A Statistical Comparison of Grammatical Evolution Strategies in the Domain of Human Genetics

Bill C. White
Computational Genetics Laboratory
Department of Genetics
Dartmouth Medical School
Lebanon, NH 03756
bill.c.white@dartmouth.edu

Joshua C. Gilbert
Computational Genetics Laboratory
Department of Genetics
Dartmouth Medical School
Lebanon, NH 03756
joshua.c.gilbert@dartmouth.edu

Jason H. Moore
Computational Genetics Laboratory
Department of Genetics
Dartmouth Medical School
Lebanon, NH 03756
Department of Computer Science
University of New Hampshire
Durham, NH 03824
jason.h.moore@dartmouth.edu

ABSTRACT

Detecting and characterizing genetic predictors of human disease susceptibility is an important goal in human genetics. New chip-based technologies are available that facilitate the measurement of thousands of DNA sequence variations across the human genome. Biologically-inspired stochastic search algorithms are expected to play an important role in the analysis of these high-dimensional datasets. We simulated datasets with up to 6000 attributes using two different genetic models and statistically compared the performance of grammatical evolution, grammatical swarm, and random search for building symbolic discriminant functions. We found no statistical difference among search algorithms within this specific domain.

Categories and Subject Descriptors

I.5 [**Computing Methodologies**]: Pattern Recognition; J.3 [**Computer Applications**]: Life and Medical Sciences

General Terms

Algorithms, Performance

Keywords

Grammatical swarm, Genetic algorithm, Particle swarm optimization, Random search, Symbolic discriminant analysis

1. INTRODUCTION

The identification of genetic polymorphisms or DNA sequence variations that are predictive of disease risk is expected to improve human health be leading to the development of effective clinical strategies for diagnosis, prevention and treatment. Success in this endeavor will depend critically on several factors including the availability of genetic polymorphisms that capture the relevant DNA sequence variation across the entire human genome. Technology is available today to measure 10^5 or more single nucleotide polymorphisms (SNPs) across the human genome in genetic and epidemiological samples of human subjects [11]. These chip-based technologies are still too expensive for large-scale use but should become practical within the next several years. While as many as $3 * 10^7$ SNPs may exist in human populations, it is generally believed that only 10^5 to 10^6 SNPs will actually need to measured to capture most of the variability in the human genome due to regions of high allelic correlation or linkage disequilibrium [3]. Even so, 10^5 or 10^6 SNPs is an enormous number of attributes in a dataset. A number of computational and statistical challenges arise when dealing with data of such a high-dimensionality [15] [6] [24]. For example, the number of false-positives is expected to be high in a traditional statistical analysis of all 10^6 SNPs.

An important factor to consider when undertaking a genetic study of human disease is that the mapping relationship between genotype to phenotype (i.e. genetic architecture) is expected to be complex [23]. Part of the complexity can be attributed to epistasis or nonlinear gene-gene interaction that has been historically defined as one gene masking the effects of another gene [1] or deviation from additivity in a linear statistical model [5]. It is our working hypothesis that epistasis will be a ubiquitous component of the genetic architecture of common human diseases and thus must be explicitly modeled in genetic and epidemiological studies [16] [14]. An important computational consideration is that statistical modeling of epistasis requires that combinations of SNPs be evaluated. With 10^5 to 10^6 attributes, the number of combinations that need to be assessed very quickly becomes astronomical [15]. Thus, deterministic search algorithms aren't practical. As an alternative, stochastic algorithms, such as those from biologically-inspired computing, need to be explored and evaluated.

The goal of the present study is to statistically evaluate and

compare two recently developed biologically-inspired algorithms by O'Neill et al., grammatical evolution [18] [19] and grammatical swarm [17], for their ability to identify symbolic discriminant function models of gene-gene interactions in artificial datasets of varying size and complexity [13] [12]. As a baseline, we also compare both grammatical strategies to a random search.

2. DATA SIMULATION

The goal of the data simulation was to generate artificial datasets that could be used to statistically evaluate and compare different computational modeling strategies in the domain of human genetics. The simulation strategy used has been previously described in detail by Reif et al. [20] and is briefly described here. We first simulated two single nucleotide polymorphisms (SNPs) each with equal allele frequencies ($p = .5$, $q = .5$) and genotype frequencies consistent with Hardy-Weinberg equilibrium (p^2, $2pq$, q^2). From each SNP, a hypothetical protein product was generated in which 60% of the variation was due to additive (i.e. linear) effects of the three genotypes. Genotypes and proteins were simulated using the Genometric Analysis Simulation Package or GASP [25]. Whether a subject in the dataset had a hypothetical disease or not was determined by a function of discretized values of the two simulated proteins (low, medium, high). Here, we used two different models of disease susceptibility. In the first model (M27), probability of disease given high or medium levels of both proteins is one, and zero otherwise. The M27 model specifies both a main effect of each SNP and an interaction (i.e. epistasis). In the second model (M170), probability of disease given medium levels of one protein, the other, but not both, is one, and zero otherwise. This model is based on the XOR function that is not linearly separable and thus specifies and interaction between each SNP in the absence of any independent main effects. Both the M27 and M170 models have been described previously [10]. Figure 1 summarizes the simulation models. We then added each functional SNP to datasets consisting of additional non-functional or randomly generated SNPs. Each dataset consisted of either 60, 600, or 6000 total attributes. This range of attributes is consistent with typical genetic and epidemiologic datasets. The total number of instances or hypothetical human subjects was 200 with disease and 200 healthy controls. Two model types and three attribute counts yielded six different generative function for the artificial datasets.

3. SYMBOLIC DISCRIMINANT ANALYSIS

It is common to utilize parametric linear models for the detection and characterization of gene-gene interactions in genetic and epidemiologic studies of human disease [4]. An important limitation of parametric statistical approaches such as linear discriminant analysis and logistic regression is the need to pre-specify the functional form of the model. To address this limitation, Moore et al. [13] [12] developed symbolic discriminant analysis or SDA for automatically identifying the optimal functional form and coefficients of discriminant functions that may be linear or nonlinear. This is accomplished by providing a list of mathematical functions and a list of explanatory variables that can be used to build discriminant scores. Similar to Koza's symbolic regression [9], genetic programming (GP) is utilized to perform a parallel search for a combination of functions and variables that optimally discriminate between two endpoint groups. The primary advantage of this approach is that the functional form of the statistical model does not need to be pre-specified. This is important for the identification of combinations of SNPs in genes whose relationship with the clinical endpoint of interest may be nonadditive or nonlinear [23] [14]. In the present study, we investigate the use of grammatical evolution or GE [18] [19] and grammatical swarm or GS [17] for the discovery of symbolic discriminant functions. Random search (RS) was also investigated. These grammatical approaches are described below. Table 1 summarizes the function set and parameters used in all algorithms. population size, number of generations, and chromosome length that were all held constant across the three search methods. Method-specific parameters are discussed below.

The performance of SDA with GE, GS, and RS wrappers was assessed by estimating the prediction error of symbolic discriminant functions using two-fold cross-validation. Best models were selected as those that minimize the difference between the classification error as assessed in the training set and the prediction error as assessed using the testing set. This greatly reduces overfitting as has been suggested by Rowland [21].

4. A GRAMMAR FOR SYMBOLIC DISCRIMINANT FUNCTIONS

Both grammatical evolution and grammatical swarm use a grammar to generate candidate solutions. The following shows the full grammar used by these algorithms to generate symbolic discriminant functions. Backus-Naur Form (BNF) is a formal notation for describing the syntax of a context-free grammar as a set of production rules that consist of terminals and nonterminals [7]. Nonterminals form the left-hand side of production rules while both terminals and nonterminals form the right-hand side. A terminal is a function component, and a nonterminal is the name of a production rule. Use of nonterminals in the right-hand side of production rules allows for recursion, deriving more complex sentences, thus functions, by expanding these nonterminals recursively. For symbolic discriminant functions, the terminal set includes, for example, the basic building blocks of a function: unary and binary expressions, and IF statements with conditionals. The nonterminal set includes the names of production rules that construct the parts of the function. The complete grammar specification follows. Beginning with the start symbol *expression*, complete functions are derived by decoding GA chromosomes or GS/RS vectors.

⟨*expression*⟩ ::= (⟨*unary_operator*⟩ ⟨*operand*⟩)
| (⟨*binary_operator*⟩ ⟨*operand*⟩ ⟨*operand*⟩)
| (IF ⟨*conditional_expression*⟩ ⟨*expression*⟩ ⟨*expression*⟩)

⟨*conditional_expression*⟩ ::= (⟨*conditional_operator*⟩ ⟨*operand*⟩ ⟨*operand*⟩)
| (⟨*logical_operator_1*⟩ ⟨*operand*⟩)
| (⟨*logical_operator_2*⟩ ⟨*operand*⟩ ⟨*operand*⟩)

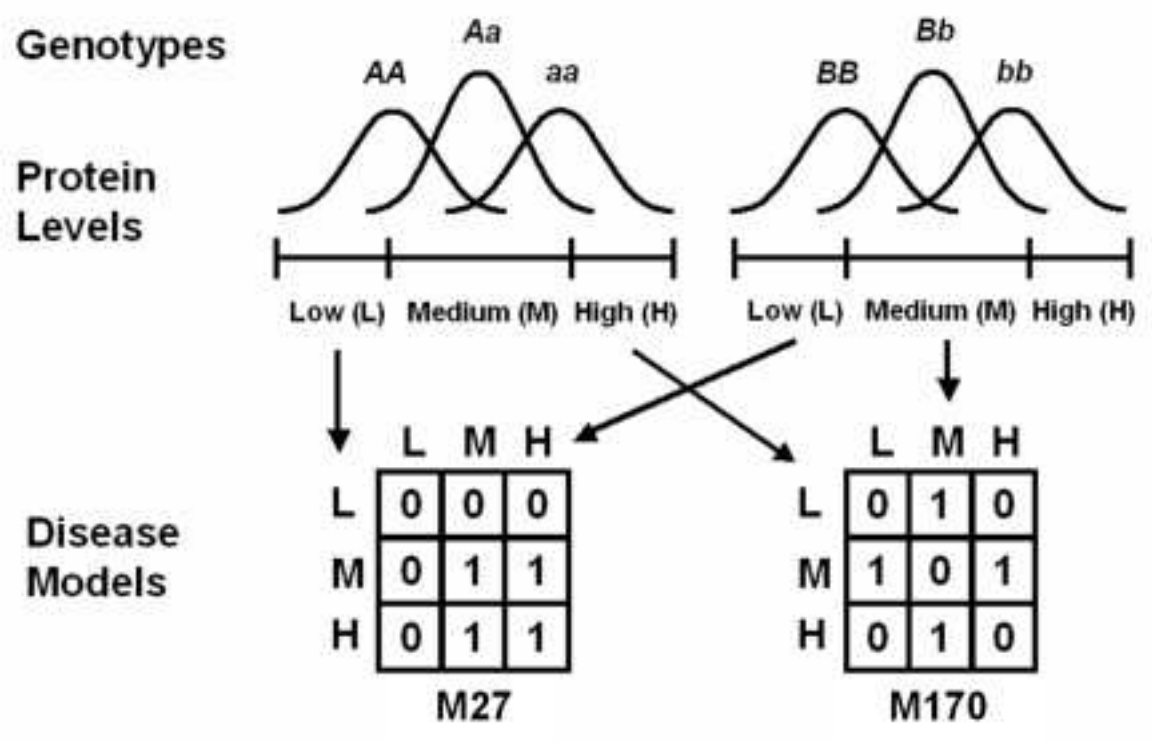

Figure 1: Data simulation models.

Objective:	Induce symbolic discriminant functions for SDA that classify case-control datasets.
Terminal set:	Variables and generated constants.
Function set:	+,-,*,/,IF,AND,OR,XOR,TRUE,NOT,TRUE,SIN,COS,RLOG,MIN,MAX
Fitness cases:	Genetic datasets of genotypes of 100 individuals labeled case or control.
Raw fitness:	The number of points misclassified.
Standardized fitness:	Misclassification error.
Wrapper:	Maps S-expressions with positive values to 1 and all others to 0.
Parameters:	Population size = 1000, Number of generations = 500, Number of codons = 500, Codon values = (0, 255).
Success predicate:	0.0 classification error.

Table 1: Koza-style tableau of algorithmic parameters in common.

Crossover rate:	0.9
Mutation rate:	0.01

Table 2: GE parameters.

c1:	1.0
c2:	1.0
VMIN:	0.0
VMAX:	255.0

Table 3: GS parameters.

⟨*constant_expression*⟩ ::= ⟨*constant*⟩
| (⟨*binary_operator*⟩ ⟨*constant_expression*⟩ ⟨*constant_expression*⟩)

⟨*operand*⟩ ::= ⟨*constant*⟩
| (VAR ⟨*constant_expression*⟩)
| ⟨*expression*⟩

⟨*binary_operator*⟩ ::= +
| -
| *
| /
| MIN
| MAX

⟨*unary_operator*⟩ ::= SIN
| COS
| RLOG

⟨*logical_operator_1*⟩ ::= NOT
| TRUE

⟨*logical_operator_2*⟩ ::= AND
| OR
| XOR

⟨*conditional_operator*⟩ ::= LT
| LE
| GT
| GE
| EQ
| NE

5. GRAMMATICAL EVOLUTION STRATEGY

Grammatical evolution (GE) is a flexible evolutionary computing (EC) method first described by O'Neill and Ryan [18] [19] as a variation on genetic programming (GP) as presented by Koza [9]. Here, a BNF grammar is specified that allows a computer program or model to be constructed by a simple genetic algorithm (GA) operating on a vector of numbers called codons (the GA chromosome). Each codon is used to decode a choice from the grammar beginning with the start symbol. Each codon is applied with a modulus operator and the number of choices possible from the grammar to select a grammar production. The process continues until all nonterminals have been replaced with terminals from the grammar. At this point a symbolic discriminant function has been generated. This procedure is repeated for each individual in the current population. Through crossover and mutation at every iteration (generation) of the algorithm, GA chromosomes evolve to produce better grammar decodings, thus minimizing the classification error. The GE parameters use are shown in Table 2.

6. GRAMMATICAL SWARM STRATEGY

Particle swarm optimization (PSO) was first introduced by Kennedy and Eberhart [8] as a search and optimization algorithm operating in real number space. O'Neill and Brabazon [17] incorporated PSO into a new GE method called grammatical swarm (GS). In this approach the GA chromosome is replaced with particle swarm vectors for position on velocity. The PSO position vector is constrained to the same codon range as that described in Table 1 above. PSO vectors are modified over the run by querying the global and personal best particle positions and velocities to update the current particle's position (codon vector) and velocity. The velocity and location vector update formulas are those used by O'Neill and Barbazon [17]. Velocity is updated:

$$v_i = (w*v_i)+(c1*R_1*(p_{best}-p_i))+(c2*R_2*(g_{best}-p_i)) \quad (1)$$

where w is a weighting factor that adapts over the run:

$$w = wmax - ((wmax - wmin)/itermax) * iter \quad (2)$$

Position is then updated:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

The update is performed for all particles in the swarm (population). The resulting location vectors are used as the chromosomes for GE. Over time the particles are expected to converge on better solutions. GS-specific parameters are shown in Table 3.

7. RANDOM SEARCH STRATEGY

The random search (RS) strategy is a very simple form of GE using randomly-generated codon vectors. For a population size m running n generations in the GE/GS algorithms, $m * n$ random chromosomes were generated (with replacement)and evaluated. The best vector from all evaluations was chosen as the best-of-run.

8. EXPERIMENTAL DESIGN AND ANALYSIS

The goal of our experimental design and analysis was to statistically compare the performance of the GE, GS, and RS search strategies. This was accomplished by specifying a full factorial experimental design [2] with three simulated datasets per level combination. We evaluated 1) the simulation model used (M27 and M170), 2) the number of attributes (60, 600, and 6000), and 3) the search or wrapper method used (GE, GS, RS). All other factors such as population size, number of generations, and the chromosome or vector length were held constant for all analyses as described above. Thus, we evaluated the effects of three factors with 18 total level combinations on the prediction error of SDA models. Table 4 lists all of the factors and their level combinations. We employed analysis of variance (ANOVA)

to test the null hypothesis that average prediction error is not different among levels within each of the three factors (i.e. independent main effects) and not different among levels within pairwise combinations of factors (i.e. interaction effects). All results were considered statistically significant at a type I error or false-positive rate of $\alpha = 0.05$.

9. RESULTS

Table 4 summarizes the mean and standard deviation of prediction errors for each of the three search algorithms across each of the six different types of datasets. Table 5 summarizes the ANOVA results testing the main effect of each factor and all the pairwise interactions. The model used (M27 or M170) and the number of attributes (60, 600, or 6000) had the largest independent main effects on prediction error with p-values between 0.05 and 0.10 indicating a trend toward statistical significance. The search method used (GE, GS, or RS) had no effect on prediction error ($p > 0.10$) and there were no significant interactions among the factors considered ($p > 0.10$).

Figure 2 summarizes the distribution of prediction errors for each single factor using boxplots. Boxplots summarize distributions of continuous data by plotting the interquartile range or the inner 50% of the data in the box and the full range of the data in the whiskers or lines drawn outside the box. The horizontal line within each box represents the median of the distribution. The diagonal notches in the boxes represent an approximate 95% confidence interval around the median and can be used to statistically compare the medians between the different factor levels. It is clear in the first plot that the median prediction error is significantly higher (worse) for the M170 model than the M27 model. Further, the median prediction error for the datasets with 6000 attributes is higher than for the datasets with either 60 or 600 attributes. Note that the confidence intervals for each of the three methods in the third plot all overlap indicating there is no difference in performance between GE, GS, and RS. These results are all consistent with the statistical comparison using ANOVA.

10. DISCUSSION AND CONCLUSIONS

The goal of this study was to use rigorous experimental design and statistical analysis methods to compare grammatical evolution strategies for attribute selection and modeling of gene-gene interactions in high-dimensional datasets from the domain of human genetics. The main finding of this study is that there is no statistical difference in the performance of grammatical evolution (GE) and grammatical swarm (GS) as assessed by prediction error estimated using cross-validation. Not only was the performance of the grammatical learning strategies not statistically different from one another, they were not statistically different from the performance of a simple random search (RS). This was true across the different models and the different numbers of attributes which both had an effect on performance of all three methods, as expected. On the basis of these results, and within this narrowly defined domain of human genetics, we conclude that a RS is just as effective as GE or GS.

While the results of this study suggest the grammatical strategies are no better than RS in this specific domain, it is perhaps premature to conclude they should be abandoned altogether. First, we compared the three algorithms for a fixed population size (1000), a fixed number of generations (500), and a fixed chromosome or vector length (500). It is possible, for example, that GE and/or GS will outperform RS (or vice versa) with a different set of parameter settings (e.g. smaller population size). Exploring this possibility is the focus of ongoing studies. Second, the use of grammars is very appealing because of their flexibility and grounding in computer science theory. It is entirely possible that using a genetic algorithm or a particle swarm algorithm with a grammar for automatic programming is not optimal for the human genetics domain defined in this paper. For example, the crossover operator in GE can be very disruptive and may decrease performance [19]. The GS approach does not use crossover but may not be ideal for problems where there is not a gradient of real values. It will be important to explore other grammar-based search algorithms such as those based on estimation of distribution algorithms [22]. Thus, whether grammar-based search algorithms will be useful in the domain of human genetics is still and open question.

As human genetics moves from measuring several DNA sequence variations in a handful of genes, to measuring every informative variation in the entire human genome, it will be important to explore the use of stochastic search algorithms such as those that are inspired by biology and evolution. Developing and evaluating intelligent search algorithms should be a priority in this domain, especially for common human disease such as cancer and cardiovascular disease where the underlying genetic architecture is expected to be highly nonlinear. The present study is a first step to scaling our data analysis algorithms to the dimensionality of the entire human genome.

11. ACKNOWLEDGEMENTS

This work was supported by National Institutes of Health grants AI59694 and RR018787.

12. ADDITIONAL AUTHORS

Additional authors: David M. Reif (Dartmouth Medical School email: `david.m.reif@vanderbilt.edu`).

13. REFERENCES

[1] W. Bateson. *Mendel's Principles of Heredity*. Cambridge University Press, 1909.

[2] G. Box, W. Hunter, and J. Hunter. *Statistics for experimenters*. Wiley, New York, 1978.

[3] C. Carlson, M. Eberle, L. Kruglyak, and D. Nickerson. Mapping complex disease loci in whole-genome association studies. *Nature*, 429(6990):446–52, 2004.

[4] H. Cordell. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics*, 11(20):2463–8, 2002.

[5] R. A. Fisher. The correlations between relatives on the supposition of mendelian inheritance. *Transactions of the Royal Society of Edinburg*, 52:399–433, 1918.

[6] J. Hirschhorn and M. Daly. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, 2005.

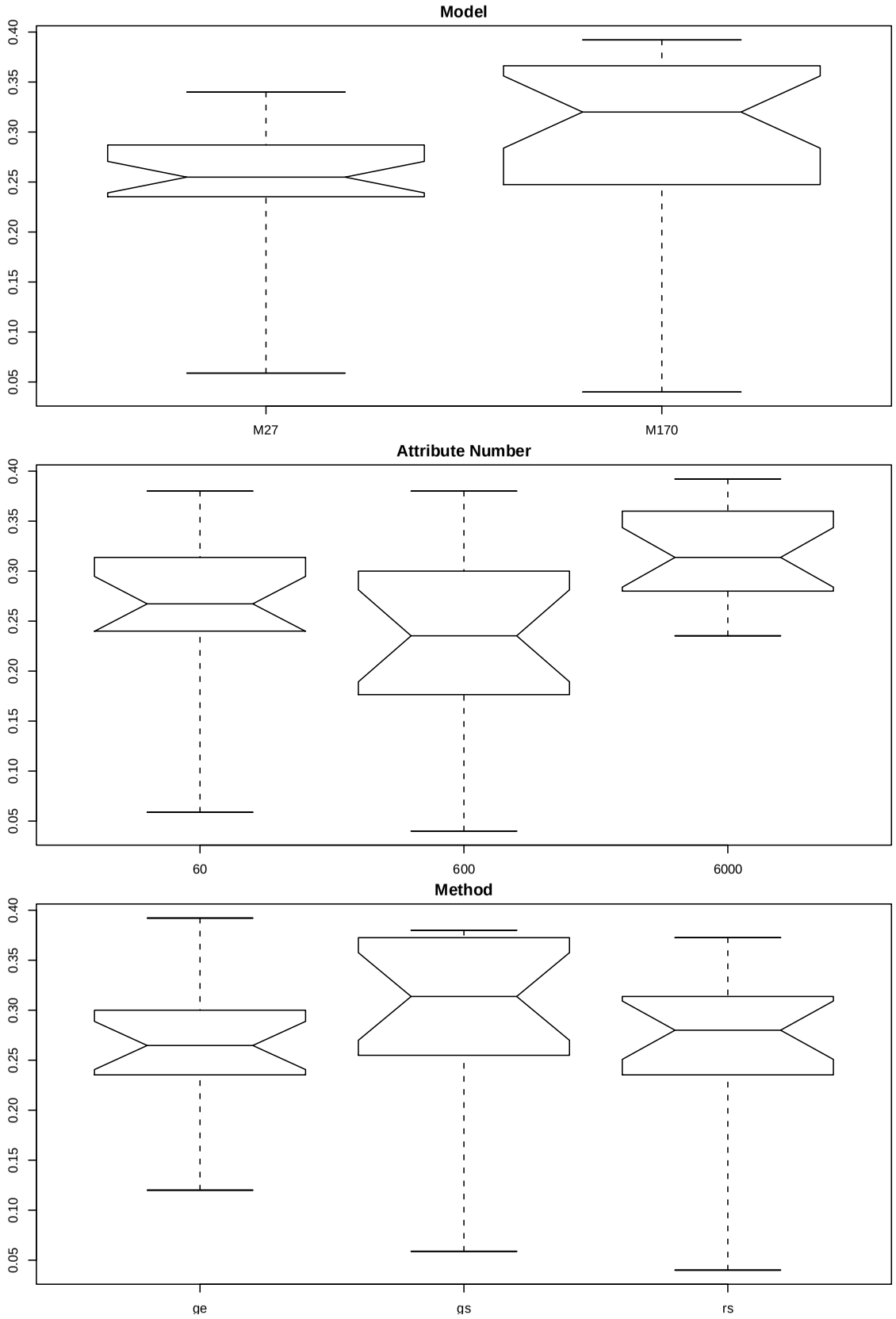

Figure 2: Boxplots summarizing the distribution of prediction errors for each factor.

			Method	
Model	Attributes	GE	GS	RS
M27	60	0.2531 (0.0012)	0.2432 (0.0057)	0.2471 (0.0015)
M27	600	0.2560 (0.0017)	0.2519 (0.0020)	0.2321 (0.0015)
M27	6000	0.2768 (0.0011)	0.2807 (0.0009)	0.2552 (0.0022)
M170	60	0.2564 (0.0079)	0.3579 (0.0014)	0.3221 (0.0013)
M170	600	0.2895 (0.0123)	0.3631 (0.0005)	0.1489 (0.0034)
M170	6000	0.3499 (0.0020)	0.3576 (0.0010)	0.3498 (0.0004)

Table 4: Summary results for all runs.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
method	1	0.001706	0.001706	0.2687	0.60663
model	1	0.018649	0.018649	2.9369	0.09317 .
attributes	1	0.020246	0.020246	3.1883	0.08062 .
method:model	1	0.000013	0.000013	0.0021	0.96356
method:attributes	1	0.002576	0.002576	0.4056	0.52729
model:attributes	1	0.000037	0.000037	0.0058	0.93952
method:model:attributes	1	0.005306	0.005306	0.8327	0.36626

Table 5: ANOVA summary results.

[7] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.

[8] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kauffman Publishers, San Francisco, 2001.

[9] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[10] W. Li and J. Reich. A complete enumeration and classification of two-locus disease models. *Human heredity*, 50(6):334–49, 2000.

[11] H. Matsuzaki, S. Dong, H. Loi, X. Di, G. Liu, E. Hubbell, J. Law, T. Berntsen, M. Chadha, H. Hui, G. Yang, G. Kennedy, T. Webster, S. Cawley, P. Walsh, K. Jones, S. Fodor, and R. Mei. Genotyping over 100,000 snps on a pair of oligonucleotide arrays. *Nature Methods*, 1(2):109–11, 2004.

[12] J. Moore, J. Parker, and L. Hahn. Symbolic discriminant analysis for mining gene expression patterns. 2167:191–205, 2001.

[13] J. Moore, J. Parker, N. Olsen, and T. Aune. Symbolic discriminant analysis of microarray data in autoimmune disease. *Genetic Epidemiology*, 23:57–69, 2002.

[14] J. H. Moore. The ubiquitous nature of epistasis in determining susceptibility to common human diseases. *Human Heredity*, 2003.

[15] J. H. Moore and M. Ritchie. The challenges of whole-genome approaches to common diseases. *Journal of the American Medical Association*, 291:1642–1643, 2004.

[16] W. S. Moore JH. New strategies for identifying gene-gene interactions in hypertension. *Annals of Medicine*, 34:88–95, 2002.

[17] M. O'Neill and A. Brabazon. Grammatical swarm. In K. Deb, R. Poli, and W. Banzhaf, editors, *Lecture Notes in Computer Science*, volume 3102. Springer, New York, 2004.

[18] M. O'Neill and C. Ryan. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5:349–358, 2001.

[19] M. O'Neill and C. Ryan. *Grammatical Evolution*. Klewer, Boston, 2003.

[20] D. Reif, B. White, and J. Moore. Integrated analysis of genetic, genomic, and proteomic data. *Expert Review of Proteomics 1*, pages 67–75, 2004.

[21] J. Rowland. Model selection methodology in supervised learning with evolutionary computation. *Biosystems*, 72(1-2):187–196, 2003.

[22] Y. Shan., R. McKay, H. Abbass, , and D. Essam. Program distribution estimation with grammar models. In *Proceedings of the 8th Asia-Pacific Symposium on Intelligent and Evolutionary Systems*, pages 191–205, 2004.

[23] T. Thornton-Wells, J. H. Moore, and J. Haines. Genetics, statistics and human disease: analytical retooling for complexity. *Trends in Genetics*, 2004.

[24] W. Wang, B. Barratt, D. Clayton, and J. Todd. Genome-wide association studies: theoretical and practical concerns. *Nature Reviews Genetics*, 6(2):109–18, 2005.

[25] A. Wilson, B.-J. Wilson, E. Pugh, and et al. The genometric analysis simulation program (g.a.s.p.): A software tool for testing and investigating methods in statistical genetics. *American Journal of Human Genetics*, 59:A193, 1996.