimum frequency of error (the sum of Type I and II error) occurs when the Log-LR threshold value is near zero. Figure 26 shows a close-up view of the region between -2.5 and 2.5 and Table 10 shows the percentage of false positive and false negative classifications as the log-likelihood ratio threshold value varies from -2.0 to 2.0, illustrating the trade-off between the number of incorrectly discovered pedigrees (Type I) and the number of missed pedigrees (Type II).

The distribution curves in Figure 24 above approximate a normal distribution, given by:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2 / (2\sigma^2)}$$

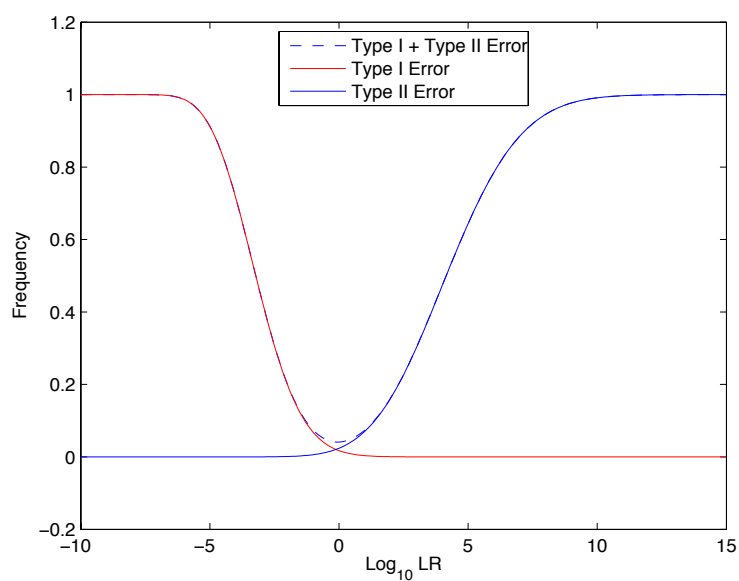


Figure 25: Comparison of Type I and Type II error for the full-sibling relationship.

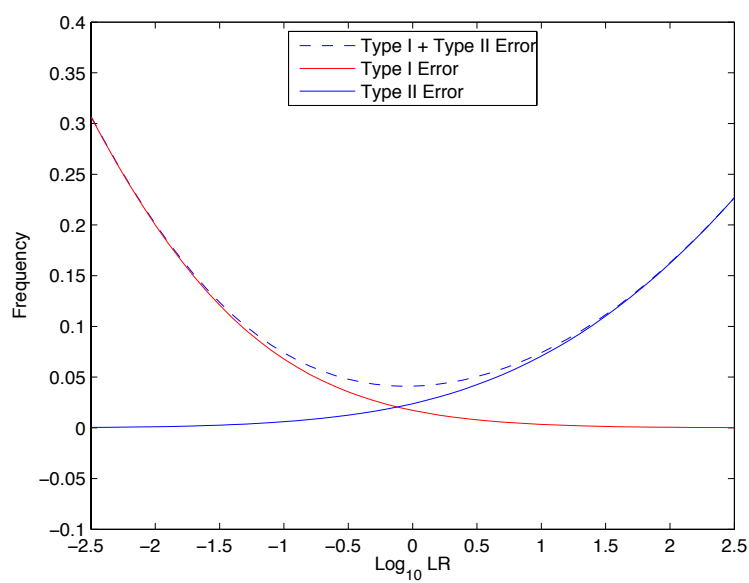


Figure 26: Close-up comparison of Type I and Type II error for the Log-LR range of -2.5 to 2.5.

where μ is the mean and σ is the standard deviation of the curve. The curves can be compared to those of Buckleton et al., shown in Figure 27 [6]. The means and standard deviations of the curves in Figure 24 and the curves from Buckleton are shown in Table 11. The measurements from the Buckleton curves were interpolated and approximated from the original curves on page 412 of [6], which also show the distributions of half-sibling and cousin relationships. The data from the specimen generator (Section 3.5) creates full-sibling log-likelihood ratio distributions very close to those of Buckleton. Differences between the two sets of distributions can be explained by possible differences in the choice of minimum allele frequency value and the number of loci used in the profiles (Buckleton used 15 loci [6]). The agreement between the specimen generator curves and the Buckleton curves validates both the generation of specimens and the calculation of likelihood ratios.

4.5 Experiment: Influence of Edge Weight Threshold Value

4.5.1 Objective

The purpose of this experiment was to determine how much influence the choice of edge weight threshold value has on the performance of the discovery

Table 10: Percentage of Type I and Type II error as Log-LR threshold value varies between -2.0 and 2.0.

	Log-Likelihood Ratio Threshold Value										
	-2.0	-1.6	-1.2	-0.8	-0.4	0.0	0.4	0.8	1.2	1.6	2.0
Type I	20	13.5	8.6	5.3	3.1	1.7	0.9	0.5	0.2	0.1	0.1
Type II	0.1	0.2	0.4	0.8	1.4	2.4	3.8	5.8	8.5	12.0	16.2
Sum	20.1	13.7	9.0	6.1	4.5	4.1	4.7	6.3	8.7	12.1	16.3

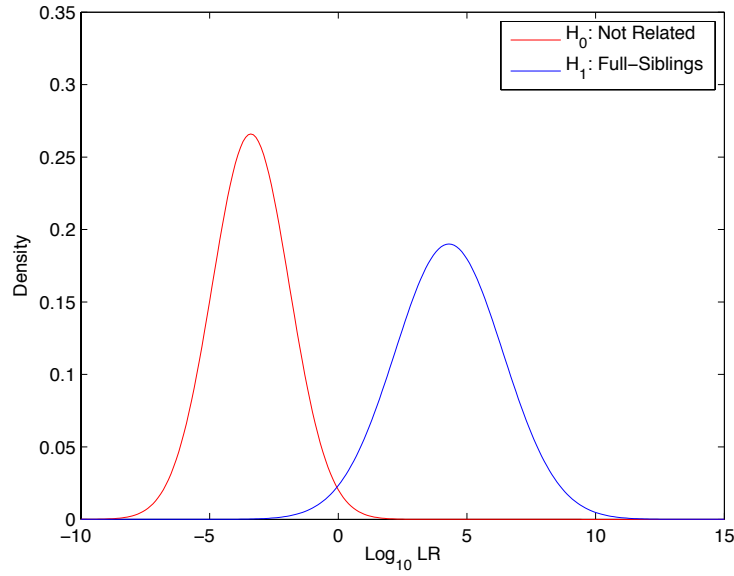


Figure 27: Distributions of the log-likelihood ratio for pairs of unrelated individuals and pairs of individuals related as full-siblings from Buckleton et al. [6].

Table 11: Comparison of LR distributions from Specimen Generator and Buckleton [6].

Data Set	H_0 Full-Sibs		H_1 Full-Sibs	
	Mean	Std. Dev.	Mean	Std. Dev.
Specimen Generator	-3.2	1.4	4.3	2.2
Buckleton et al. [6]	-3.4	1.5	4.3	2.1

algorithm. The data sets used in this experiment consisted of collections containing varying numbers of pedigrees of uniform type and edges representing log-likelihood ratios for parent-child and full-sibling relationships. Additionally, data sets containing varying numbers of individual specimens were used.

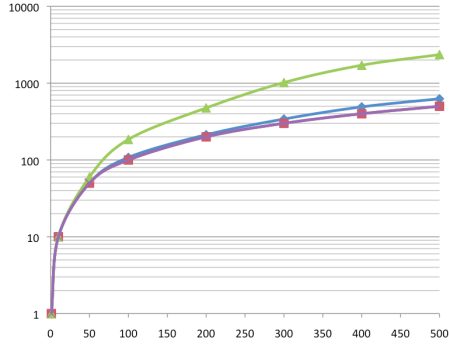
4.5.2 Results

As indicated by the results of the previous experiments, the number of full-sibling edges in the graph degrades the performance of the discovery algorithm, however, an appropriate selection of a likelihood ratio threshold value can reduce the error. In this experiment the edge weight threshold value, which determines whether or not an edge in the data set is accepted as an edge in the graph, was varied with a goal of reducing the number of Sib-LR edges in the graph. The discovery algorithm was run on each of the pedigree collections with each choice of threshold value to determine whether the reduction had a positive impact on the accuracy of discovery.

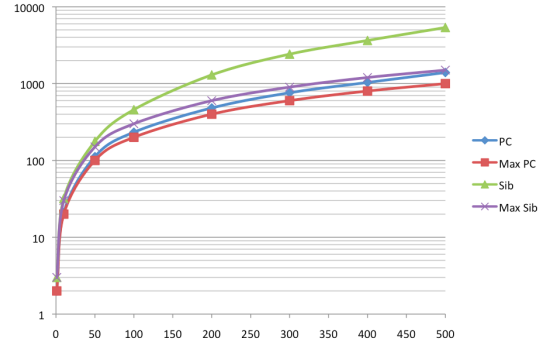
The first series of tests used a threshold value of 0.0 and provided the results shown in the previous experiments. The second series of tests used a threshold value of 1.0. Using log-likelihood ratios, this requires that the hypothesis that two individuals are related as full-siblings must be at least ten times more likely than that of the hypothesis that they are not related. The plots in Figure 28 show that increasing the threshold value to 1.0 greatly reduced the total number of Sib-LR edges in the graph. For example, the number of Sib-LR edges in the collection containing 400 PPCCC pedigrees was reduced from around 40,000 to around 11,000. The plots in Figure 29 show that the composition of the graph has improved slightly for the larger data sets, and the discovery accuracy has improved for each pedigree type, as shown in Figure 30. However, the plots in Figure 28 show that the number of Sib-LR edges in the graph still grows at a much higher rate than the number of true Sib-LR edges because the edge weights are larger than the threshold value, indicating that further increasing the threshold value may yield additional improvements.

The next series of tests used a threshold value of 2.0 with the intent of reducing the number of Sib-LR edges in the graph such that the total number of Sib-LR edges is only slightly higher than the number of true Sib-LR edges in the graph. The expectation is that the composition of the graph will improve and the false Sib-LR

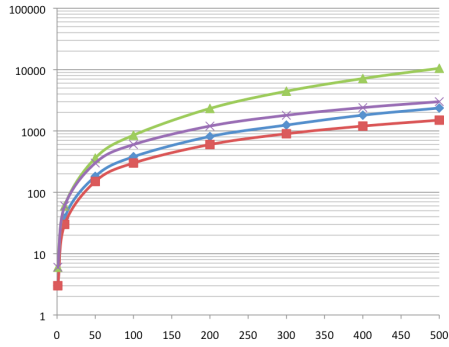
edges will have less of an impact on the discovery algorithm, further increasing the accuracy of discovery. The plots in Figure 31 show that increasing the threshold value to 2.0 has greatly reduced the number of sibling edges as well as the rate of growth for sibling edges. The composition of the graph has also improved, as indicated by Figure 32, which shows a relatively steady ratio of full-sibling to parent-child edges. These improvements have resulted in increased discovery accuracy for each pedigree type, as shown in Figure 33.



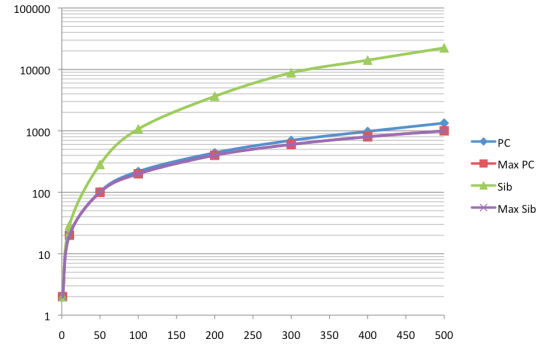
a) PC Type



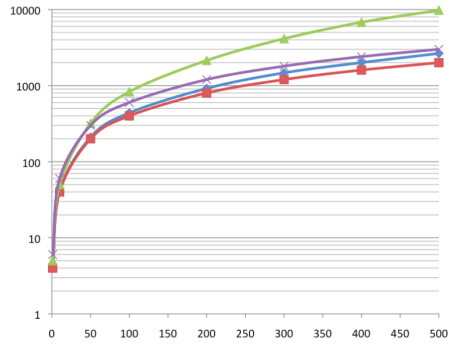
b) PCC Type



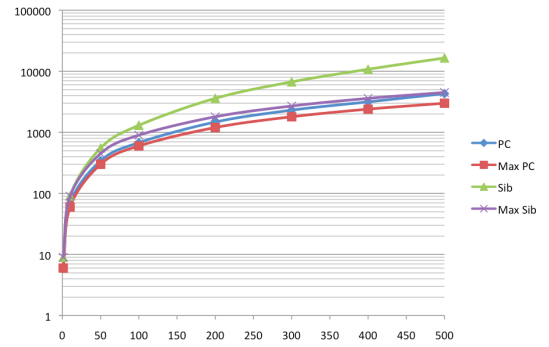
c) PCCC Type



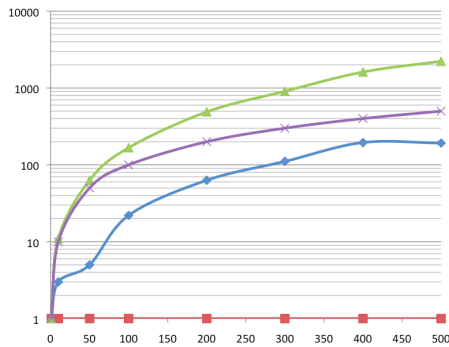
d) PPC Type



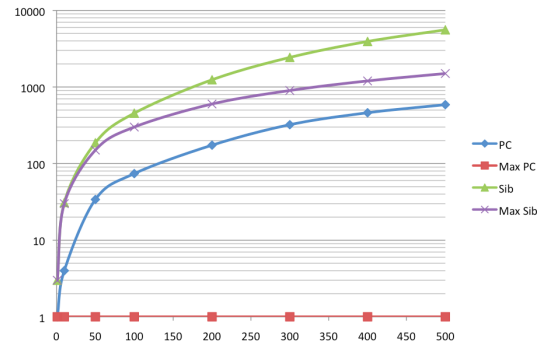
e) PPCC Type



f) PPCCC Type



g) SS Type



h) SSS Type

Figure 28: Number of PC and Sib edges in graph versus the maximum number of true PC and Sib edges with threshold = 1.0.

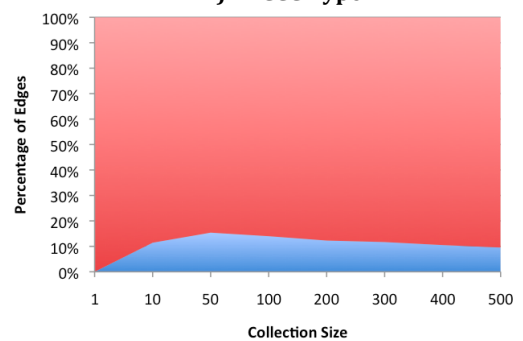
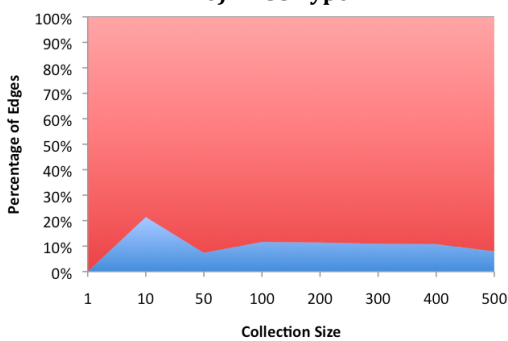
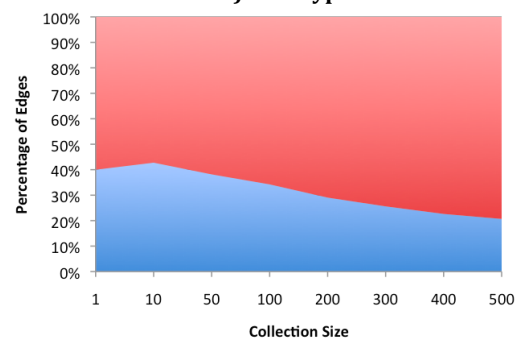
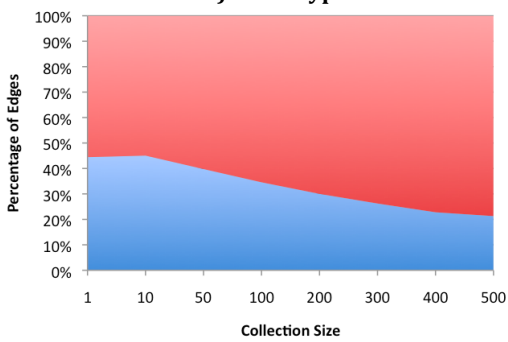
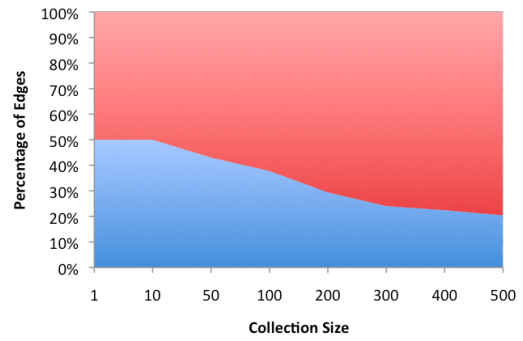
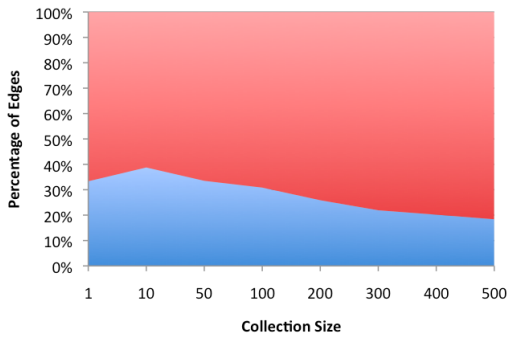
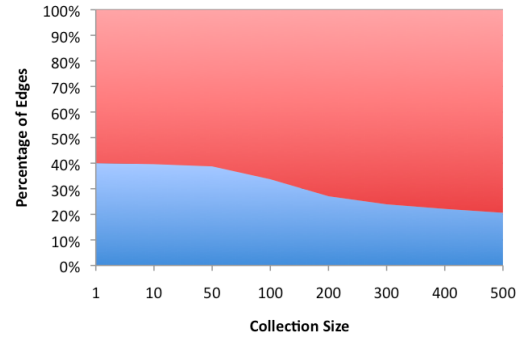
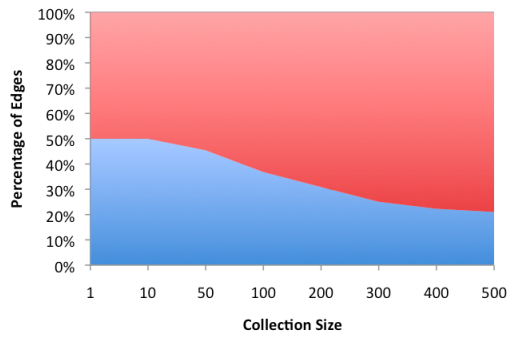


Figure 29: Graph composition with threshold = 1.0.

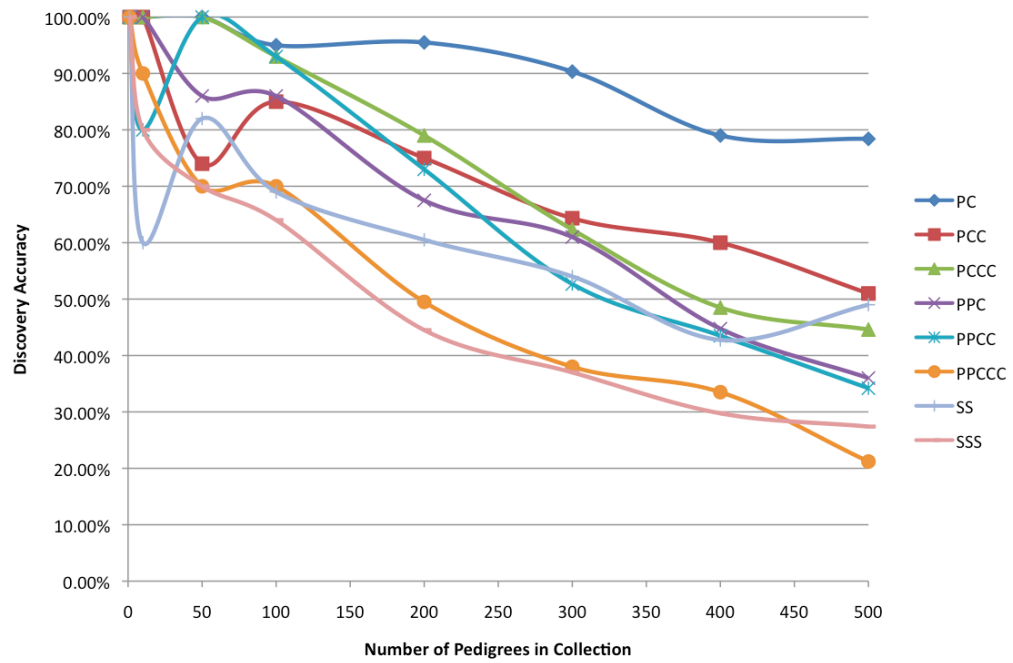
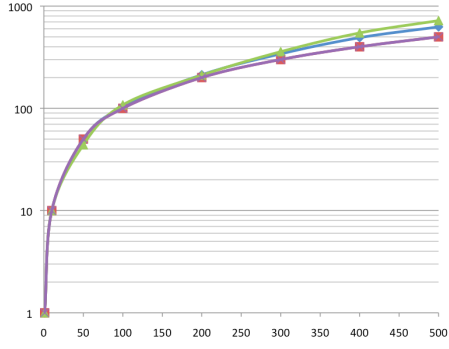
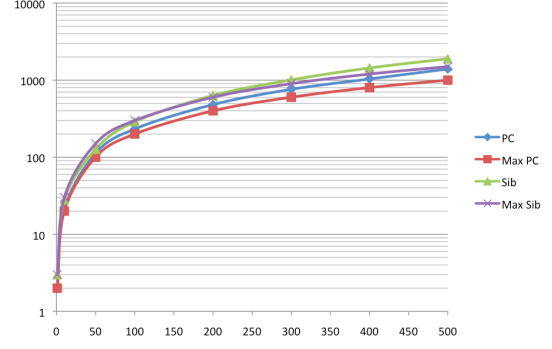


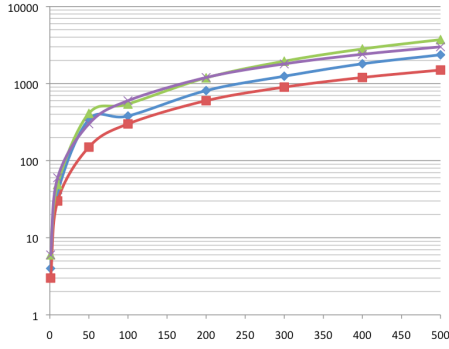
Figure 30: Accuracy of discovering true pedigree with threshold = 1.0.



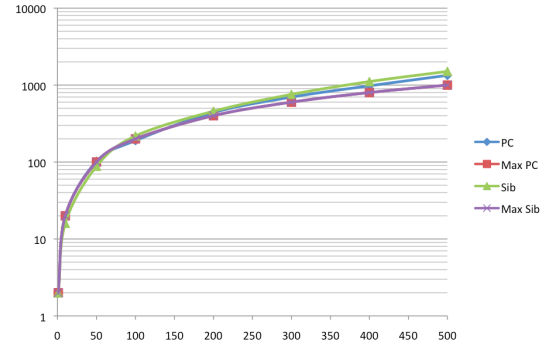
a) PC Type



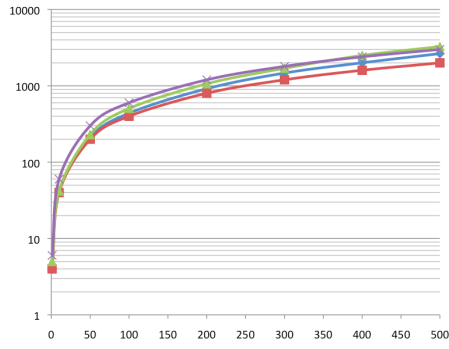
b) PCC Type



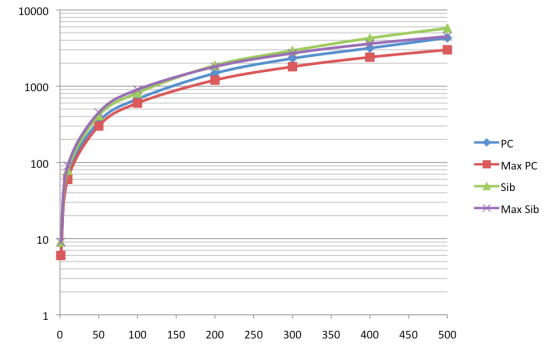
c) PCCC Type



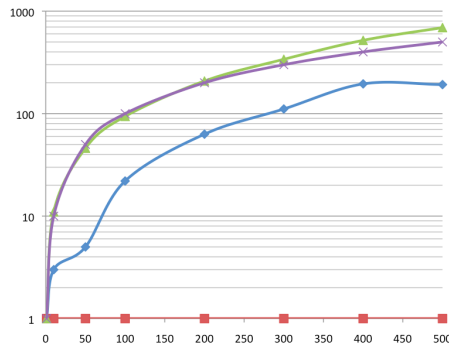
d) PPC Type



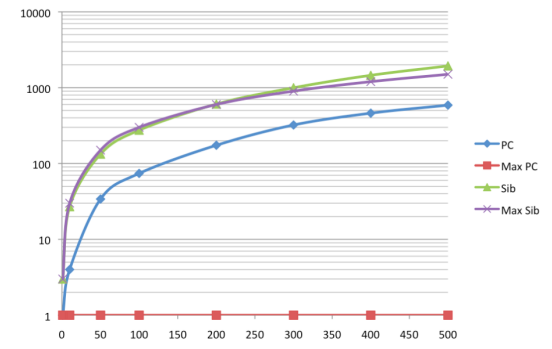
e) PPCC Type



f) PPCCC Type

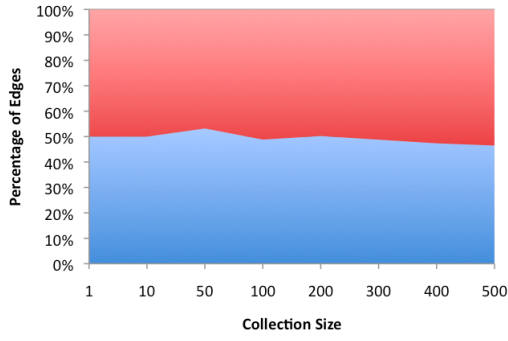


g) SS Type

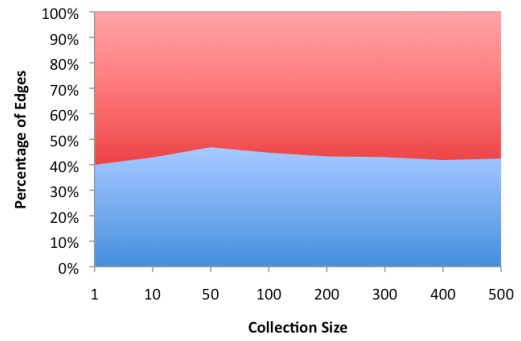


h) SSS Type

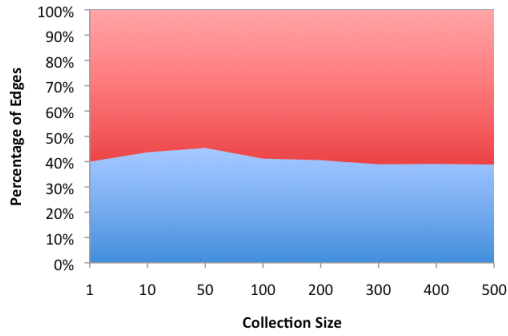
Figure 31: Number of PC and Sib edges in graph versus the maximum number of true PC and Sib edges with threshold = 2.0.



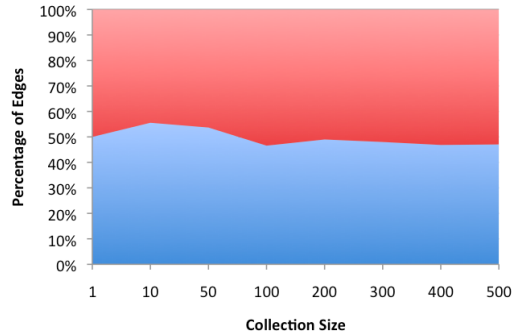
a) PC Type



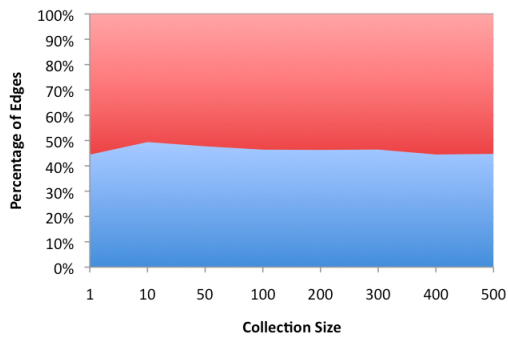
b) PCC Type



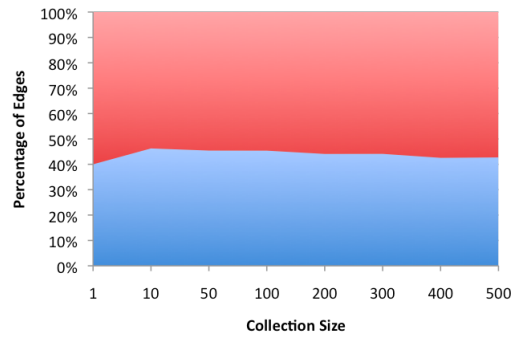
c) PCCC Type



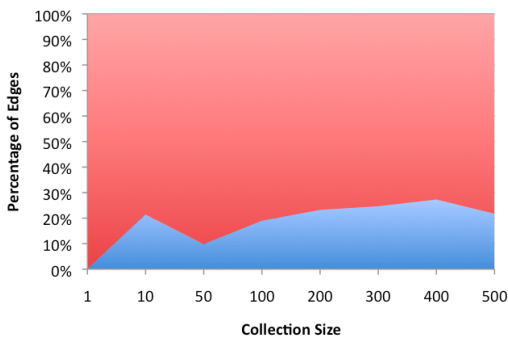
d) PPC Type



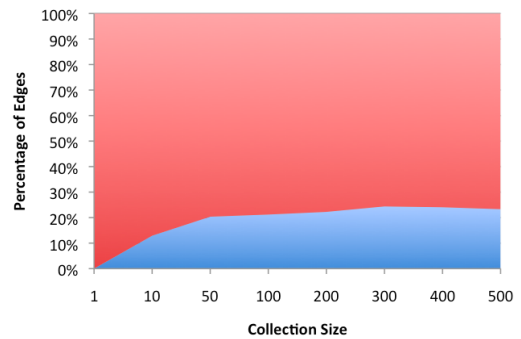
e) PPCC Type



f) PPCCC Type



g) SS Type



h) SSS Type

Figure 32: Graph composition with threshold = 2.0

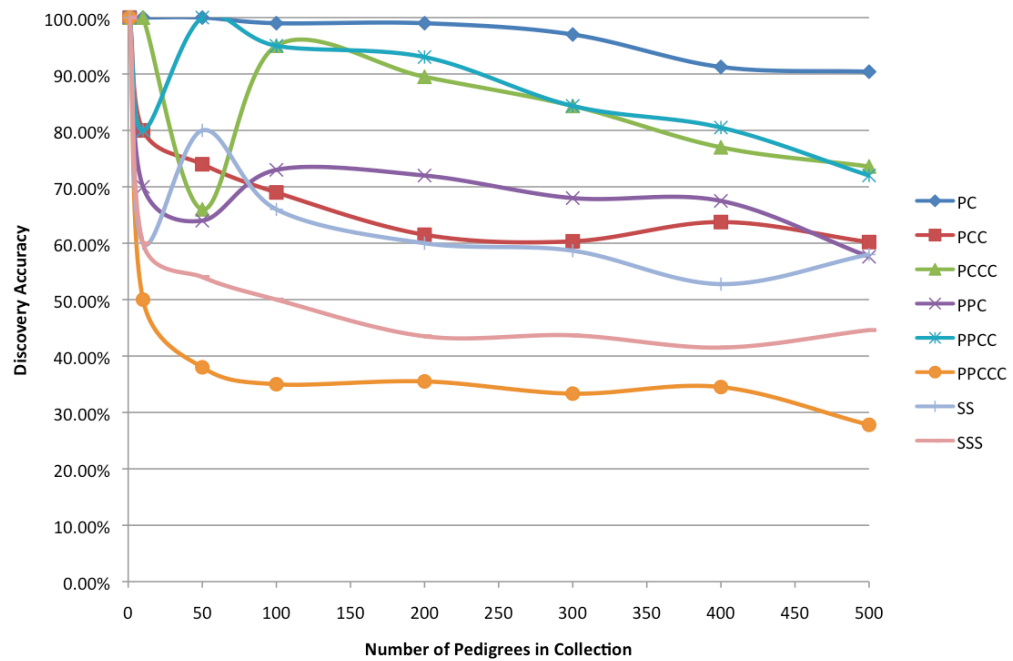


Figure 33: Accuracy of discovering true pedigree with threshold = 2.0.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

While software exists to search for pedigrees among victims of a mass disaster (such as LeClair, et al. [9]), the methods used are tedious and require exhaustive comparisons of the specimens in the data set. The method presented in this thesis automates the discovery process by using Link Discovery Tool to cluster a graph representing a collection of specimens and the potential biological relationships among pairs of specimens in the collection. The clusters produced by LDT represent groups of specimens that are more likely to be related to each other than to specimens in other clusters. The snip algorithm reduces the clusters into candidate clusters, which can be analyzed and classified using the analysis algorithm. The discovery algorithm is limited in that it cannot identify arbitrary pedigree types and cannot provide certainty in the classification of pedigrees.

The results of the experiments described in the previous chapter show that the pedigree discovery algorithm is able to accurately discover different types of pedigrees. The number of full-sibling edges in the graph created by Link Discovery Tool for a given data collection has a large negative impact on the performance of the algorithm presented in Chapter 3. The use of an edge weight threshold value was shown to greatly reduce the number of full-sibling edges in the graph and improve the performance of the algorithm. The PC-type pedigree was consistently the most-easily identified pedigree while the PCCC and SSS pedigrees were the most difficult to identify. This is an expected result, as the PC-type is the simplest pedigree with the fewest number of full-sibling edges.

Given the results of the experiments, the discovery algorithm and extension of Link Discovery Tool presented in this thesis have shown to be a viable method of automatically identifying pedigrees in collections of specimens. While the algorithm did not perfectly discover all types pedigrees, this is an expected result since likelihood ratios do not provide certainty. Several ambiguous cases can arise in the data set in which educated guesses are made as to the most likely classification of the pedigree. These cases are often exaggerated by large numbers of false full-sibling edges and make it very difficult to accurately determine the correct pedigree structure. However, Link Discovery Tool can identify simple pedigree types and can be used to reduce the data set to a more manageable size.

The inclusion of gender markers, such as Amelogenin, and lineage markers, such as those found on Y-STR and Mitochondrial DNA, as well as metadata, such as age, should improve the classification accuracy of the pedigree discovery algorithm by allowing candidate clusters to be validated for gender and lineage. For example, a candidate cluster representing a potential PPC pedigree can be marked as invalid if the two parent specimens are the same gender.

5.2 Future Work

Future work on the pedigree discovery algorithm is likely to further improve the performance of the algorithm. The choice of edge weight threshold value was shown to have a large impact on the accuracy of the algorithm in general, although performance decreased for small data sets for some of the pedigree types. A study to determine an optimal threshold value for each type of pedigree as collection size increases may allow LDT to determine the best threshold value for a particular data set, further improving performance. Additionally, since the discovery algorithm requires the graph to be clustered by Link Discovery Tool, the ability to solve for a

steady-state configuration of the graph, rather than incrementally moving towards a steady-state configuration, is likely to further improve performance by ensuring that graph clusters have fully formed and are in the correct locations relative to one another. Finally, the use of metadata, such as age, as well as gender and lineage markers, such as Amelogenin, Y-STR, and Mitochondrial DNA, should further increase the accuracy of the algorithm by allowing LDT to determine whether candidate clusters are in fact viable candidate clusters.

References

References

- [1] Horn, R. D., Birdwell, J. D., and Leedy, L. W. *Link Discovery Tool*, Presented at the ONDCP/CTAC International Symposium, Chicago, IL, August 18-22, 1997.
- [2] Goodwin, W., Linacre, A., Hadi, S., *An Introduction to Forensic Genetics*. Jon Wiley & Sons, Ltd., 2007.
- [3] Fung, W. K., and Hu, Y., *Statistical DNA Forensics*. Jon Wiley & Sons, Ltd., 2008.
- [4] Rudin, N., and Inman, K., *An Introduction to Forensic DNA Analysis*. 2nd Ed., 2002.
- [5] NIST, "Short Tandem Repeat DNA Internet DataBase." September 5, 2008. <<http://www.cstl.nist.gov/biotech/strbase>>.
- [6] Buckleton, J., Triggs, C. M., Walsh, S. J., *Forensic DNA Evidence Interpretation*. CRC Press, 2005.
- [7] Balding, D.J., and Nichols, R.A., "DNA profile match probability calculation: how to allow for population stratification, relatedness, database selection and single bands," *Forensic Science Int.*, vol. 64, 1994, pp. 125-140.
- [8] Evett, I.W., and Weir, B.S., *Interpreting DNA Evidence – Statistical Genetics for Forensic Scientists*. Sinauer Associates, Inc., 1998.

- [9] LeClair, B., Frégeau, C. J., Bowen, K. L., and Fourney, R. M., “Enhanced Kinship Analysis and STR-based DNA Typing for Human Identification in Mass Fatality Incidents: The Swissair Flight 111 Disaster”, *Journal of Forensic Science*, vol. 49, no. 5, September 2004.
- [10] Cannings, C., Thompson, E. A., and Skolnick, M. H., “Probability Functions on Complex Pedigrees.”, *Advanced Applied Probability* **10**, 26-61, 1978.
- [11] Trudeau, R. J., *Introduction to Graph Theory*. Dover Publications, Inc., 1993.
- [12] NIST, “Population Studies Conducted by the NIST Forensics/Human Identity Project Team,” September 5, 2008.
<<http://www.cstl.nist.gov/biotech/strbase/NISTpop.htm>>.
- [13] Muller, M. E., “An Inverse Method for the Generation of Random Normal Deviates on Large-Scale Computers.”, *Mathematical Tables and Other Aids to Computation*, vol. 12, no. 3, July 1958, pp. 167-174.
- [14] Butler, J.M., *Forensic DNA Typing – Biology & Technology behind STR Markers*. Academic Press, 2001.
- [15] Chackraborty, R., Birdwell, J. D., Wang, T. W., and Stansberry, D. V., *Methods of Associating an Unknown Biological Specimen with a Family*. U.S. Patent Application 11/467,834, Publication Number 2008/0040046. August 28, 2006.

Appendix A: Tables and Derivations for Likelihood Ratios

Appendix A: Tables and Derivations for Likelihood Ratios

A.1 Probability of Sharing IBD Alleles

The numerator in the likelihood ratios can be evaluated using the method of Balding and Nichols, which states that if we consider any two individuals, then there will be four alleles at a given locus and one of the individuals can share zero, one, or two of the alleles IBD with the other individual [6]. This provides the three possible events:

Z_0 : 0 alleles are IBD

Z_1 : 1 allele is IBD

Z_2 : 2 alleles are IBD

The probability of occurrence of these events is dependent upon the relationship in question. For example, given a parent-child relationship, the child genotype will share exactly one allele IBD with the parent genotype and $Pr(Z_1)=1$, and $Pr(Z_0)=Pr(Z_2)=0$. The probabilities for the full-sibling relationship can be similarly determined using the principle of Mendelian inheritance and are shown in Table 12 and in [6].

Table 12: Probabilities that two individuals share a number of IBD alleles given a relationship [6].

Relationship	$Pr(Z_0)$	$Pr(Z_1)$	$Pr(Z_2)$
Parent-Child	0	1	0
Full-Sibling	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

A.2 Probabilities and Likelihood Ratios for a Parent-Child Relationship

The parent-child relationship is the simplest relationship to evaluate between two individuals. Since the parent and child share exactly one IBD allele at each locus (unless a mutation has occurred), the likelihood ratio will evaluate to zero when two individuals cannot be related. A non-zero LR, therefore, indicates that the biological relationship may exist. The likelihood ratio for a parent-child relationship is stated as follows [6]:

$$LR = \frac{\Pr(G_{S1} | G_{S2}, H_{PC})}{\Pr(G_{S1} | H_0)} = \frac{\Pr(G_{S2} | G_{S1}, H_{PC})}{\Pr(G_{S2} | H_0)}$$

The denominator above is the probability of observing the genotype G_{Sx} of specimen S_x at random in the population. The probability of observing a genotype at a given locus is:

$$\Pr^i(G^i | H_0) = p_a^2$$

for a homozygous genotype consisting of the profile aa at locus i , and:

$$\Pr^i(G^i | H_0) = 2p_a p_b$$

for a heterozygous genotype consisting of the profile ab at locus i [6].

The conditional probability in the likelihood ratio has a slightly different form for each possible combination of genotypes of the parent and child specimens. The cases for each combination are shown in Table 13 [6]. The likelihood ratios for each case are from Buckelton et al. [6] and are shown in Table 14. Since the allele values are statistically independent under the assumption that the population is under

Hardy-Weinberg equilibrium and linkage equilibrium, the combined likelihood ratio over all loci in the profile can be calculated as:

$$LR = \prod_{i=1}^L \frac{\Pr^i(G_{S1}^i | G_{S2}^i, H_{PC})}{\Pr^i(G_{S1}^i | H_0)} = \prod_{i=1}^L \frac{\Pr^i(G_{S2}^i | G_{S1}^i, H_{PC})}{\Pr^i(G_{S2}^i | H_0)}$$

where L is the number of loci in the profiles.

Table 13: Conditional probabilities for the parent-child relationship from Buckleton [6].

Genotype of specimen S_2	Genotype of specimen S_1	Conditional probability of parent-child relationship
G_{S2}	G_{S1}	$\Pr^i = \Pr^i(G_{S1}^i G_{S2}^i, H_{PC})$
aa	aa	$\Pr^i = \frac{1}{2} \Pr(Z_1) p_a + \frac{1}{2} \Pr(Z_1) p_a = p_a$
	ab	$\Pr^i = \frac{1}{2} \Pr(Z_1) p_b + \frac{1}{2} \Pr(Z_1) p_b = p_b$
ab	aa	$\Pr^i = \frac{1}{2} \Pr(Z_1) p_a = \frac{1}{2} p_a$
	ab	$\Pr^i = \frac{1}{2} \Pr(Z_1) p_b + \frac{1}{2} \Pr(Z_1) p_a = \frac{1}{2} (p_a + p_b)$
	ac	$\Pr^i = \frac{1}{2} \Pr(Z_1) p_c = \frac{1}{2} p_c$
	bb	$\Pr^i = \frac{1}{2} \Pr(Z_1) p_b = \frac{1}{2} p_b$

Table 14: Likelihood ratios for the parent-child relationship from Buckleton [6].

Genotype of specimen S_2 G_{S2}	Genotype of specimen S_1 G_{S1}	Likelihood ratio of parent-child relationship $LR^i = \frac{\Pr^i(G_{S1}^i G_{S2}^i, H_{PC})}{\Pr^i(G_{S1}^i H_0)}$
aa	aa	$LR^i = \frac{p_a}{p_a^2} = \frac{1}{p_a}$
	ab	$LR^i = \frac{p_b}{2p_a p_b} = \frac{1}{2p_a}$
ab	aa	$LR^i = \frac{\frac{1}{2}p_a}{p_a^2} = \frac{1}{2p_a}$
	ab	$LR^i = \frac{\frac{1}{2}(p_a + p_b)}{2p_a p_b} = \frac{p_a + p_b}{4p_a p_b}$
	ac	$LR^i = \frac{\frac{1}{2}p_c}{2p_a p_c} = \frac{1}{4p_a}$
	bb	$LR^i = \frac{\frac{1}{2}p_b}{p_b^2} = \frac{1}{2p_b}$

A.3 Probabilities and Likelihood Ratios for a Full-Sibling Relationship

The full-sibling relationship can be evaluated similarly to the parent-child relationship in the previous section except that full-siblings can also share zero or two alleles IBD. The conditional probabilities for each of the possible full-sibling genotype combinations are shown in Table 15 [6]. The likelihood ratios for each case are from Buckleton et al. [6] and are shown in Table 16. The combined likelihood ratio over all loci can be calculated as in the previous section.

Table 15: Conditional probabilities for the full-sibling relationship from Buckleton [6].

Genotype of specimen S_2	Genotype of specimen S_1	Conditional probability of full-sibling relationship $\Pr^i = \Pr^i(G_{S1}^i G_{S2}^i, H_S)$
G_{S2}	G_{S1}	
aa	aa	$\Pr^i = \Pr(Z_2) + \frac{1}{2}\Pr(Z_1)p_a + \frac{1}{2}\Pr(Z_1)p_a + \Pr(Z_0)p_a^2$ $= \frac{1}{4} + \frac{1}{4}p_a + \frac{1}{4}p_a + \frac{1}{4}p_a^2 = \frac{1}{4}(1 + 2p_a + p_a^2)$ $= \frac{1}{4}(1 + p_a)^2$
	ab	$\Pr^i = \frac{1}{2}\Pr(Z_1)p_b + \frac{1}{2}\Pr(Z_1)p_b + \Pr(Z_0)2p_ap_b$ $= \frac{1}{4}p_b + \frac{1}{4}p_b + \frac{1}{2}p_ap_b = \frac{1}{2}p_b + \frac{1}{2}p_ap_b$
	bb	$\Pr^i = \Pr(Z_0)p_b^2 = \frac{1}{4}p_b^2$
	bc	$\Pr^i = \Pr(Z_0)2p_bp_c = \frac{1}{2}p_bp_c$
ab	aa	$\Pr^i = \frac{1}{2}\Pr(Z_1)p_a + \Pr(Z_0)p_a^2 = \frac{1}{4}p_a + \frac{1}{4}p_a^2 = \frac{1}{4}p_a(1 + p_a)$
	ab	$\Pr^i = \Pr(Z_2) + \frac{1}{2}\Pr(Z_1)p_b + \frac{1}{2}\Pr(Z_1)p_a + \Pr(Z_0)2p_ap_b$ $= \frac{1}{4} + \frac{1}{4}p_b + \frac{1}{4}p_a + \frac{1}{2}p_ap_b = \frac{1}{4}(1 + p_a + p_b + 2p_ap_b)$
	ac	$\Pr^i = \frac{1}{2}\Pr(Z_1)p_c + \Pr(Z_0)2p_ap_c = \frac{1}{4}p_c + \frac{1}{2}p_ap_c$ $= \frac{1}{4}p_c(1 + 2p_a)$
	cc	$\Pr^i = \Pr(Z_0)p_c^2 = \frac{1}{4}p_c^2$
	cd	$\Pr^i = \Pr(Z_0)2p_cp_d = \frac{1}{2}p_cp_d$

Table 16: Likelihood ratios for the full-sibling relationship from Buckleton [6].

Genotype of specimen S_2	Genotype of specimen S_1	Likelihood ratio of full-sibling relationship $LR^i = \frac{\Pr^i(G_{S1}^i G_{S2}^i, H_S)}{\Pr^i(G_{S1}^i H_0)}$
G_{S2}	G_{S1}	
aa	aa	$LR^i = \frac{\frac{1}{4}(1+p_a)^2}{p_a^2} = \frac{(1+p_a)^2}{4p_a^2}$
	ab	$LR^i = \frac{\frac{1}{2}p_b + \frac{1}{2}p_ap_b}{2p_ap_b} = \frac{1+p_a}{4p_a}$
	bb	$LR^i = \frac{\frac{1}{4}p_b^2}{p_b^2} = \frac{1}{4}$
	bc	$LR^i = \frac{\frac{1}{2}p_bp_c}{2p_bp_c} = \frac{1}{4}$
ab	aa	$LR^i = \frac{\frac{1}{4}p_a(1+p_a)}{p_a^2} = \frac{1+p_a}{4p_a}$
	ab	$LR^i = \frac{\frac{1}{4}(1+p_a+p_b+2p_ap_b)}{2p_ap_b} = \frac{1+p_a+p_b+2p_ap_b}{8p_ap_b}$
	ac	$LR^i = \frac{\frac{1}{4}p_c(1+2p_a)}{2p_ap_c} = \frac{1+2p_a}{8p_a}$
	cc	$LR^i = \frac{\frac{1}{4}p_c^2}{p_c^2} = \frac{1}{4}$
	cd	$LR^i = \frac{\frac{1}{2}p_cp_d}{2p_cp_d} = \frac{1}{4}$

Appendix B: Pedigree Classification Decision Rules

Appendix B: Pedigree Classification Decision Rules

The decision rules discussed here are used in the pedigree discovery analysis algorithm (Section 3.8) to classify candidate clusters. The clusters are classified by measuring certain characteristics of the cluster, including the size of the cluster, the number of parent-child edges, the number of full-sibling edges, the parent-child edge degree of each vertex, and the full-sibling edge degree of each vertex. Table 17 lists the characteristics for each pedigree type that is classifiable by the analysis algorithm.

The analysis algorithm characterizes candidate clusters by matching the cluster characteristics to those of the appropriate pedigree type. In some cases, the cluster characteristics will match the characteristics of one pedigree type exactly. In such cases, the classification of the cluster is straightforward using Table 17. In other cases, however, the cluster characteristics will not match the characteristics of any of the pedigree types, due to a missing edge (removed by snipping) or an extra edge (a false edge that wasn't snipped). If the removal of the extra edge, or the addition of the missing edge (if it existed before snipping) does not produce a viable match to a pedigree type, then the cluster is labeled as an unknown type and re-snipped until a matching pedigree type can be determined. However, if removal or addition of a single edge does produce a viable match to a pedigree type, then the cluster can be classified.

The assumptions used to guide the classification process in such cases are shown in Table 18, which lists the true type of the mismatched pedigree with the characteristics that allow the pedigree to be classified, and the associated classification, for viable mismatched clusters. For each pedigree type, the presence of an extra edge, or the absence of a required edge must produce a viable and likely pedigree in order to be classified. For example, consider the PCCC pedigree, which consists of one parent and three children. This pedigree will contain three parent-

child edges (between the parent and each child) and six full-sibling edges (three between the parent and three between the children). However, if one of the full-sibling relationships were strong enough that a parent-child edge also appeared between the siblings, the cluster would now contain four parent-child edges: three between the parent and each child and an extra between two of the children. In this case it is more likely that the extra parent-child edge is a result of a strong full-sibling relationship than if the two specimens connected by the edge were unrelated (since an unrelated specimen would not likely share edges with the other members in the pedigree as well), and so the cluster can be matched correctly to the PCCC pedigree type.

Table 17: Pedigree characteristic for decision rules in the analysis algorithm.

Pedigree	
Type	Relevant Pedigree Characteristics
Individual (I)	Size = 1
PC	Size = 2, Number of PC edges = 1, Number of Sib edges = 1
PCC	Size = 3, Number of PC edges = 2, Number of Sib edges = 3, parent-member PC degree = 2, child-member PC degree = 1, parent-member Sib degree = 2, child-member Sib degree = 2
PPCC	Size = 4, Number of PC edges = 3, Number of Sib edges = 6, parent-member PC degree = 3, child-member PC degree = 1, parent-member Sib degree = 3, child-member Sib degree = 3
PPC	Size = 3, Number of PC edges = 2, Number of Sib edges = 2, parent-member PC degree = 1, child-member PC degree = 2 parent-member Sib degree = 1, child-member Sib degree = 2
PPCC	Size = 4, Number of PC edges = 4, Number of Sib edges = 5, parent-member PC degree = 2, child-member PC degree = 2, parent-member Sib degree = 2, child-member Sib degree = 3
PPCCC	Size = 5, Number of PC edges = 6, Number of Sib edges = 9, parent-member PC degree = 3, child-member PC degree = 2, parent-member Sib degree = 3, child-member Sib degree = 4
SS	Size = 2, Number of PC edges = 0, Number of Sib edges = 1
SSS	Size = 3, Number of PC edges = 0, Number of Sib edges = 3, sibling-member PC degree = 0, sibling-member Sib degree = 2

Table 18: Pedigree characteristics for classifying clusters with additional or missing edges.

Mismatched		Classified
Pedigree	Characteristics of Mismatched Pedigree	As
PC	Missing PC edge	SS
	Missing Sib edge	PC
PCC	Additional PC edge	PCC
	Missing PC edge	SSS
	Missing Sib edge (between children)	PPC
	Missing Sib edge (between parent and child)	PCC
PCCC	Additional PC edge	PCCC
	Missing PC edge	?
	Missing Sib edge (between parent and child)	PCCC
	Missing Sib edge (between children)	PCCC
PPC	Additional PC edge	?
	Additional Sib edge	PCC
	Missing PC edge	?
	Missing Sib edge	PPC
PPCC	Additional PC edge (between parents)	?
	Additional PC edge (between children)	PPCC
	Additional Sib edge	PPCC
	Missing PC edge	PPCC
	Missing Sib edge (between parent and child)	PPCC
	Missing Sib edge (between children)	PPCC
PPCCC	Additional PC edge (between parents)	?
	Additional PC edge (between children)	PPCCC
	Additional Sib edge	PPCCC
	Missing PC edge	PPCCC
	Missing Sib edge (between parent and child)	PPCCC
	Missing Sib edge (between children)	PPCCC

Mismatched		Classified
Pedigree	Characteristics of Mismatched Pedigree	As
SS	Additional PC edge	PC
SSS	Additional PC edge	SSS
	Missing Sib edge	SSS

Vita

Alex Haun was born in Johnson City, Tennessee in 1981. He grew up in East Tennessee, where he studied percussion while attending high school at Sullivan Central. His love for music and the Volunteers led him to the University of Tennessee, where he was a squad leader and member of the drumline in the Pride of the Southland Marching Band. His love for computers and electronics led him to study in the field of Computer Science. After finishing his Bachelor of Science in Computer Science in 2004, he entered the engineering program and began work on a Bachelor of Science in Computer Engineering, which he completed in 2008. He immediately began his pursuit of a Master of Science in Computer Engineering and was provided with the opportunity to work in the Laboratory for Information Technologies on campus. His work at LIT involving forensic DNA analysis kindled his interest in this area and the research topic presented in this thesis. He completed his Master of Science in May 2010 and immediately began working as an Embedded Systems Engineer in Indianapolis, Indiana.