Chapter 2

GENOME-WIDE GENETIC ANALYSIS USING GENETIC PROGRAMMING: THE CRITICAL NEED FOR EXPERT KNOWLEDGE

Jason H. Moore[1] and Bill C. White[1]
[1] *Computational Genetics Laboratory, Department of Genetics, Dartmouth Medical School*

Abstract Human genetics is undergoing an information explosion. The availability of chip-based technology facilitates the measurement of thousands of DNA sequence variation from across the human genome. The challenge is to sift through these high-dimensional datasets to identify combinations of interacting DNA sequence variations that are predictive of common diseases. The goal of this study is to develop and evaluate a genetic programming (GP) approach to attribute selection and classification in this domain. We simulated genetic datasets of varying size in which the disease model consists of two interacting DNA sequence variations that exhibit no independent effects on class (i.e. epistasis). We show that GP is no better than a simple random search when classification accuracy is used as the fitness function. We then show that including pre-processed estimates of attribute quality using Tuned ReliefF (TuRF) in a multi-objective fitness function that also includes accuracy significantly improves the performance of GP over that of random search. This study demonstrates that GP may be a useful computational discovery tool in this domain. This study raises important questions about the general utility of GP for these types of problems, the importance of data pre-processing, the ideal functional form of the fitness function, and the importance of expert knowledge. We anticipate this study will provide an important baseline for future studies investigating the usefulness of GP as a general computational discovery tool for large-scale genetic studies.

Keywords: genetic programming, human genetics, expert knowledge, epistasis, multifactor dimensionality reduction

1. Introduction

Genetic programming (GP) is an automated computational discovery tool that is inspired by Darwinian evolution and natural selection (Koza, 1992; Koza, 1994; Koza et al., 1999; Koza et al., 2003; Banzhaf et al., 1998; Langdon, 1998; Haynes et al., 1999). The goal of GP is to evolve computer programs to solve problems. This is accomplished by first generating random computer programs that are composed of the building blocks needed to solve or approximate a solution to a problem. Each randomly generated program is evaluated and the good programs are selected and recombined to form new computer programs. This process of selection based on fitness and recombination to generate variability is repeated until a best program or set of programs is identified. Genetic programming and its many variations have been applied successfully to a wide range of different problems including data mining and knowledge discovery e.g. (Freitas, 2002). Despite the many successes, there are a large number of challenges that GP practitioners and theorists must address before this general computational discovery tool becomes a standard in the modern problem solver's toolbox. (Yu et al., 2005) list 22 such challenges. Several of these are addressed by the present study. First, is GP useful for the analysis of large and high-dimensional datasets? Second, what is the best way to use pre-processing? Third, what is the best way to construct more complicated fitness functions? Finally, what is the best way to incorporate domain-specific knowledge? The goal of this paper is to explore the feasibility of using GP for genome-wide genetic analysis in the domain of human genetics.

The Problem Domain: Human Genetics

Biological and biomedical sciences are undergoing an information explosion and an understanding implosion. That is, our ability to generate data is far outpacing our ability to interpret it. This is especially true in the domain of human genetics where it is now technically and economically feasible to measure thousands of DNA sequence variations from across the human genome. For the purposes of this paper we will focus exclusively on the single nucleotide polymorphism or SNP which is a single nucleotide or point in the DNA sequence that differs among people. It is anticipated that at least one SNP occurs approximately every 100 nucleotides across the $3 * 10^9$ nucleotide human genome. An important goal in human genetics is to determine which of the many thousands of SNPs are useful for predicting who is at risk for common diseases such as prostate cancer, cardiovascular disease, or bipolar depression. This genome-wide approach is expected to revolutionize the genetic analysis of common human diseases (Hirschhorn and Daly, 2005; Wang et al., 2005).

The charge for computer science and bioinformatics is to develop algorithms for the detection and characterization of those SNPs that are predictive of human

health and disease. Success in this genome-wide endeavor will be difficult due to nonlinearity in the genotype-to-phenotype mapping relationship that is due, in part, to epistasis or nonadditive gene-gene interactions. Epistasis was recognized by (Bateson, 1909) nearly 100 years ago as playing an important role in the mapping between genotype and phenotype. Today, this idea prevails and epistasis is believed to be a ubiquitous component of the genetic architecture of common human diseases (Moore, 2003). As a result, the identification of genes with genotypes that confer an increased susceptibility to a common disease will require a research strategy that embraces, rather than ignores, this complexity (Moore, 2003; Moore and Williams, 2005; Thornton-Wells et al., 2004). The implication of epistasis from a data mining point of view is that SNPs need to be considered jointly in learning algorithms rather than individually. Because the mapping between the attributes and class is nonlinear, the concept difficulty is high. The challenge of modeling attribute interactions has been previously described (Freitas, 2001). Due to the combinatorial magnitude of this problem, intelligent feature selection strategies are needed.

A Simple Example of the Concept Difficulty

Epistasis can be defined as biological or statistical (Moore and Williams, 2005). Biological epistasis occurs at the cellular level when two or more biomolecules physically interact. In contrast, statistical epistasis occurs at the population level and is characterized by deviation from additivity in a linear mathematical model. Consider the following simple example of statistical epistasis in the form of a penetrance function. Penetrance is simply the probability (P) of disease (D) given a particular combination of genotypes (G) that was inherited (i.e. $P[D|G]$). A single genotype is determined by one allele (i.e. a specific DNA sequence state) inherited from the mother and one allele inherited from the father. For most single nucleotide polymorphisms or SNPs, only two alleles (encoded by A or a) exist in the biological population. Therefore, because the order of the alleles is unimportant, a genotype can have one of three values: AA, Aa or aa. The model illustrated in Table 2-1 is an extreme example of epistasis. Let's assume that genotypes AA, aa, BB, and bb have population frequencies of 0.25 while genotypes Aa and Bb have frequencies of 0.5 (values in parentheses in Table 2-1). What makes this model interesting is that disease risk is dependent on the particular combination of genotypes inherited. Individuals have a very high risk of disease if they inherit Aa or Bb but not both (i.e. the exclusive OR function). The penetrance for each individual genotype in this model is 0.5 and is computed by summing the products of the genotype frequencies and penetrance values. Thus, in this model there is no difference in disease risk for each single genotype as specified by the single-genotype penetrance values. This genetic model was first described by (Li and

Table 2-1. Penetrance values for genotypes from two SNPs.

	AA (0.25)	Aa (0.50)	aa (0.25)
BB (0.25)	0	1	0
Bb (0.50)	1	0	1
bb (0.25)	0	1	0

Reich, 2000). Heritability or the size of the genetic effect is a function of these penetrance values. In this model, the heritability is maximal at 1.0 because the probability of disease is completely determined by the genotypes at these two DNA sequence variations. This is a special case where all of the heritability is due to epistasis. As (Freitas, 2001) reviews this general class of problems has high concept difficulty.

Genome-Wide Genetic Analysis: A Needle-in-a-Haystack Problem

(Moore and Ritchie, 2004) have outlined three significant challenges that must be overcome if we are to successfully identify genetic predictors of health and disease. First, powerful data mining and machine learning methods will need to be developed to statistically model the relationship between combinations of DNA sequence variations and disease susceptibility. Traditional methods such as logistic regression have limited power for modeling high-order nonlinear interactions (Moore and Williams, 2002). A second challenge is the selection of genetic variables or attributes that should be included for analysis. If interactions between genes explain most of the heritability of common diseases, then combinations of DNA sequence variations will need to be evaluated from a list of thousands of candidates. Filter and wrapper methods will play an important role here because there are more combinations than can be exhaustively evaluated. A third challenge is the interpretation of gene-gene interaction models. Although a statistical model can be used to identify DNA sequence variations that confer risk for disease, this approach cannot be translated into specific prevention and treatment strategies without interpreting the results in the context of human biology. Making etiological inferences from computational models may be the most important and the most difficult challenge of all (Moore and Williams, 2005).

Combining the concept difficulty described in Section 1.3 with the challenge of attribute selection yields what (Goldberg, 2002) calls a needle-in-a-haystack problem. That is, there may be a particular combination of SNPs that together with the right nonlinear function are a significant predictor of disease susceptibility. However, individually they may not look any different than thousands of other SNPs that are not involved in the disease process and are thus noisy.

Under these models, the learning algorithm is truly looking for a genetic needle in a genomic haystack. A recent report from the International HapMap Consortium (Altshuler et al., 2005) suggests that approximately 300,000 carefully selected SNPs may be necessary to capture all of the relevant variation across the Caucasian human genome. Assuming this is true (it is probably a lower bound), we would need to scan $4.5 * 10^{10}$ pairwise combinations of SNPs to find a genetic needle. The number of higher order combinations is astronomical. Is GP suitable for a problem like this? At face value the answer is no. There is no reason to expect that a GP or any other wrapper method would perform better than a random attribute selector because there are no building blocks for this problem when accuracy is used as the fitness measure. The fitness of any given classifier would look no better than any other with just one of the two correct SNPs in the model. Indeed, we have observed this in our preliminary work (White et al., 2005).

Research Questions Addressed

The goal of the present study was to develop and evaluate a GP approach to genetic analysis in the context of genome-wide data. How does GP perform in this problem domain? Is GP a good approach for attribute selection? Is GP better than a random search when there are no building blocks? Is expert knowledge useful for defining building blocks that can be used by the GP?

The rest of this paper is organized in the following manner. Section 2 describes the GP algorithm we used. Section 3 describes the multifactor dimensionality reduction (MDR) method used as a function in the GP trees. Section 4 describes the attribute quality measure that is used as expert knowledge. Section 5 summarizes the data simulation and data analysis methods used to evaluate the GP approaches.

2. Genetic Programming Methods

There are two general approaches to selecting attributes for predictive models. The filter approach pre-processes the data by algorithmically assessing the quality of each attribute and then using that information to select a subset for classification. The wrapper approach iteratively selects subsets of attributes for classification using either a deterministic or stochastic algorithm. The key difference between the two approaches is that the classifier plays no role in selecting which attributes to consider in the filter approach. As (Freitas, 2002) reviews, the advantage of the filter is speed while the wrapper approach has the potential to do a better job classifying. For the problem domain considered here, there is an additional concern that the filter approach may eliminate important attributes from the dataset since no estimator of attribute quality will be perfect across all datasets. Thus, a stochastic wrapper or search method such as GP

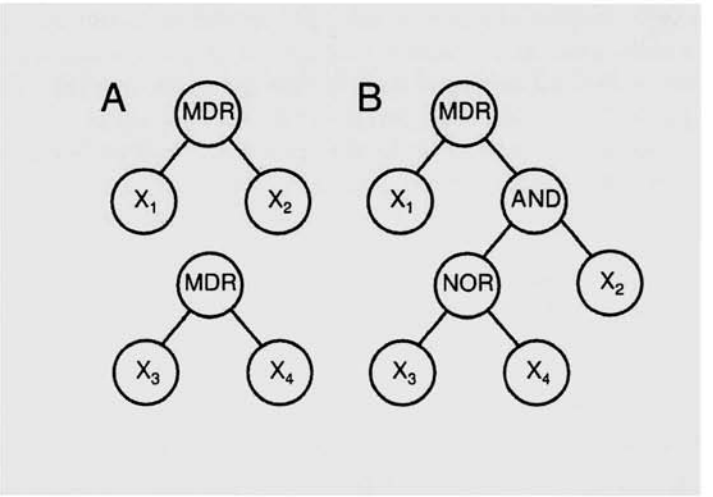

Figure 2-1. Example GP trees for solutions (A). Example of a more complex tree that will be considered in future studies (B).

always maintains some probability of including any attribute in the dataset. The goal of the present study is to develop and evaluate a GP approach to genome-wide genetic analysis. In this initial study, the GP is functioning exclusively as an attribute selector. We have intentionally kept the solution representation simple as a baseline to demonstrate whether the learning algorithm is performing better than random search. Future studies will expand the function set to more than one function.

Tree Representation of Solutions

Figure 2-1A illustrates an example GP tree for this problem. As stated, we have kept the initial solution representation simple with one function in the root node and two children. We have selected the multifactor dimensionality reduction or MDR approach as an attribute constructor for the function set because it is able to capture interaction information (see Section 3). Each tree has two leaves or terminals consisting of attributes. The terminal set consists of 1000 attributes. Although we have started with a simple tree representation, Figure 2-1B illustrates what a more complex tree structure for a higher-order model derived from a larger function set might look like. Expanding the size and complexity of GP trees will be the focus of future studies.

Table 2-2. Summary of GP Parameters.

Population Size	5,000
Generations	10
Crossover	Single-point subtree
Crossover frequency	0.9
Mutation frequency	0.0
Fitness Function	$\alpha * A + \beta * Q$
Selection	Binary tournament
Function Set	MDR
Terminal Set	Attributes 1-1000
Maximum Tree Depth	1.0

Fitness Function

We used a multiobjective fitness function in this study that consisted of two pieces in a simple linear combination of the form $\alpha * A + \beta * Q$. Here, A is our measure of accuracy obtained from the analysis of the single constructed attribute from the GP tree using a naive Bayes classifier. The parameter α is used to weight the accuracy measures. Q in this function represents the attribute quality estimate obtained from pre-processing the attributes using the TuRF algorithm (see Section 4). The parameter β is used to weight the quality measures. We explored parameter settings of $\alpha = 1$ and $\beta = 0$, $\alpha = 1$ and $\beta = 1$, and $\alpha = 1$ and $\beta = 2$. When $\beta = 0$ the fitness is solely determined by accuracy. Both A and Q were scaled using a Z transformation.

Parameter Settings and Implementation

Table 2-2 summarizes the parameter settings for the GP in a Koza-style tableau (Koza, 1992). Since each tree has exactly two attributes, an initial population size of 5,000 trees will include 10,000 total attributes. Since there are only 1,000 attributes in the terminal set we are confident that each attribute will be represented as a building block in the initial population. However, the probability of any one tree receiving both functional attributes (i.e. the solution) is $0.001 * 0.001$ or 10^{-6}. Thus, it is unlikely that any one tree in the initial population will be the correct solution. For the random search, we generated an initial population of $5,000 * 10$ or 50,000 trees and selected the best. The GP was implemented in C++ using GAlib (http://lancet.mit.edu/ga/). The crossover operator was modified to ensure binary trees of depth one.

3. Multifactor Dimensionality Reduction (MDR) for Attribute Construction

Multifactor dimensionality reduction (MDR) was developed as a nonparametric and genetic model-free data mining strategy for identifying combination of SNPs that are predictive of a discrete clinical endpoint (Ritchie et al., 2001; Hahn et al., 2003; Ritchie et al., 2003; Hahn and Moore, 2004; Moore, 2004; Moore et al., 2006). The MDR method has been successfully applied to detecting gene-gene interactions for a variety of common human diseases including, for example, sporadic breast cancer (Ritchie et al., 2001), essential hypertension (Moore and Williams, 2002; Williams et al., 2004), atrial fibrillation (Tsai et al., 2004), myocardial infarction (Coffey et al., 2004), type II diabetes (Cho et al., 2004), prostate cancer (Xu et al., 2005), bladder cancer (Andrew et al., 2006), schizophrenia (Qin et al., 2005), and familial amyloid polyneuropathy (Soares et al., 2005). The MDR method has also been successfully applied in the context of pharmacogenetics and toxicogenetics e.g. (Wilke et al., 2005). At the heart of the MDR approach is an attribute construction algorithm that creates a new attribute by pooling genotypes from multiple SNPs. Constructive induction using the MDR kernel is accomplished in the following way. Given a threshold T, a multilocus genotype combination is considered high-risk if the ratio of cases (subjects with disease) to controls (healthy subjects) exceeds T, else it is considered low-risk. Genotype combinations considered to be high-risk are labeled G1 while those considered low-risk are labeled G0. This process constructs a new one-dimensional attribute with levels G0 and G1. It is this new single variable that is returned by the MDR function in the GP function set. Open-source software in Java and C are freely available from http://www.epistasis.org/mdr.html.

4. Expert Knowledge from Tuned ReliefF

Our goal was to provide an external measure of attribute quality that could be used as expert knowledge by the GP. Here, this external measure used was statistical but could just as easily be biological, for example. There are many different statistical and computational methods for determining the quality of attributes. Our goal was to identify a method that is capable of identifying attributes that predict class primarily through dependencies or interactions with other attributes. (Kira and Rendell, 1992) developed an algorithm called Relief that is capable of detecting attribute dependencies. Relief estimates the quality of attributes through a type of nearest neighbor algorithm that selects neighbors (instances) from the same class and from the different class based on the vector of values across attributes. Weights (W) or quality estimates for each attribute (A) are estimated based on whether the nearest neighbor (nearest hit, H) of a randomly selected instance (R) from the same class and the nearest neighbor from

the other class (nearest miss, M) have the same or different values. This process of adjusting weights is repeated for m instances. The algorithm produces weights for each attribute ranging from -1 (worst) to +1 (best). (Kononenko, 1994) improved upon Relief by choosing n nearest neighbors instead of just one. This new ReliefF algorithm has been shown to be more robust to noisy attributes (Robnik-Sikonja and Kononenko, 2003) and is widely used in data mining applications.

We have previously developed our own extension, Tuned ReliefF (TuRF), that is significantly better than ReliefF in this domain (Moore et al., 2006). ReliefF is able to capture attribute interactions because it selects nearest neighbors using the entire vector of values across all attributes. However, this advantage can also be problematic because the presence of many noisy attributes can reduce the signal the algorithm is trying to capture. The TuRF algorithm systematically removes attributes that have low quality estimates so that the ReliefF values in the remaining attributes can be re-estimated. The motivation behind this algorithm is that the ReliefF estimates of the true functional attributes will improve as the noisy attributes are removed from the dataset. We applied TuRF as described by (Moore et al., 2006) to each dataset.

5. Data Simulation and Analysis

The goal of the simulation study is to generate artificial datasets with high concept difficulty to evaluate the power of GP in the domain of human genetics. We first developed 30 different penetrance functions (see Section 1.3) that define a probabilistic relationship between genotype and phenotype where susceptibility to disease is dependent on genotypes from two SNPs in the absence of any independent effects. The 30 penetrance functions include groups of five with heritabilities of 0.025, 0.05, 0.1, 0.2, 0.3, or 0.4. These heritabilities range from a very small to a large genetic effect size. Each functional SNP had two alleles with frequencies of 0.4 and 0.6. Table 2-3 summarizes the penetrance values to three significant digits for one of the 30 models. The values in parentheses are the genotype frequencies. All 30 models with full precision are available upon request. Each of the 30 models was used to generate 100 replicate datasets with a sample size of 1600. This is a medium sample size for a typical genetic study. Each dataset consisted of an equal number of case (disease) and control (no disease) subjects. Each pair of functional SNPs was combined within a genome-wide set of 998 randomly generated SNPs for a total of 1000 attributes. A total of 3,000 datasets were generated and analyzed. For each set of 100 datasets we count the number of times the correct two functional attributes are selected as the best model by the GP. This count expressed as a percentage is an estimate of the power of the method. That is, how often does GP find the right answer that we know is there? We statistically compared these

Table 2-3. Penetrance values for an example epistasis model.

	AA (0.25)	Aa (0.50)	aa (0.25)
BB (0.25)	0.137	0.484	0.187
Bb (0.50)	0.482	0.166	0.365
bb (0.25)	0.193	0.361	0.430

power estimates between the methods (e.g. random search vs. GP) using a chi-square test of independence. Results were considered statistically significant when the p-value for the chi-square test statistic was ≤ 0.05.

6. Experimental Results

Figure 2-2 summarizes the average power for each method and each heritability level. Each bar in the barplots represents the power averaged over the five different models for each of the heritabilities. Here, power represents the number of times out of 100 replicates the GP found the right two attributes (SNPs). Results are shown for random search (R), GP using classification accuracy (A) as the fitness function ($\alpha = 1$ and $\beta = 0$), GP with accuracy and attribute quality (Q1) with a weight of one as the fitness function ($\alpha = 1$ and $\beta = 1$), and GP with accuracy and attribute quality (Q2) with a weight of two as the fitness function ($\alpha = 1$ and $\beta = 2$).

We find that GP with accuracy (A) as the fitness function does no better than random search (R) across all genetic models and all genetic effect sizes. In a few select cases random search was significantly better ($P < 0.05$) than GP using just accuracy for fitness. One might expect random search to outperform GP in this case because random search consists of one population of 50,000 solutions. The GP only works with an initial population of 5,000 that is then processed for 10 generations. Thus, random search starts with a greater diversity of trees than GP. If GP is truly learning then this difference shouldn't matter.

At a heritability of 0.05 and greater there is clear difference between the GP that uses attribute quality in the fitness function (Q1 and Q2) versus the GP that just uses accuracy (A). This difference was statistically significant ($P < 0.05$) across most models and most heritabilities. Here, GP is also outperforming random search ($P < 0.05$). This is clear evidence that learning is occurring. It is interesting to note that increasing the weight of the attribute quality to twice that of accuracy ($\alpha = 1$ and $\beta = 2$) performed no better than equal weighting ($\alpha = 1$ and $\beta = 1$) ($P > 0.05$).

7. Discussion and Conclusion

There are several important conclusions from this study. First, a GP that uses classifier accuracy as the fitness function does not perform better than random

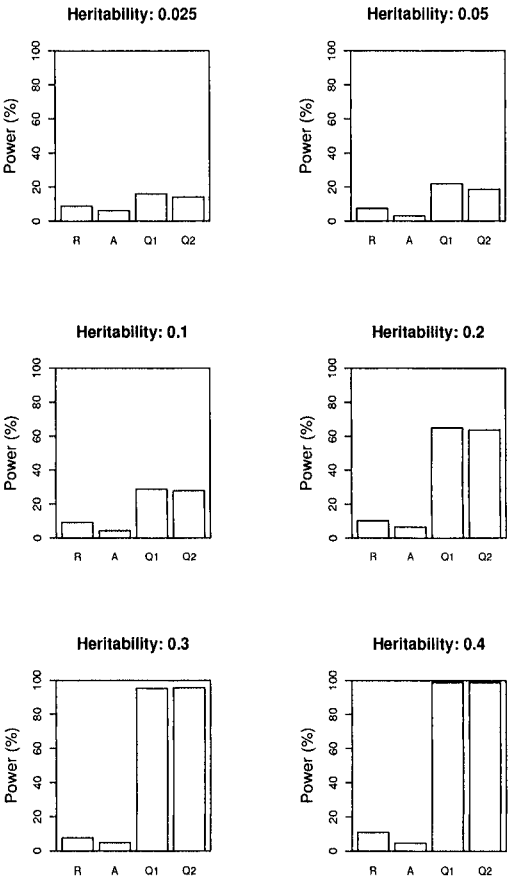

Figure 2-2. Barplots summarizing the power of random search (R), GP using classification accuracy (A) as the fitness function ($\alpha = 1$ and $\beta = 0$), GP with accuracy and attribute quality (Q1) with a weight of one as the fitness function ($\alpha = 1$ and $\beta = 1$), and GP with accuracy and attribute quality (Q2) with a weight of two as the fitness function ($\alpha = 1$ and $\beta = 2$).

search in this specific domain. Second, a multi-objective fitness function that uses expert knowledge in addition to classifier accuracy improves the ability of GP to exploit building blocks and thus learn in a manner that is significantly better than a random search. The discussion of these findings is organized according to the four questions presented in Section 1.1 that are also listed by (Yu et al., 2005).

Is Genetic Programming Useful for the Analysis of Genome-Wide Datasets in the Domain of Human Genetics?

(Langdon, 1998) reviews three general classes of search methods that can be employed for solving large-scale problems. The first and simplest is the enumerative approach. The goal of this search method is to explore all possible solutions. This is clearly the first choice because it is guaranteed to find the best solution. However, it is often the case that the enumerative approach exceeds available computer time. The next class of search methods includes calculus based algorithms. Calculus-based search methods are often looking for maxima or minima using derivatives or gradients. These approaches are also called hill-climbers because they inch towards a global best solution at the top of a smooth hill. The third general class of search algorithms is referred to as stochastic. Stochastic algorithms are based on random number generators and probabilities rather than deterministic rules. The simplest and most naive stochastic search simply generates random solutions that are independently evaluated. Genetic programming is an example of a stochastic search algorithm that usually starts out random and then uses probability functions to select and recombine solutions based on their fitness or value.

Stochastic search algorithms such as GP are more appealing for the genome-wide genetic analysis problem because the search space is astronomical and the fitness landscape is rugged, perhaps even resembling a needle in a haystack. Enumerative approaches aren't computationally feasible and hill-climbers will get lost in the local structure of the fitness landscape. Is a stochastic approach like GP useful for this type of problem? Is it better than a simple random search? Based on the results of the present study we would argue that GP is useful for the analysis of complex genetic datasets only when building blocks are present. When building blocks are not present or are poorly defined a GP may not perform any better than a random search. This is consistent with our previous experiments in this domain (White et al., 2005). This is also consistent with the idea of a competent genetic algorithm (cGA) reviewed by (Goldberg, 2002). Goldberg argues that understanding and exploiting building blocks (schemata) is essential to the success of GAs and by extension to GP (Sastry et al., 2004). There are two important issues here. The first issue is to make sure the building blocks needed to construct good solutions are present.

The second is to make sure the good building blocks are used and exploited during evolution. The present paper uses pre-processing the quality of the attributes to establish building blocks that otherwise don't exist. It was noted by (Yu et al., 2005) that providing rewards for building blocks is necessary for complex adaptation. This idea came from the artificial life work of (Lenski et al., 2003).

How Important is Pre-Processing to the Success of Genetic Programming for Genome-Wide Genetic Analysis?

As described above, the problem as we have defined it lacks building blocks that are critical to GP success. We have approached this problem by first estimating the quality of each genetic attribute or SNP using the TuRF algorithm that is based on ReliefF (see Section 4). Here, we used the attribute quality information as expert knowledge in a multi-objective fitness function. This use of the pre-processing information is described below in Sections 7.3 and 7.4. Although not implemented here, the attribute quality information could also be used to seed an initial GP population as a form of sensible initialization (Ryan and Azad, 2003). This is consistent with Goldberg's ideas for a competent GA (Goldberg, 2002). The idea behind sensible initialization is to fill the initial population with valid solutions.

Do More Complicated Fitness Functions Improve the Success of Genetic Programming for Genome-Wide Genetic Analysis?

We explored two fitness functions in the present study. First, we used a fitness function based exclusively on the estimate of accuracy obtained from a naive Bayes classifier. Second, we used a multi-objective fitness function that included the TuRF score in addition to accuracy in a linear function. We showed that including the expert knowledge in the fitness function significantly improved the performance of the GP. In fact, the GP approaches that measured fitness only as a function of accuracy did not perform better than a simple random search. Both pieces of this fitness function are important. The TuRF scores "help" the fitness by exploiting good building blocks. The accuracy piece comes into play when the right building blocks come together to form a predictive statistical model. One piece of the fitness measure cannot succeed without the other. The use of multi-objective fitness functions has been explored extensively (Coello et al., 2002; Deb, 2001; Zhang and Rockett, 2006). For example, (Koza et al., 2005) used a GP with a multi-objective fitness function that had 16 different pieces to design analog circuits. As (Freitas, 2002) reviews, others have included pre-computed attribute quality estimates in the fitness function for attribute selection e.g. (Bala et al., 1996). Exploring the use of Pareto fronts will also be important.

What is the Best Way to Include Expert Knowledge in Genetic Programming for Genome-Wide Genetic Analysis?

There are multiple different sources of information that could be used as expert knowledge in a GP. In this study, we used a statistical measure of attribute quality. However, future work needs to explore ways to include domain specific knowledge in the GP. There are a number of different public databases available to geneticists that could be mined for expert knowledge. For example, the PubMed database (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed) from the U.S. National Library of Medicine holds over 16 million citations from life science journal articles. There are a number of computational algorithms and tools available now for extracting information such as the co-occurrence of keywords from abstracts from the PubMed database (Jensen et al., 2006). If two genes co-occur frequently in journal abstracts then one could infer that there is a functional relationship. This type of information could be used to guide a GP search for combinations of SNPs that predict disease.

The availability of domain-specific expert knowledge raises the question of the best way to use it in a GP. This is a topic that has received some attention in recent years. (Jin, 2005) covers the use of expert knowledge in population initialization, recombination, mutation, selection, reproduction, multi-objective fitness functions, and human-computer interaction, for example. We focused in this study exclusively on the fitness function. It would be interesting to see if expert knowledge might play an important role in selection, for example. Using TuRF scores for selection might make sense in this domain given accuracy doesn't provide any useful information until the right model is found. Similar arguments could be made for reproduction, recombination and mutation, for example.

Future Studies

This study presents preliminary evidence suggesting that GP might be useful for the genome-wide genetic analysis of common human diseases that have a complex genetic architecture. These results raise numerous questions, some of which have been discussed here. It will be important to extend this study to higher-order genetic models. How well does GP do when faced with finding three, four, or more SNPs that interact in a nonlinear manner to predict disease susceptibility? How does extending the function set to additional attribute construction functions impact performance? How does extending the attribute set impact performance? Is using GP better than available or similar filter approaches? To what extent can GP theory help formulate an optimal GP approach to this problem? Does GP outperform other evolutionary or non-

evolutionary search methods? This paper provides a starting point to begin addressing some of these questions.

8. Acknowledgment

This work was supported by National Institutes of Health (USA) grants LM009012, AI59694, HD047447, RR018787, and HL65234. We also thank the anonymous reviewers for their time and effort to help make this manuscript better.

References

Altshuler, D., Brooks, L.D., Chakravarti, A., Collins, F.S., Daly, M.J., and Donnelly, P. (2005). International hapmap consortium: A haplotype map of the human genome. *Nature*, 437:1299–1320.

Andrew, A.S., Nelson, H.H., Kelsey, K.T., Moore, J.H., Meng, A.C., Casella, D.P., Tosteson, T.D., Schned, A.R., and Karagas, M.R. (2006). Concordance of multiple analytical approaches demonstrates a complex relationship between dna repair gene snps, smoking and bladder cancer susceptibility. *Carcinogenesis*.

Bala, J., Jong, K. De, Huang, J., Vafaie, H., and Wechsler, H. (1996). Using learning to facilitate the evolution of features for recognizing visual concepts. *Evolutionary Computation*, 4:297–312.

Banzhaf, W., Nordin, P., Keller, R.E., and Francone, F.D. (1998). *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers.

Bateson, W. (1909). *Mendel's Principles of Heredity*. Cambridge University Press, Cambridge.

Cho, Y.M., Ritchie, M.D., Moore, J.H., Park, J.Y., Lee, K.U., Shin, H.D., Lee, H.K., and Park, K.S. (2004). Multifactor-dimensionality reduction shows a two-locus interaction associated with type 2 diabetes mellitus. *Diabetologia*, 47:549–554.

Coello, C.A., Veldhuizen, D.A. Van, and Lamont, G.B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer.

Coffey, C.S., Hebert, P.R., Ritchie, M.D., Krumholz, H.M., Morgan, T.M., Gaziano, J.M., Ridker, P.M., and Moore, J.H. (2004). An application of conditional logistic regression and multifactor dimensionality reduction for detecting gene-gene interactions on risk of myocardial infarction: The importance of model validation. *BMC Bioninformatics*, 4:49.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley.

Freitas, A. (2001). Understanding the crucial role of attribute interactions. *Artificial Intelligence Review*, 16:177–199.

Freitas, A. (2002). *Data Mining and KNowledge Discovery with Evolutionary Algorithms*. Springer.

Goldberg, D.E. (2002). *The Design of Innovation*. Kluwer.

Hahn, L.W. and Moore, J.H. (2004). Ideal discrimination of discrete clinical endpoints using multilocus genotypes. *Silico Biology*, 4:183–194.

Hahn, L.W., Ritchie, M.D., and Moore, J.H. (2003). Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. *Bioinformatics*, 19:376–382.

Haynes, Thomas, Langdon, William B., O'Reilly, Una-May, Poli, Riccardo, and Rosca, Justinian, editors (1999). *Foundations of Genetic Programming*, Orlando, Florida, USA.

Hirschhorn, J.N. and Daly, M.J. (2005). Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(95):108–118.

Jensen, L.J., Saric, J., and Bork, P. (2006). Literature mining for the biologist: from information retrieval to biological discovery. *Nature Review Genetics*, 7:119–129.

Jin, Y. (2005). *Knowledge Incorporation in Evolutionary Computation*. Springer.

Kira, K. and Rendell, L.A. (1992). A practical approach to feature selection. In *Machine Learning: Proceedings of the AAAI'92*.

Kononenko, I. (1994). Estimating attributes: analysis and extension of relief. *Machine Learning: ECML*, 94:171–182.

Koza, John R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Koza, John R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge Massachusetts.

Koza, John R., Andre, David, Bennett III, Forrest H, and Keane, Martin (1999). *Genetic Programming 3: Darwinian Invention and Problem Solving*. Morgan Kaufman.

Koza, John R., Keane, Martin A., Streeter, Matthew J., Mydlowec, William, Yu, Jessen, and Lanza, Guido (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers.

Koza, J.R., Jones, L.W., Keane, M.A., Streeter, M.J., and Al-Sakran, S.H. (2005). Toward automated design of industrial-strength analog circuits by means of genetic programming. In O'Reilly, U.M., Yu, T., Riolo, R., and Worzel, B., editors, *Genetic Programming Theory and practice*. Springer.

Langdon, William B. (1998). *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!*, volume 1 of *Genetic Programming*. Kluwer, Boston.

Lenski, R.E., Ofria, C., Pennock, R.T., and Adami, C. (2003). The evolutionary origin of complex features. 423:139–144.

Li, W. and Reich, J. (2000). A complete enumeration and classification of two-locus disease models. *Human Heredity*, 50:334–349.

Moore, J.H. (2003). The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Human Heredity*, 56:73–82.

Moore, J.H. (2004). Computational analysis of gene-gene interactions in common human diseases using multifactor dimensionality reduction. *Expert Rev. Mol Diagn*, 4:795–803.

Moore, J.H., Gilbert, J.C., Tsai, C.T., Chiang, F.T., Holden, W., Barney, N., and White, B.C. (2006). A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. *Journal of Theoretical Biology*.

Moore, J.H. and Ritchie, M.D. (2004). The challenges of whole-genome approaches to common diseases. *JAMA*, 291:1642–1643.

Moore, J.H. and Williams, S.W. (2002). New strategies for identifying gene-gene interactions in hypertension. *Annals of Medicine*, 34:88–95.

Moore, J.H. and Williams, S.W. (2005). Traversing the conceptual divide between biological and statistical epistasis: Systems biology and a more mordern synthesis. *BioEssays*, 27:637–646.

Qin, S., Zhao, X., Pan, Y., Liu, J., Feng, G., Fu, J., Bao, J., Zhang, Z., and He, L. (2005). An association study of the n-methyl-d-aspartate receptor nr1 subunit gene (grin1) and nr2b subunit gene (grin2b) in schizophrenia with universal dna microarray. *European Journal of Human Genetics*, 13:807–814.

Ritchie, M.D., Hahn, L.W., and Moore, J.H. (2003). Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, phenocopy and genetic heterogeneity. *Genetic Epidemiology*, 24:150–157.

Ritchie, M.D., Hahn, L.W., Roodi, N., Bailey, L.R., Dupont, W.D., Parl, F.F, and Moore, J.H. (2001). Multifactor dimensionality reduction reveals high-order interactions among estrogen metabolism genes in sporadic breast cancer. *American Journal of Human Genetics*, 69:138–147.

Robnik-Sikonja, M. and Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69.

Ryan, C. and Azad, R.M. (2003). Sensible initialization in chorus. *EuroGP 2003*, pages 394–403.

Sastry, Kumara, O'Reilly, Una-May, and Goldberg, David E. (2004). Population sizing for genetic programming based on decision making. In O'Reilly, Una-May, Yu, Tina, Riolo, Rick L., and Worzel, Bill, editors, *Genetic Programming Theory and Practice II*, chapter 4, pages 49–65. Springer, Ann Arbor.

Soares, M.L., Coelho, T., Sousa, A., Batalov, S., Conceicao, I., Sales-Luis, M.L., Ritchie, M.D., Williams, S.M., Nievergelt, C.M., Schork, N.J., Saraiva, M.J., and Buxbaum, J.N. (2005). Susceptibility and modifier genes in portuguese

transthyretin v30m amyloid polygeuropathy: complexity in a single-gene disease. *Human Molecular Genetics*, 14:543–553.

Thornton-Wells, T.A., Moore, J.H., and Haines, J.L. (2004). Genetics, statistics and human disease: analytical retooling for complexity. *Trends in Genetics*, 20:640–647.

Tsai, C.T., Lai, L.P., Lin, J.L., Chiang, F.T., Hwang, J.J., Ritchie, M.D., Moore, J.H., Hsu, K.L., Tseng, C.D., Liau, C.S., and Tseng, Y.Z. (2004). Renin-angiotensin system gene polymorphisms and atrial fibrillation. *Circulation*, 109:1640–1646.

Wang, W.Y., Barratt, B.J., Clayton, D.G., and Todd, J.A. (2005). Genome-wide association studies: theoretical and practical concerns. *Nature Reviews Genetics*, 6:109–118.

White, B.C., Gilbert, J.C., Reif, D.M., and Moore, J.H. (2005). A statistical comparison of grammatical evolution strategies in the domain of human genetics. *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 676–682.

Wilke, R.A., Reif, D.M., and Moore, J.H. (2005). Combinatorial pharmacogenetics. *Nature Reviews Drug Discovery*, 4:911–918.

Williams, S.M., Ritchie, M.D., 3rd, J.A. Phillips, Dawson, E., Prince, M., Dzhura, E., Willis, A., Semenya, A., Summar, M., White, B.C., Addy, J.H., Kpodonu, J., Wong, L.J., Felder, R.A., Jose, P.A., and Moore, J.H. (2004). Multilocus analysis of hypertension: a hierarchical approach. *Human Heredity*, 57:28–38.

Xu, J., Lowery, J., Wiklund, F., Sun, J., Lindmark, F., Hsu, F.C., Dimitrov, L., Chang, B., Turner, A.R., Adami, H.O., Suh, E., Moore, J.H., Zheng, S.L., Isaacs, W.B., Trent, J.M., and Gronberg, H. (2005). The interaction of four inflammatory genes significantly predicts prostate cancer risk. *Cancer Epidemiology Biomarkers and Prevention*, 14:2563–2568.

Yu, Tina, Riolo, Rick L., and Worzel, Bill (2005). Genetic programming: Theory and practice. In Yu, Tina, Riolo, Rick L., and Worzel, Bill, editors, *Genetic Programming Theory and Practice III*, volume 9 of *Genetic Programming*, chapter 1, pages 1–14. Springer, Ann Arbor.

Zhang, Yang and Rockett, Peter I. (2006). Feature extraction using multi-objective genetic programming. In Jin, Yaochu, editor, *Multi-Objective Machine Learning*, volume 16 of *Studies in Computational Intelligence*, chapter 4, pages 79–106. Springer. Invited chapter.