A Genetic Risk System for Genetic Counselors

by

Jason C. Miller

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 21, 1999

© Copyright 1999 Jason C. Miller. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author

Department of Electrical Engineering and Computer Science
May 21, 1999

Certified by

Peter Szolovits
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by

Arthur C. Smith
Chairman, Department Committee on Graduate Theses
A Genetic Risk System for Genetic Counselors
by
Jason C. Miller

Submitted to the
Department of Electrical Engineering and Computer Science

May 21, 1999

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis describes a program named GenInfer+ which uses Bayesian belief networks to model the inheritance of genetic diseases. GenInfer+ is based upon an earlier genetic risk program created by Peter Szolovits named GenInferII which used a Bayesian network to calculate risks to individuals for simple inheritable diseases.

This thesis had two goals. The first goal was to design a useful tool for genetic counselors in genetic risk analysis. This entailed not only creating an easy to use system, but also designing an extensible system which would easily support the addition of future functionality needed by genetic counselors. Particular attention was paid to how genetic diseases can be modeled using a standard Bayesian network.

The second goal of this thesis was to create a system which would serve as a test bed for research into Bayesian networks and their application to practical problems. The system was designed using independent modules which communicate through a standard interface. New implementations of Bayesian networks and inheritable diseases can be inserted without disrupting the whole system.

Thesis Supervisor: Peter Szolovits
Title: Professor of Computer Science and Engineering
Acknowledgments

I'd like to thank the following people for their help:

- Peter Szolovits and Jon Doyle for recommending a project in genetic counseling.
- Peter Szolovits for being a great resource in designing the genetic system.
- Daniel Nigrin for his help and knowledge of secure applets.
- Fabio Cozman at CMU for letting me use his Java Bayes program for Bayesian networks.
- Tim Macinta for his expert software knowledge of Java and for helping me get past some major software roadblocks.
- Left Quad for helping me stay focused during the course of this thesis.
# Table of Contents

ABSTRACT ......................................................................................................................... 2

TABLE OF CONTENTS ............................................................................................................. 4

CHAPTER 1 - INTRODUCTION ............................................................................................... 5
  1.1 OVERVIEW .................................................................................................................... 7
  1.2 HUMAN GENETICS ....................................................................................................... 8
  1.3 INHERITABLE DISEASES ............................................................................................. 10
  1.4 GENETIC COUNSELING ................................................................................................ 12

CHAPTER 2 - CREATING A GENETIC RISK SYSTEM ........................................................ 15
  2.1 PREVIOUS PROGRAMS DEALING WITH INHERITABLE DISEASES ................................ 15
  2.2 GENINFERII .................................................................................................................. 18
  2.3 THE ELEMENTS OF AN IDEAL SOLUTION .................................................................... 20
    2.3.1 Design requirements ............................................................................................... 20
    2.3.2 Functionality requirements .................................................................................... 21
    2.3.3 User interface requirements .................................................................................... 22

CHAPTER 3 - GENINFER+ ................................................................................................... 24
  3.1 DESIGN ......................................................................................................................... 24
  3.2 FUNCTIONALITY ............................................................................................................ 29
  3.3 USER INTERFACE .......................................................................................................... 31

CHAPTER 4 - PEDIGREE ANALYSIS USING BAYESIAN NETWORK ..................................... 35
  4.1 SINGLE LOCUS TWO ALLELE DISEASES ..................................................................... 36
  4.2 MULTIPLE ALLELES ....................................................................................................... 39
  4.3 EXPRESSIVITY .............................................................................................................. 40
  4.4 MUTATION ..................................................................................................................... 42
  4.5 MULTIPLE PHENOTYPES ............................................................................................... 42
  4.6 MULTILocal DISEASEs ................................................................................................. 43
  4.7 GENETIC SCREENING .................................................................................................. 44
  4.8 MULTIPLE BIRTHS ......................................................................................................... 45

CHAPTER 5 - FUTURE DIRECTIONS .................................................................................... 47
  5.1 CURRENT LIMITATIONS ............................................................................................. 47
  5.2 CONCLUSION ............................................................................................................... 49

BIBLIOGRAPHY .................................................................................................................... 50

APPENDIX A ......................................................................................................................... 52
  APPENDIX A.1 - BAYES NETWORK INTERFACE ................................................................ 52
  APPENDIX A.2 - BAYES NETWORK NODE INTERFACE ..................................................... 53
  APPENDIX A.3 - EXCERPTS FROM DISEASE INTERFACE .................................................. 54
  APPENDIX A.4 - EXCERPTS FROM THE GENETIC SCREEN DESCRIPTOR INTERFACE .... 54
Chapter 1
Introduction

Medical practitioners must make critical clinical decisions with incomplete information. A doctor must recommend treatment for a patient with multiple systems that could be caused by several different disorders. Because some treatments have potential risks, a doctor must attempt to minimize the risk to his patient using the available information. Besides physical risk, a doctor must also attempt to minimize the financial costs involved in treating a patient without seriously compromising the quality of his treatment. For example, patients with head injuries are rarely given an expensive brain scan for fatal brain hemorrhaging because of its low likelihood and the high cost of the procedure. Medical practice is really a complex game of probability. Determining the most likely explanation for a patient's condition and minimizing the risks and costs of treatment is a very difficult process. A doctor must understand the causal relationships between disorders and symptoms. He must also understand the relative likelihood of possible disorders in the patient based on any available information. As new information about the patient's condition becomes available, the doctor must update his beliefs based on the most probably explanation.
Uncertainty is also an important issue in artificial intelligence. Often a program must make conclusions based on an incomplete picture of important issues or the nature of their relationships. Just like a medical doctor, a program must update its view of the world based on the most currently available information. The Bayesian belief network has been the most successful system developed in artificial intelligence to model probabilistic elements and their relationships. Bayesian belief networks explicitly state the probabilistic relationship between events and use Bayes' Rule to update the likelihood of events using available information.

With the success of Bayesian networks in artificial intelligence, many attempts have been made to use them in medical decision-making. This thesis will focus on the application of Bayesian networks to the field of genetic counseling. Genetic counselors must advise a patient about his risk for genetic disorders with incomplete information of the patient's family history and his genetic makeup. Genetic counselors spend substantial time analyzing a patient's family history and demographics to estimate his risk and the risk for his future children. Because genetic counselors help patients make extremely sensitive decisions, finding an accurate estimate of risk is critical. For example, if a patient is deciding whether or not to have children because of high risk for a genetic disorder, the genetic counselor wants to be certain that he has correctly taken all information into account in determining the genetic risk.

Numerous artificial intelligence systems have been developed to use Bayesian networks to aid genetic counselors. The goal of these systems has generally been to demonstrate the feasibility of implementing a Bayesian network and using Bayesian networks to make real time risk estimates for medical purposes. The focus has been on
creating an efficient and fully functional Bayesian network. This research was very successful in breaking new ground in the study of Bayesian networks. However, these system had only limited success in making the leap from the research lab to the genetic counselor’s office.

The goal of this thesis is twofold. The primary objective is to develop a Bayesian network risk system that will be used by genetic counselors. The practical use of Bayesian networks in medical decision-making may reveal new areas of research that remain unexplored. The second goal of this thesis is to develop a genetic counseling system that allows for future research of Bayesian network implementations. It should be fairly easy to try new Bayesian algorithms or network designs without major software reconstruction of the system. The final system should satisfy the interests of two major groups: the genetic counselors who want a software system that will aid their risk analysis, and artificial intelligence researchers who want a software system to experiment with new ideas in Bayesian networks.

1.1 Overview
The remaining sections of Chapter 1 discuss the basics of genetics and genetic diseases. Chapter 1 also describes the role of genetic counselors and how a genetic risk system like GenInfer+ can help genetic counselors perform this role. Chapter 2 discusses past genetic risk systems, including GenInferII, that have used probabilistic systems to model genetic diseases. Chapter 2 also describes the requirements for an ideal genetic risk system for genetic counselors. Chapter 3 describes the design and operation of the GenInfer+ system. It describes how GenInfer+ can be used by genetic counselors to
determine the genetic risk of patients. Chapter 4 focuses on how to model genetic
diseases using a Bayesian network. Chapter 5 discusses limitations of the current
GenInfer+ system and directions for future improvements.

1.2 Human Genetics

A gene represents a single location (locus) on a chromosome. A human has 22
pairs of autosomal chromosomes and 2 sex chromosomes, the X and Y chromosome,
which determine sex. A woman carries two X chromosomes, and a male carries an X and
a Y chromosome. Humans are diploid, meaning they carry autosomal chromosomes, and
thus genes, in pairs. Genes on the X chromosome are an obvious exception to the diploid
rule because a man has only one X chromosome and only one copy of any X located
genes. Human traits can be determined by any number of genes. Single locus traits are
the result of a single gene. Multilocal traits, such as height, result form the interaction of
several genes. In a population there may be variants on any gene, some of which will
lead to different notable characteristics in the individual. These different variants for a
single gene are called alleles. An individual’s pair of alleles (one from each
chromosome) for a trait is the individual’s genotype for that trait.

Genetics makes a distinction between an individual’s genotype and phenotype. The
phenotype describes the actual realization of a trait based on the genotype. The
relationship between the genotypes and phenotypes for a trait determine the inheritance
pattern of the trait. The most common inheritance patterns for autosomal single locus
traits with two alleles are recessive and dominant. Traits with two alleles, a₁ and a₂, have
three possible genotypes: a₁-a₁, a₁-a₂, and a₂-a₂. In the case of a dominant trait, the trait
is observable in the presence of the dominant trait's allele. In other words, if the dominant allele is $a_1$, then the dominant trait will appear for the $a_1-a_1$ and $a_1-a_2$ genotypes. Because a dominant trait can be caused by two different genotypes, the dominant phenotype is not a clear indication of an individual's genotype. In the case of a recessive trait, the recessive trait is observable only when an individual is homozygous for the recessive allele. If the recessive allele is $a_2$, then the recessive phenotype appears when the individual has the $a_2-a_2$ genotype. Unlike the dominant phenotype, a recessive phenotype is a clear indicator of an individual's genotype [15].

The inheritance of a child's genes from his or her parents follows Mendel's Law for autosomal single-locus diseases. Mendel's Law states that a child receives one allele from each parent for every pair of genes. Each allele is randomly chosen from the parent's pair for the gene. Mendel's Law is a reflection of the chromosome segregation process that occurs when each parent produces the gametes that later combine to form the child. During chromosome segregation, a cell splits into gametes which each carry one sex chromosome and one chromosome from each autosomal pair of chromosomes. Because each chromosome has one allele for each gene pair, a gamete receives one allele for each gene.

The inheritance of multilocal traits is more complicated. The same underlying mechanism, chromosome segregation, is at work in determining a gamete's alleles. However because the trait depends on several genes, the placement of genes on chromosomes is an important issue in determining inheritance. If two genes are on the same chromosome, then the gene alleles on this chromosome will be inherited together by the gamete. In some cases chromosomes exchange corresponding chromosome parts.
during segregation, an event called chromosomal crossover. If crossover occurs, then genes on the same chromosome may segregate apart. The distance between two genes on a chromosome affects the likelihood of crossover segregating the genes; two genes close together are less likely to be split apart.

1.3 Inheritable Diseases

There are three different categories of genetic diseases: aneuploid, unilocal, and multilocal.[3] Aneuploid diseases are caused by abnormal number of chromosomes. These disorders are caused by errors during chromosome segregation in the creation of gametes, and thus are not linked to the inheritance of specific genes. Unilocal diseases, such as sickle cell anemia, are caused by a single defective gene.[16] Multilocal diseases are caused by the interaction of several defective genes. Because unilocal and multilocal diseases are caused by genes, they can be passed from parents to children just like other genetic traits. This thesis is concerned with the inheritance of unilocal and multilocal disorders.

The most common source of a unilocal or multilocal disorder in an individual is inheritance of the defective genes. An affected individual often comes from a family with a history of genetic disease. The family history of an inheritable disease is often represented graphically with a family pedigree. Figure 1.1 shows the pedigree of a family infected with cystic fibrosis, a unilocal autosomal recessive disease.[16] In a pedigree, circles represent women and squares represent men. In this pedigree, John and Mary are married (represented by the horizontal line) and have a male child (represented by the vertical line connected to a square). The child is affected with cystic fibrosis,
which is symbolized by a filled square. Because cystic fibrosis is recessive and the child is affected, we can infer that the child is homozygous for the recessive cystic fibrosis allele. Because neither parent is affected and yet passed the cystic fibrosis allele to the child, we can infer that they are both carriers of the disease allele.

Diseases differ not only in the number and location of genes but also in expressivity. In the example of cystic fibrosis, the disease was only expressed when an individual was homozygous for the cystic fibrosis allele. In many diseases, genotypes do not often guarantee a phenotype. Individuals may respond differently to the same genotype. For example, two women may both inherit the BRCA1 gene, which increases the risk of breast cancer, but only one may ever experience breast cancer.[7] In another example of complex expressivity, symptoms for the Huntington disease do not appear until late in life.[16] The expressivity of a disease complicates the conjecture of an individual’s genotype from his phenotype. In the cystic fibrosis example, we inferred that the parents were carriers of the disease because they were not affected. However, if
cystic fibrosis has variable expressivity, then we can not make this assumption because the parents could be homozygous for the defective allele and yet still be unaffected.

The allele mutation rate is also important in characterizing a disease. Mutation of normal genes can introduce a disease into a family with no history of a genetic disorder. Once a defective allele has been created, it can be passed from parents to offspring. Like expressivity, mutations complicate the inference of parent genotypes for an affected child. In the cystic fibrosis example, we assumed that the parents were carriers for the defective allele because the child was affected. However, one parent could have been homozygous for the normal allele and yet passed a defective allele to his or her child through a mutation during chromosome segregation. Hemophilia is an example of a disease where mutation is a common source of the genetic disorder in a family.

Diseases can also differ in their incidence in the population. The frequency of a disease is often highly dependent on the ethnicity or geography of the population. Cystic fibrosis, for example, occurs in 1/2000 in some Caucasian populations and is very rare in Asians. Another single-locus disorder, sickle cell anemia, is very common in equatorial Africa and occurs in 1/400 of American Blacks. Understanding the incident rate of a disease in the members of a pedigree is important in understanding the source of defective genes.

1.4 Genetic Counseling

Genetic counseling is a process of communication, the intent of which is to provide individuals and families having a genetic disease or at risk of such a disease with information about their condition, to explore the
personal consequences of this information, and to provide information that would allow couples at risk to make informed reproductive decisions.[1]

In order to help patients make an informed reproductive decision, a genetic counselor must be able to use all existing information about a patient’s family to make accurate estimates of the genetic disease risk for unborn children. Genetic counselors use two techniques in calculating genetic risk: pedigree analysis and direct genetic testing.

In pedigree analysis, disease information and family history are used to create probabilistic estimates of the parents’ disease genotypes. This information is used to predict disease genotypes and risk for unborn children. Calculating probabilistic estimates of parental genotypes is a complicated exercise using Bayes’ Theorem. Age-dependant expressivity and race specific disease frequency information are included in the calculations. For a genetic counselor, performing accurate estimates in a large, complex pedigree is a very difficult task, and often counselors are forced to make only rough estimates.

Pedigree analysis results should be supplemented with information from genetic screening tests. Many tests exists which provide evidence about the genotype of an individual. For example, Tay-Sachs is autosomal recessive disorder caused by three different mutations.[1] Because of the limited number of alleles, effective tests exist for detecting a carrier for Tay-Sachs and eliminating uncertainty about a person’s genotype. By performing genetic screening on living family members, one can calculate better probabilistic estimates of individual genotypes than could be calculated using just the inheritance pattern of the disease in the pedigree.
Combining all available information from a pedigree and genetic testing is a daunting and error prone task for a genetic counselor. Information from the pedigree and tests interact in complicated and sometimes unintuitive ways. Therefore, there is a strong need for a genetic risk system that allows genetic counselors to combine all information about a disease, from the pedigree and from outside tests, to get accurate risk estimates for unborn children. A easy-to-use system would encourage genetic counselors to seek out more information which could be used to provide better risk estimates. The goal of this thesis is to design a genetic risk system which satisfies these needs of a genetic counselor.
Chapter 2
Creating a genetic risk system

Several projects have attempted with varying degrees of success to create genetic systems to perform pedigree analysis. The first section of this chapter discuss some of these earlier projects. The second section focuses on the GenInferII program which was the most highly developed system created explicitly for genetic counselors. The third section describes some of the requirements an ideal genetic counseling risk system must include in order to be successful.

2.1 Previous programs dealing with inheritable diseases

Past genetic systems can be divided into two classes: systems created by geneticists for genetic linkage analysis, and systems created by artificial intelligence researchers for experimentation in probabilistic reasoning. Geneticists and artificial intelligence researchers have each developed their own techniques for analyzing
information in pedigrees. I will examine past work by both research communities and discuss the lessons that can be learned from their work.

Geneticists have been concerned with the problem of determining the genetic characteristics of a disease based on evidence contained in pedigrees. Over the past decades, geneticists have developed very sophisticated systems which efficiently determine the most likely genetic explanation for the inheritance pattern of a disease. These systems have become critical in work done for the Human Genome project, which is attempting to map all human genes. One sophisticated linkage analysis program is the MAPMAKER system developed by Lander.[6] This system takes large batches of data describing family pedigrees with a certain trait and efficiently determines the genes that best explain the inheritance patterns observed in the pedigrees. MAPMAKER and other genetic linkage systems have broken new ground in likelihood estimation for genetic traits. However, because these systems are not designed to perform genotype and phenotype inferences on a single pedigree for a given disease model, they are of limited value to a genetic counselor.

Several genetic analysis systems have been developed in the field of artificial intelligence. In contrast to genetic linkage projects, these systems were designed to perform inferences on the genotypes and phenotypes of individuals in a pedigree for a given disease model. However, these systems were not specifically concerned with solving the pedigree analysis problem for genetic counselors. Instead, these systems used genetic modeling as an opportunity to experiment with different probabilistic reasoning techniques. The majority of these systems experimented with solving Bayesian networks, a problem which is NP-hard.[3] While these early programs provided some of
the functionality needed by genetic counselors, they had many limitations which prevented them from being used by genetic counselors.

One early program, PEDIG, was written in Fortran and lacked much flexibility in the type of diseases it could handle.[4] This system could perform genotype inferences for large pedigrees for autosomal and x-linked diseases with any number of alleles. Disease models allowed for different prior population probabilities of genotypes and complex genotype to phenotype mappings. However, disease models had to be entered numerically, something too difficult for most genetic counselors. PEDIG did not support pedigrees with consanguinity, a situation that occurs frequently in genetic counseling analysis.

Another program, the GENEX processor, derives probabilistic formulas instead of numerical answers for inheritance problems of qualitative traits.[5] Such results are probably of little practical value to a genetic counselor. GENEX also had a complicated interface which required pedigrees to be broken down by hand into atomic assumptions described in terms of probabilities.

Another program designed by Prokosch and his colleagues analyzed pedigrees described using a system of rules.[9] The system used a forward-chaining expert shell to examine the pedigree and make genotype inferences. The goal of the system was to design a portable, inexpensive system that genetic counselors could use on desktop computers. However, the forward-chaining process performed too slowly for much practical value, suggesting that a procedural algorithm would be necessary. It is also not clear if the system has capable of analyzing pedigrees with consanguinity, a situation that seems impossible to analyze using a forward-chaining system. To analyze a pedigree, the
system required the user to enter expert system rules, making it impractical for a genetic counselor.

Spiegelhalter explored a theoretical genetic risk system that used probabilistic belief networks to model the inheritance of diseases within a pedigree.[10] He suggested that Bayesian networks had the functionality necessary to probabilistically calculate the risk to individuals in a pedigree. Although Spiegelhalter did not create a working system, his Bayesian network methods would support consanguinity and diseases with complex penetrance and population frequencies.

2.2 GenInferII

The most promising genetic risk system has been developed through the ongoing GenInfer project, based on initial work by Harris.[3][14] The most recent system developed from this project is the GenInferII program by Szolovits. GenInferII is very successful in using Bayesian belief networks to capture much of the functionality needed by genetic counselors in pedigree analysis. The complexity of the underlying models does not appear in the user interface, which is very friendly and written in the language of genetic counseling.

GenInferII uses a Bayesian network to model the relationships between family members. This system allows the user to create sophisticated genetic disorders and quickly measure risk factors for a patient based on his family history. The latest version of this program is limited to single-locus disorders with an inheritance pattern that is either autosomal recessive, autosomal dominant, X-linked dominant, or X-linked recessive.[13] As a result of this limitation, the program only supports two alleles ("bad"
and "normal") at a single locus and only two phenotypes ("affected" and "normal"). GenInferII allows for fairly sophisticated diseases, with age-dependent penetrance and ethnic background specific genotype frequencies. Support exists for pedigrees with consanguinity but not for multiple births (identical twins).

GenInferII uses a graphical user interface in all interactions with the user. Using the mouse and a pedigree toolbar, it is very easy to draw and manipulate sophisticated pedigrees. Simple dialog boxes are used to create diseases. It is straightforward for a user to create a new disease with any supported inheritance pattern, penetrance, and population genotype frequency, with support for age-dependent expressivity and ethnic background genotype frequencies. Dialog boxes are also used to enter information about an individual in the pedigree, such as name, age, race, and phenotype. A strong feature of the interface is that it is intuitive and is expressed in the genetic language of genetic counselors. There is no mention in the interface of the underlying techniques that are used to arrive at the risk estimates. GenInferII can therefore be used without any expert knowledge of belief networks. Its independence from the belief networks allows for changes and experimentation in the probabilistic system without an impact on the user interface.

GenInferII however had some limitations which motivated the research in this thesis. As the program was improved, it became too complex to manage. It was not properly designed to allow for further experimentation with the underlining probabilistic model or for future improvements in its functionality for genetic counselors. Besides lacking extensibility, the system lacked portability. Because GenInferII was written in
Lisp for the Macintosh, it can not run on the majority of computer systems commonly used by genetic counselors.

There are additional problems with the functionality in GenInferII which need improvement. Disease model support must be extended beyond single-locus, two allele diseases. GenInferII also does not support the use of outside evidence, such as genetic screening tests, to be incorporated into the knowledge of the pedigree.

Despite these limitations, GenInferII successfully demonstrated the use of Bayesian networks to solve genetic risk problems. The GenInfer+ genetic risk system relies heavily on results obtained from the GenInfer project.

2.3 The elements of an ideal solution

The strengths and weakness of past projects provide guidelines for what is necessary for a successful genetic counseling system. Building upon the foundation of the GenInfer project, I have created a list of requirements for a genetic disease system to satisfy the needs of artificial intelligence researchers and genetic counselors.

2.3.1 Design requirements

- The system must be portable and inexpensive, so that it can be run on the various desktop computers used by genetic counselors.

- The system should be designed using an object-orientated language and composed of independent modules. For example, the user interface should be independent of the expert system used to model inheritable diseases. Separation of the system into independent components would allow for experimentation in intelligence algorithms.
without changes propagating into other modules. System modules could be improved in parallel because each component is independent of the implementation of other components.

2.3.2 Functionality requirements

- The system must support complex pedigrees, including support for unborn children of unknown sex, consanguinity, and multiple births.

- The system must support complex disease models. It should allow for multiple alleles at a single location and multilocal disorders. The system should support mutation rates for each locus and crossover rates for multilocal disorders. Additional diseases parameters such as age-dependent expressivity and ethnic-dependent population incidence should be included.

- The system should support multiple independent phenotypical traits. For example, cystic fibrosis produces lung and kidney symptoms independently in affected individuals. A disease model for cystic fibrosis should support observations on both lung and kidney conditions, each with their own phenotypical values, such as “affected” and “normal.”

- Information not contained directly in the pedigree, such as genetic screening results, should be included in genotype inferences. The user should be able to create tests for different genotypes which can be performed on individuals in the pedigree. These tests should support information such as false positive and negative rates.

- The system should be intelligent enough to recognize anomalies in the pedigree data which seem unlikely (or impossible) given the disease model. Such anomalies may
be the result of patient misinformation or yet undiscovered disease factors for the
disease being modeled.

2.3.3 User interface requirements
- The user interface should be written in the language of genetic counseling. The
  system can be easily operated by a genetic counselor and requires no knowledge
  about the expert system used to model pedigrees and diseases.
- The system should use a graphical user interface, complete with mouse support, for
  the creation of pedigrees.
- The system should allow the user to easily create simple diseases, perhaps through the
  use of dialog boxes. It should allow advanced users to access all possible features in
  disease models to create complex disease descriptions.

The functionality and the user interface requirements are difficult to satisfy together.
As more functionality is added, the user interface becomes more complex to support the
additional functionality. The complexity required in a disease description can make
disease input an extremely difficult task. It is important to realize that more often than
not, genetic counselors will not be creating new disease models, but instead will be
relying on existing disease models in the creation of new pedigrees. For a system to be
useful for genetic counselors, there should exists a library of up-to-date disease models
that they can accessed when creating patient pedigrees. In the ideal situation, a few
expert genetic counselors or researchers would create disease models (at a disease model
authority) and then share these models with genetic counselors. Genetic counselors
would be able to retrieve the most accurate disease models from the authority and would not be required to understand the specifics of the disease or the steps necessary to create a disease model.
Chapter 3
GenInfer+

In this chapter, I will discuss the GenInfer+ genetic counseling system, which is based on the guidelines presented in Chapter 2 for creating a successful genetic counseling system. These guidelines covered three different aspects of the genetic system: design, functionality, and user-interface. The design requirements describe the methods that should be used in structuring and implementing the system, so that it can be used by genetic counselors and by artificial intelligence researchers. Functionality and user interface issues deal with requirements needed to satisfy the needs of genetic counselors. I will discuss GenInfer+ and its adherence to these requirement areas.

3.1 Design

GenInfer+ was designed with two goals in mind. The first was to create a portable system that could be used on the desktop computers of genetic counselors. The
second was to separate implementation aspects of the system into distinct modules, allowing for future changes to be isolated to one module.

The first step in the design was choosing a software language which would support the design requirements. After considering the platform and functionality requirements for this system, the Java programming language was chosen. One primary reason for using Java was its portability; a Java based program can be run on any system that has a Java virtual machine. Because of the ubiquity of web browsers on desktop computers, all genetic counselors should be able to run a Java based genetic counseling system, without purchasing a new system or buying additional software. This aspect of Java eliminated a barrier to entry into the genetic counseling office.

In addition to its portability, Java’s natural relationship to network computing provides added benefits to its use. If the genetic counseling system is created as an applet, it can be centrally stored and downloaded on demand to genetic counselors. This would allow genetic counselors access to the latest version of the system as changes are made and new features are added. The applet design of GenInfer+ also encourages genetic counselors to import new disease and genetic screening models from the Internet. The applet design would allow genetic counselors to easily provide the most state of the art advice to their patients.

Java is an object-orientated language, which naturally encourages the separation of a software project into independent components. To create a modular system, I split the different implementation issues of the software into different classes. Once a well defined interface is established for each class, these modules can then interact through the interfaces without any concern with the underlying implementation of the classes. As
long as the interfaces remain unchanged, the implementation of the system can be improved without changes leaving the affected module. However, separating the system into modules is not enough to minimize future changes to the system. When a new feature is added, several class interfaces will probably be changed. This change will affect all objects that interact through the old class interfaces. To limit the impact of future improvements on the system, I have attempted to limit the interface dependencies between classes. Figure 3.1 shows the major GenInfer+ classes and their interdependencies. I will discuss how implementation issues are isolated to each module and how future improvements can be easily made due to this modularity.

**Figure 3.1 – Dependency diagram of major GenInfer+ classes**

The Bayes Network and the Bayes Network Node classes work together to describe a very general Bayesian network interface which can be used to model inheritable diseases and genotype screening tests. Appendix A.1 and A.2 describes the
interface to the *Bayes Network* and the *Bayes Network Node* classes. The interface allows for simple manipulation of the network, updating observations on nodes, and the performance of inferences. Creating a Bayesian network from scratch was beyond the scope of this project. Fortunately, a freely available GNU Java based Bayesian network has been created by Fabio Cozman at Carnegie Mellon University. By wrapping Cozman’s Bayesian network in the generic *Bayes Network* class and the *Bayes Network Node* class interfaces, I was able to use his Bayesian implementation while hiding the implementation details from the other modules in the system.

GenInfer+’s use of the Cozman Bayesian network implementation demonstrates the versatility and power of the Java class interface in the design of complex systems. Despite Cozman’s different Bayesian interface, it was fairly straightforward to create a *Bayes Network* class which used Cozman’s network. In fact, any type of Bayesian network can be inserted into the GenInfer+ system if it adheres to the *Bayes Network* interface, and the system will continue functioning without any additional changes. The separation between the GenInfer+ system and its expert system through the *Bayes Network* interface allows artificial intelligence researchers to experiment easily with different Bayesian network implementations.

The *Pedigree* class maintains the family relationship structures of the program. However, the *Pedigree* class implementation does not depend on how diseases are represented in the Bayesian network. When creating family relationships, the *Pedigree* object uses the *Disease* class to make the probabilistic relationships in the Bayesian network on its behalf. Using the *Disease* functions described in Appendix A.3, a *Pedigree* can model all family relationships in the Bayesian network without any
knowledge of the disease structure. Because all disease use the Disease interface, a Pedigree uses the same functions to implement a 5 locus, 10 gene, 20 allele disease in the Bayesian network as a single locus, two allele disease. All information about diseases is isolated to the Disease class, which allows for additional disease types to be added to the GenInfer+ system without changing other modules in the system. Just like the Disease class, the Genetic Screen Descriptor (GSD) class use an interface (described in Appendix A.4) to create Genetic Screens on behalf of the Pedigree, thereby isolating the Pedigree from the probabilistic modeling of the genetic screens in the Bayesian network.

The Person class plays a critical role in the interaction between the Pedigree, Disease, and Genetic Screen classes. Person acts as a go-between for these three classes, storing Bayesian nodes for the phenotype, genotypes, and genetic screens, as well as pedigree information such as marriage and children relationships. While the Person object allows the system to isolate the Pedigree, Disease, and Genetic Screen class interfaces, it also makes the Person class’s interface and functionality strongly dependent on all these classes. Any change to one of these three classes, such as using additional Bayesian nodes for disease or genetic screen modeling, would require Person to be changed as well.

The Pedigree Canvas class creates the user interface and responds to user input with the Pedigree class. It has no knowledge of the implementation of the pedigree or any of the probabilistic modeling occurring in the system. Because the Pedigree Canvas class only interacts with the Pedigree and Person class interfaces, it is completely independent of the other major class interfaces. Once the underlying models have been
created, the user interface can be improved for genetic counselors without requiring any changes to other modules in the system.

Saving and loading class objects to files posed another dependency challenge in the design. The user interface and other modules should not be dependent on the file representation of other objects. To eliminate this potential dependency, each class is responsible for saving and loading its own representation from a string. Loading and saving a session of the system is done by asking each object to load or save its representation from the file.

The design of the GenInfer+ system requires that new implementations of a class conform to the established class interface. If new implementations do not maintain the interface, then the modular independence of class implementations will break down. To help enforce the interface requirement, the GenInfer+ design includes testing modules for several classes. For example, the Bayes Network class testing module verifies that the Bayesian network performs as specified in the interface. Developers can use the testing modules to see if their new class implementation meets the interface specifications. If the test passes, then the system should continue functioning properly with this new implementation.

3.2 Functionality

GenInfer+ supports a wider range of functionality than any past project in genetic counseling. The majority of the functionality requirements described in Chapter 2 are satisfied. Any additional features to improve the system’s overall functionality can be added fairly easily thanks to the powerful design.
GenInfer+ offers support for complex pedigrees, including support for unborn children and consanguinity. Unborn children are of unknown sex and therefore can have a wide range of genotypes in the case of sex linked diseases. The system allows the user to make genotype and phenotype observations, and then perform inferences on the phenotypes and genotypes of all individuals. The pedigree does not currently offer support for multiple births, although this functionality is possible with the current system and is discussed in Chapter 4.

Genetic screening tests can be designed and performed on pedigree members. Genetic screens are designed for a specific disease because the screens are genotype specific. They allow a genetic counselor to make further inferences on an individual's genotypes based on test results. Genetic screens are probabilistically conditioned on genotypes and therefore support false positive and negative rates. A pedigree can include multiple types of tests and multiple tests can be conducted on a single individual.

GenInfer+ offers support for fairly complex disease models. Current disease models support multiple allele single locus gene disorders. There is not yet support for multilocal disorders, though this disease type is possible in the current system and is discussed in Chapter 4. Because the majority of diseases analyzed by genetic counselors are single locus, this limitation is acceptable in most cases.[16] Diseases currently have only one phenotype, which can be either “Affected” or “Unaffected.” While multiple phenotypes are possible, they are not currently supported because of the limited number of disease where this information is useful. The system does support mutation rates from the normal allele to the mutated alleles. Like the GenInferII system, it does not support reverse mutation from a mutated gene back to a normal gene because of the rarity of this
Disease models also support age-dependent expressivity and ethnic dependent population genotype frequencies. Genotype frequencies can be determined using the Hardy-Weinberg equilibrium from the disease frequency.

GenInfer+ is also capable of determining inconsistencies and anomalies in the information contained in a pedigree. For example, if an individual is known to have two bad genes for a recessive disease, and yet is unaffected by the disease (this should be impossible assuming one hundred percent penetrance), GenInfer+ will inform the user of a mistaken observation somewhere in the pedigree. This feature is important in catching user input mistakes and discovering unknown features of a disease.

### 3.3 User Interface

GenInfer+ uses a graphical interface in all interactions with the user. All the program's functionality is accessible through menus. A drawing toolbar is used for the creation and manipulation of pedigrees and genetic screens. Dialog boxes are used to create new types of diseases and genetic screens. The goal of the interface is to make it easy and intuitive for genetic counselors to use the system. The interface is expressed solely in the language of genetic counseling, with no mention of the underlying implementation of the system.

The user interface and functionality of GenInfer+ is best shown through an example interaction with the system. In this demonstration, GenInfer+ will be used to create a model for a single locus two allele recessive disorder.

To begin, we must first create a model for this disease. We can create the disease through the Disease Dialog box, shown in Figure 3.2. The Disease Dialog box makes it
easy to create simple diseases, but also offers access to complex disease features. Our simple disease can be created using the simple disease option, with a normal allele $A$ and a defective allele is $a$. We can specify the mutation rate and the disease frequency in the population. The Hardy-Weinberg equilibrium will be used to calculate the genotype frequencies.

Once we have created a disease model, we can begin creating a family pedigree for the disease. A blank pedigree is created using a menu option and specifying a disease model to use in the pedigree. Once a blank pedigree is created, the pedigree drawing toolbar can be used to create family members and family relationships. Figure 3.3 shows a three generation family, with two individuals affected with the recessive disorder.

People can modified using the Person Dialog box. This dialog box allows us to make observations on an individual’s phenotype or genotype, as well as change personal information such as the individual’s name, race, and age. The Person Dialog box also
Figure 3.3 – Pedigree created using GenInfer+

Figure 3.4 – Person Dialog box for Bob

includes the latest inference information about this individual's genotype and phenotype. Figure 3.4 shows the Person Dialog box for Bob.

Genetic screens can be created using the Genetic Screen Dialog box. When creating a genetic screen, we must specify a disease for the genetic screen to test. Once a genetic screen has been created, it can be applied to any pedigree with this disease. Using the toolbar, any family member in the pedigree can be tested. By observing the test results of the genetic screen, we can improve our genotype inferences on all individuals in the pedigree.
As this example demonstrates, new diseases and pedigrees can be easily created using the GenInfer+ interface. The system allows for the simple addition of genetic screens and quick inferences on genotype and phenotype risks for all individuals in the pedigree.
Past projects in pedigree analysis for genetic counseling have generally used Bayesian belief networks to model the inheritance patterns of genetic diseases. Bayesian belief networks are excellent for modeling probabilistic events where some events influence and cause other events. Discrete belief networks are therefore a natural representation for inheritable diseases, where parental genotypes cause a child's genotype based on the probabilistic properties of chromosome segregation. The causality relationship extends beyond parental and child genotypes, as an individual’s genotype causes his or her phenotype based on the relationship between genotypes and their expressivity. The flexibility of Bayesian belief networks also allows for additional information, such as genetic screening test results, to be incorporate into the phenotype information about a family described by the network. Once a Bayesian network has been
created that contains all information about the pedigree, simple inferences can be used to make predictions about the phenotypes of unborn children or the genotypes of family members.

4.1 Single locus two allele diseases

The simplest inheritable disease that can be modeled using a Bayesian belief network is a single locus autosomal two allele diseases with 100% penetrance for a phenotype which is either present ("affected") or not present ("normal"). A disease of this form can be either dominant or recessive with regard to the affected phenotype. The modeling of such a simple disease is fairly straightforward using a standard Bayesian belief network and lays a foundation for more complicated disease modeling.

Modeling a disease in a Bayesian belief network is best done through an example. I will describe the general theory used to model simple diseases and then demonstrate these concepts through the use of an imaginative disease named Red Hook. This inheritable disease makes affected individuals become ocean sailing pirates that spend their lives attacking merchant ships and burying stolen treasure on deserted islands. Red Hook is a simple recessive disease due to an autosomal gene. This gene has the normal allele $P$, and the recessive Red Hook allele $p$. Because this disease is recessive, Red Hook only occurs in individuals with the $pp$ genotype.

For a simple inheritable disease, two belief network node are used to represent a individual: one node for the genotype and one node for the phenotype. The genotype node can take on values representing the different possible genotypes, and the phenotype node has the value either “affected” or “normal” to represent the possible phenotypic
conditions. Because an individual’s phenotype depends on his genotype, an individual’s genotype node will have a directed arc to his phenotype node. Using a network for a single individual, probabilistic estimates for his genotype can be made using the individual’s observed phenotype. For Red Hook, an individual’s genotype node has three possible values: PP, Pp, and pp. The phenotype node is either affected or normal. The arc table in Figure 4.1 describes the probabilistic relationship in the arc between the genotype node and the phenotype node for Red Hook: individuals with the genotypes PP or Pp are phenotypically normal 100% of the time, and individuals with the pp genotypes are affected 100% of the time.

![Arc table for Red Hook disease](image)

**Figure 4.1 - Incomplete Bayesian model of an individual for Red Hook disease**

Inferences on an individual in isolation can not be performed until a probabilistic distribution is specified for his possible genotypes. The distribution for a disease’s genotypes is typically determined through the use of the Hardy-Weinberg equilibrium. The Hardy-Weinberg equilibrium [15] determines the genotype probabilities based on the frequencies of the disease in the population. Let’s assume that for Red Hook, the genotype frequencies based on a Hardy-Weinberg equilibrium are 1/100, 19/100, and
80/100 for \( pp \), \( Pp \), and \( PP \) respectively. This information allows us to complete the Bayesian network representation of an individual as shown in Figure 4.2.

<table>
<thead>
<tr>
<th>Genotype</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>0.80</td>
</tr>
<tr>
<td>Pp</td>
<td>0.19</td>
</tr>
<tr>
<td>pp</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Affected</th>
<th>Unaffected</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Pp</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pp</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4.2 – Complete Bayesian model on an individual for Red Hook disease

The relationship between parents and children is modeled by creating arcs from parent genotype nodes to children genotype nodes. Every child will have two arcs affecting its genotype node, one from the mother’s genotype node and one from the father’s genotype node. The probabilistic relationships in these parental arcs describe the gene segregation process for the disease. For a given parental genotype combination, a distribution is described for the child’s possible genotypes. Using Red Hook as an example of simple two allele gene segregation and inheritance, if both parents have the genotype \( Pp \), then the child has a .25 chance of \( PP \), a .50 chance of \( Pp \), and a .25 chance of \( pp \) for his genotype. The complete parental relationship table is shown on the next page in Figure 4.3.

Complex pedigrees can be created using the simple two node individual model and parental relationship arcs. This model has no limitation on the number of family generations or the relationships within the family (consanguinity can be included). While the Red Hook example demonstrated a Bayesian network for a recessive disease, this
model could also simulate a dominant disease by just changing the genotype to phenotype arc relationships.

![Figure 4.3 - Bayesian model of parental genotype relationships for Red Hook](image)

**4.2 Multiple Alleles**

The Bayesian network model used to simulate the inheritance of a single locus two allele disease can be extended to model a single locus multiple allele disease. The pedigree model for a multiple allele disease would still have a phenotype and genotype node for each individual, and parental arcs between parent genotype nodes and children genotype nodes. To model the multiple alleles, each genotype node will have more possible values to represent the additional genotypes. The parental arc relationships will have to be updated to assign genotype values to children based on the parental genotypes.

For a single locus disease with \( n \) different alleles, there are \( \frac{n^2 + n}{2} \) possible genotypes. This representation results in a \( O(n^4) \) sized table to represent the parental arc relationship between parental and child genotypes.
Using the Red Hook example, imagine that a second abnormal allele is discovered which causes the Red Hook disorder. This new allele, named $s$, causes the disorder if it occurs with the other abnormal allele $p$. In other words, this modified Red Hook disease will produce the disorder for the $pp$ and $ps$ genotypes. Tables 4.1 and 4.2 show the new parental genotype arc and genotype-phenotype relationship tables for this version of the disease. The genotype population distributions must also be updated for genotype nodes with no parents.

<table>
<thead>
<tr>
<th>PP</th>
<th>Pp</th>
<th>Ps</th>
<th>ps</th>
<th>pp</th>
<th>ss</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pp$-$pp$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$PP$-$pp$</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$PP$-$Ps$</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$PP$-$ps$</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$PP$-$pp$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$PP$-$ss$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Pp$-$Pp$</td>
<td>0.25</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>$Pp$-$Ps$</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>$Pp$-$pp$</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>$Pp$-$ss$</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Ps$-$Ps$</td>
<td>0.25</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Ps$-$ps$</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>$Ps$-$pp$</td>
<td>0</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>$Ps$-$ss$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>$Ps$-$pp$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>$Ps$-$ss$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>$Ps$-$pp$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$Ps$-$ss$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$ss$-$ss$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.1 - Parental arc table for three allele Red Hook

$|\begin{array}{c|c|c}
\text{Affected} & \text{Unaffected} \\
\hline
PP & 0 & 1 \\
Ps & 0 & 1 \\
ps & 1 & 0 \\
pp & 1 & 0 \\
ss & 0 & 1 \\
\end{array}$

Table 4.2 - Phenotype to genotype table for three allele Red Hook

4.3 Expressivity

Some genetic disorders do not appear in all individuals with the affected genotype. Other disorders do not make their presence known until the individual has reached a certain age. Different phenotype expressivity relationships can be modeled
using the Bayesian networks described in the two previous sections. Expressivity is modeled in the genotype-phenotype arc for each individual. In cases where a genotype produces a disorder with only a certain probability, this expressivity changes the phenotype probabilities conditioned on this genotype value. In the case of age-dependent expressivity, every individual in a different age range is represented with a different genotype-phenotype table which describes the phenotype expression for a given genotype.

Consider the original Red Hook disease in section 4.1, which had a normal and mutated allele (\(P\) and \(p\) respectively). Consider a new version of this disease which has age dependent expressivity of the pirate disorder for individuals with the \(pp\) genotype. Individuals with the \(pp\) genotypes have a .50 chance of showing symptoms of this disorder from birth. If a \(pp\) genotype person is not affected at birth, he will become affected at the age of 20. Because there are two age groups (before 20 years and after 20 years of age), there are two different genotype-phenotype tables (see Tables 4.3 and 4.4). The table used for an individual would depend on his age.

<table>
<thead>
<tr>
<th>Genotype</th>
<th>Affected</th>
<th>Unaffected</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Pp</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pp</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.3 – Genotype to phenotype table for individuals under 20 years of age for Red Hook

<table>
<thead>
<tr>
<th>Genotype</th>
<th>Affected</th>
<th>Unaffected</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Pp</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pp</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.4 – Genotype to phenotype table for individuals over 20 years of age for Red Hook
4.4 Mutation

Spontaneous mutation can introduce a genetic disorder into a family which had no previous carriers for the disease. A mutation results in a change from the normal allele to a defective allele. Mutations can be represented in the previously described Bayesian network models by changing the parental genotype-genotype arc relationship tables. The new table will have positive conditional probabilities for entries that had previously been zero, because a mutated allele can always be introduced into the child genotype regardless of the parental genotypes. While mutation rates are useful for explaining unusual situations in pedigrees, they cause some computational problems in pedigrees where normal inheritance is the cause of the disorder.[3] The problem is that nearly any observation, no matter how unlikely, is possible for a disease with a mutation rate. This prevents the genetic counseling system from discovering errors in the input of the pedigree, because nearly all inputs are possible.

4.5 Multiple Phenotypes

Some disorders appear in individuals through multiple and independent symptoms. For example, cystic fibrosis causes both lung and kidney conditions. Multiple phenotypes are represented in a Bayesian network by adding additional individual phenotype nodes for each independent symptom of the disorder. The individual’s genotypes node has an arc to each of its phenotype nodes. Each genotype-phenotype arc table describes that phenotype’s conditional probabilities for different genotype values.
Consider a modified Red Hook disease which has two independent phenotype symptoms: a missing right leg and a missing right eye. Each symptom occurs independently with a probability of .5 given the pp genotype. To model this new disease, each individual will have a leg phenotype node and an eye phenotype node. An individual’s genotype node is connected to each of these phenotype nodes with an arc. Figure 4.4 shows a pedigree Bayesian network for this new disease.

![Bayesian Model for an individual with multiple phenotypes for Red Hook](image)

**Figure 4.4** – Bayesian Model for an individual with multiple phenotypes for Red Hook

### 4.6 Multilocale Diseases

Some diseases are caused by defects at multiple genes. If these genes occur on the same chromosome, then the gene alleles will be inherited together when crossover does not occur. If crossover does occur, then alleles of each chromosome may be inherited by children. The probability of crossover depends on the distance between genes on the same chromosome. The inheritance of genes on one chromosome are independent of the inheritance of other chromosome genes.[5]

While one genotype node could be used to represent all the alleles for all genes, the table necessary for the parental genotype arcs would be extremely large. Using the fact that chromosome genes are independent of the segregation of other chromosome genes, this model can be improved. A better Bayesian network model for a multilocale
disease represents an individual using one genotype node for each chromosome pair. Each genotype node would represent the gene alleles for all genes on each chromosome pair. All the chromosome genotype nodes are connected to the individual’s phenotype node. Each parental genotype node will have an arc to the child genotype node for this chromosome pair, so that each of the child’s genotype node will only depend on the parent’s two chromosome genotype nodes and not all the other genotype nodes. Parental arcs will have very complicated tables, which describe the inheritance of multiple genes and include the probability of crossover of alleles for each chromosome of the pair. Figure 4.5 shows a Bayesian model for a disease with genes on two chromosome pairs.

![Bayesian model for a multilocal disease with two chromosomes](image)

**Figure 4.5 - Bayesian model for a multilocal disease with two chromosomes**

### 4.7 Genetic Screening

Genetic counselors often try to use additional information outside the pedigree to infer the genotypes of family members. Genetic screening tests is one such technique where the individual’s genotype affects his test result. Because of the dependency of test
results on genotype, genetic screens fit well into the Bayesian belief network model. A genetic screen is represented by an additional genetic screen node which has values representing the test’s results. An additional arc is added from the individual’s genotype node to his genetic screen node to represent the screening test’s dependency. The arc’s probability table will include information about the genetic screen such as false positive and negative rates. As additional tests are performed, new genetic screen nodes are added to the individual through connections to the genotype node.

Figure 4.6 shows a Bayesian network model for an individual with a genetic screen for the original Red Hook disease described in section 4.1. This genetic screen tests positive for someone with a $p$ allele and negative for someone with the $PP$ genotype. There is a false positive and false negative rate of 5% for this test, which appear in the genotype to genetic screen arc table. This test could be used by a genetic counselor to test individuals to see if they are carriers for the defective $p$ allele.

![Figure 4.6](image)

**Figure 4.6 – Bayesian model of a genetic screen for Red Hook**

### 4.8 Multiple Births

Multiple births, such as identical twins, have additional genotype dependencies which are not represented in the standard Bayesian network model used thus far to
describe diseases. In the standard model, children genotypes are decided independently from their parent genotype values. Identical twins, on the other hand, have the same genotype. Twins can provide valuable information about a pedigree for a disease with complex expressivity. If two twins carry the affected genotype for a disease with a low penetrance, there will be a higher likelihood that the disorder will appear because the phenotypical traits of the twins are decided independently from the same affected genotype.

To represent twins in a Bayesian network, twins would share the same genotype node. Each twin has his own phenotype node which is connected to the shared genotype node. The arc relationships between the nodes are the same as in the standard Bayesian network model. This model forces the twins to have the same genotype but still allows for the twins to have phenotypes chosen independently given their shared genotype. Figure 4.7 shows a twin relationship for a simple genetic disease.

Figure 4.7 – Bayesian model of two twins with a shared genotype node
Chapter 5
Future Directions

5.1 Current Limitations

While satisfying the design requirements, the GenInfer+ system does not fully meet the functionality expectations for a successful genetic counseling system. For example, the pedigree does not support multiple births, and there is no support for diseases with multiple genes and phenotypes. The steps necessary to remove these limitations vary widely depending on the feature. Using the concepts in Chapter 4, it would be fairly straightforward to add support for multiple births and multiple phenotypes. While Chapter 4 suggest methods to model a multiple locus disease, these techniques may be impractical due to the size and complexity of the arc relationships in the Bayesian network. Alternative methods for modeling disease, using phased haplotypes and elimination of impossible genotypes [6], should be pursued which can simplify the complexity of the model. Efforts should also be made to examine the
techniques used by genetic linkage programs like MAPMAKER to model multilocal diseases. Perhaps these gene mapping techniques can be used to create efficient multilocal models for pedigree analysis.

Despite its functionality limitations, GenInfer+ is a practical tool which can greatly aid genetic counselors in pedigree analysis of risk factors. The simple user interface will encourage genetic counselors to incorporate more information into a pedigree analysis than would be traditionally possible using hand analysis. Because most diseases are unilocal, the lack of support for multilocal diseases should not limit the program usefulness. In fact, even if the program did support multilocal diseases, it is not clear that enough is known about such diseases to create a reliable disease model.[5]

A remaining obstacle to the acceptance of the GenInfer+ by genetic counselors is the lack of existing disease models. Many genetic counselors may not feel comfortable creating disease models, due to either intimidation with the program or a lack of knowledge about the disease. A library of updated disease models should be made available over the Internet which allows genetic counselors to find and use the most accurately known model for common diseases.

Another area that has not been explored by this work is modeling nontraditional disease transmission from parent to child, such as mitochondrial inheritance, imprinting, and phenomena due to the expansion of trinucleotide repeats. As more diseases are discover which use nontraditional inheritance, it will become more important to find ways to model these types of diseases using Bayesian networks. Specialized modeling techniques may be required that do not involve Bayesian networks.
As research continues into genetic diseases, it is likely that more forms of inheritance will be discovered. Is there a way to allow for the creation of new inheritance models in the system that are not yet anticipated? The system design of GenInfer+ may break down if the current modular design prevents the modeling of undiscovered inheritance patterns.

5.2 Conclusion
To the best of my knowledge, GenInfer+ is the most advanced pedigree analysis tool ever designed for genetic counselors. It adheres closely to the design, functionality, and user interface requirements described in this thesis for the ideal genetic counseling system. However, as the field of genetic counseling continues to change, the description of the ideal genetic counseling system is likely to change with it.

Bayesian belief networks are powerful and can model very complicated genetic diseases. While the GenInfer+ system does not support all disease types, the implementation of disease models in the system could support extremely complex diseases. However, future work must be done to create more efficient models for multilocal diseases and to find ways to model nontraditional disease types.
Bibliography


Appendix A

Appendix A.1 - Bayes Network Interface

abstract class BayesNetwork {

abstract public BayesNetworkNode createNode(String[] values,
       double[] prob) throws ProbabilityException;
requires: values and prob must not be null
effects: Creates and returns a BayesNetworkNode which is added to this
BayesNetwork. The node is initialized to have the values in values[]
with initial probabilities contained in probs[], where each entry in
probs[] corresponds to the value of equal index in values[].
A ProbabilityException occurs if the size of values and probs is not
equal, if a value appears more than once in values[], or if probs[] do
not sum close to 1.0.

abstract public Enumeration getNodes();
effects: Returns an Enumeration of BayesNetworkNodes contained in the
network.

abstract public void removeNode(BayesNetworkNode node);
requires: node is not null and was created from the createNode method
of this network.
effects: Removes node and all of its arcs from the network. If node
has already be removed, nothing is done.

abstract public void createArc(BayesNetworkNode source,
       BayesNetworkNode dest) throws InvalidArcException;
requires: source and dest are not null, and were created by the
network.
effects: Creates an arc from source to dest in the network. This
operation reinitializes the probability table of the dest node, using a
uniform distribution for its values conditioned on parent values. If
either node is no longer in the network, nothing is done. If the arc
would create a loop, an InvalidArcException is thrown and the network
is unchanged.

abstract public void removeArc(BayesNetworkNode source,
       BayesNetworkNode dest);
requires: source and dest are not null, and were created by the
network.
effects: Removes the arc from source to dest if one exists. This
operation reinitializes the probability table of the dest node, using a
uniform distribution for its values conditioned on parent values. If
either node is no longer in the network, nothing is done.

abstract double[] inferenceOnNode(BayesNetworkNode node) throws
       ProbabilityException;
requires: node is not null and was created by this network.
effects: Returns a double array containing the current inference probabilities on this node, where each entry in the double array corresponds to the values from node.getValues array. If this node has been removed, then null is returned. If there are inconsistencies in the network, a ProbabilityException is thrown.

Appendix A.2 – Bayes Network Node Interface

abstract class BayesNetworkNode {

abstract public int getUniqueID();
effects: get the unique id of this node

abstract public String[] getValues();
effects: returns the values this node can take on. They are returned in the same order as they were for this node's creation

abstract public double[] getProb(BayesNetworkNode[] parentNodes,
String[] parentValues) throws ProbabilityException,
InvalidValueException;
effects: Returns the probs of each values (the order of probs correspond to the order of values in getValues()) when the parent nodes have the parentValues (where each parent value corresponds with the parent with the same array position in parentNodes). If this node has no parents, parentNodes and parentValues should be set to null. A ProbabilityException occurs if all parent nodes are not given once in the parentNodes, or if not enough or too many parent values are given. An invalidValueException occurs if a parentValue is invalid for its parent.

abstract public void changeProbFunction(BayesNetworkNode[] parentNodes,
String[] parentValues, double[] probsForValues) throws
ProbabilityException, InvalidValueException;
effects: For the parentNodes and their values given in parentValues (where a parent's value is at the same array index in the parentValues array), this node's value probs are set by probsForValues, where each value in this array refers to the value at this array index returned by getValues(). If this node has no parents, parentNodes and parentValues should be set to null. A probabilityException is thrown if all parent nodes are not given once, if the number of parent values is wrong, or if the wrong number of probs are given for this nodes values. An InvalidValueException is thrown if a parent's value is invalid. The node is left unchanged if an exception occurs.

abstract public void makeObservation(String valueObserved) throws
InvalidValueException;
effects: Set this node's observed value to valueObserved. An invalidValueException is thrown if the value is not valid. An exception leaves this node unchanged.

abstract public void clearObservation();
effects: clear any observation on this node.
abstract public String getObservation();
effects: returns this node's observed value. It returns null if no observation has been made.

abstract public Enumeration getChildren();
effects: returns an enumeration of this node's children nodes.

abstract public Enumeration getParents();
effects: returns an enumeration of this node's parent nodes.

Appendix A.3 – Excerpts from Disease Interface

abstract public void addPerson(BayesNetwork bn, Person indiv);
requires: bn and indiv are not null
effects: This individual is added to bn, and the indiv's phenotype and genotype nodes are created.

abstract public void removePerson(BayesNetwork bn, Person indiv);
requires: bn and indiv are not null
effects: This individual is removed from this bn, and the indiv's phenotype and genotype nodes are set to null. If person has children or parents, nothing is done.

abstract public void addChild(BayesNetwork bn, Person child, Person parent1, Person parent2);
requires: bn and the people are not null. All the people must have been added to the bn and have valid genotype and phenotype nodes. Requires that this child is valid (parents are opposite sex) and no loops are created.
effects: Arcs and probabilistic relations are established in bn to represent the child relationship.

abstract public void removeChild(BayesNetwork bn, Person child, Person parent1, Person parent2);
requires: bn and the people are not null. All the people must have been added to the bn and have valid genotype and phenotype nodes.
effects: Arcs and probabilistic relations are removed in bn to for the child is it was a child of these parents.

Appendix A.4 – Excerpts from the Genetic Screen Descriptor Interface

public GeneticScreen createGeneticScreen(Person p, BayesNetwork bn)
throws InvalidGeneticScreenDescriptorException;
requires: p and bn are not null.
effects: Adds a Genetic Screen to this bn for person p using the description of the Genetic Screen contained in this descriptor.
public void removeGeneticScreen(GeneticScreen gs, BayesNetwork bn;
requires: gs and bn are not null. gs is a Genetic Screen created in bn.
effects: Removes gs from the bn.