The Power of Quantitative Grammatical Evolution Neural Networks to Detect Gene-Gene Interactions

Nicholas E. Hardison
Bioinformatics Research Center
North Carolina State University
1 Lampe Dr, Raleigh NC, USA
1 (919) 515-3574
nhardis@ncsu.edu

Alison A. Motsinger-Reif
Bioinformatics Research Center, Department of Statistics
North Carolina State University
307 Ricks Hall, 1 Lampe Dr, Raleigh NC, USA
1 (919) 515-3574
motsinger@stat.ncsu.edu

ABSTRACT
Applying grammatical evolution to evolve neural networks (GENN) has been increasing used in genetic epidemiology to detect gene-gene or gene-environment interactions, also known as epistasis, in high dimensional data. GENN approaches have previously been shown to be highly successful in a range of simulated and real case-control studies, and has recently been applied to quantitative traits. In the current study, we evaluate the potential of an application of GENN to quantitative traits (QTGENN) to a range of simulated genetic models. We demonstrate the power of the approach, and compare this power to more traditional linear regression analysis approaches. We find that the QTGENN approach has relatively high power to detect both single-locus models as well as several completely epistatic two-locus models, and favorably compares to the regression methods.

Categories and Subject Descriptors
I.5.0 [**Computing Methodologies**]: Pattern Recognition

General Terms: Performance

Keywords
Grammatical evolution, neural networks, genetic association, epistasis, interactions

1. INTRODUCTION
The decreasing cost of whole-genome genotyping has led to the rapidly exploding availability of genomic data for disease mapping in humans. The use of traditional statistical methodology has led to some enticing discoveries and confirmations [1]. However, analyzing this data on a gene-by-gene basis has not been successful in determining the etiology of many human diseases, such as diabetes and cardiovascular disease, despite repeated studies indicating their relatively high heritability (the proportion of trait variance due to genetic variation) [2].

GECCO'11, July 12–16, 2011, Dublin, Ireland.

One hypothesized reason that many association results have not revealed reliable predictors of disease across replication datasets is epistasis: interactions between multiple genes with little to no main effects that have not often been investigated [3]. The search for such interactions is a challenge. One of the key challenges is the actual statistical modeling of such interactions. Traditional statistical models can be limited in their ability to model such interactions due to the large number of empty contingency table cells in higher order interactions given the limited sample sizes of human genetic samples [3, 4].

Additionally, as genotyping technology has advances, and hundreds of thousands or millions of genetic variables are readily genotyped, variable selection (choosing the most predictive variables from the many genotyped) is an important challenge. Variable selection is difficult even in selecting univariate models, and the combinatorics of evaluating interactions expands the challenge exponentially when considering interactions. Whereas analyzing each single-locus model for a traditional regression analysis on a genome-wide scale is well within the capabilities of modern computers, the number of multi-locus models increases exponentially, rendering an exhaustive search of all models impractical as the number of loci surveyed increase from the thousands up to the millions of variants available on current-generation genome-wide genotyping arrays [5]. Besides this technical limitation, the statistical implications of multiple comparisons on the order of 10^{12} make this approach computationally infeasible.

The standard tool used for association analyses of quantitative traits in population-based cohorts is linear regression. This technique has a number of good attributes: it has solid theoretical grounding in statistics, is easily interpretable, and generally accepted by the research community [6]. While checking all multi-locus models is impractical (as stated earlier), variable selection approaches (backward, forward, etc) are often applied in an attempt to ameliorate this problem [6]. One popular method of searching for genetic models with multiple loci is forward stepwise selection [7]. In this procedure, loci are tested one at a time for main effects using regression. Loci are added at each step based on how much their presence improves the model. However, this approach is limited to constructing models with only main effects (while multiple loci may be included, their interaction terms are not).

In response to these challenges, machine-learning approaches are gaining in popularity in human genetics [8]. The development, optimization, and application of new data mining approaches is a

booming area of research, and a wide number of methods are being developed to simultaneously perform statistical modeling and variable selection tasks in genetic association studies.

One very promising area of research is the application of evolutionary algorithms to build a variety of classifiers for detecting gene-gene interactions in high throughput data. These methods have emerged as a viable alternative to exhaustive search approaches when the scale of data (in terms of number of input variables) is very large.

A number of evolutionary algorithms have been used to evolve a wide range of classifiers, but the use of grammatical evolution to evolve neural networks (GENN) [9] has shown particular success. GENN was originally developed for case-control data (with a binary trait) and was recently extended to quantitative traits [10, 11]. In the current study, we use a quantitative trait GENN (QTGENN) implementation similar to those described previously to investigate the power of the approach to detect quantitative trait associations with a broad range of effect sizes. Such power studies are an important stage in methods development, and are crucial to understand the potential of a new method for application to real data. Additionally, we compare the performance of the QTGENN approach to traditional regression approaches, in order to understand how these power results compare to traditional statistical approaches.

2. METHODS

2.1 Grammatical Evolution

Genetic algorithms (GA) [12] are a problem-solving technique where an initial population of many potential solutions to a given problem is generated, evaluated according to how well they solve the problem at hand, then 'breed' to produce the next generation. This process is iterated with the goal of having solutions converge on the correct solution. The breeding process can involve such actions as randomly mutating some of the solutions, or combining parts of multiple solutions to come up with new ones.

Grammatical Evolution (GE) is a form of a GA that allows the generation of computer programs using grammars. It is described in detail in [13, 14]. GE uses linear genomes and grammars to define populations for the GA process. In GE, each individual consists of a binary string divided into codons, where mutation takes place on individual bits along this binary string and crossover only takes place between codons. The relative fitness of a GE model (the "phenotype") is produced by translating the codons according to the grammar, and evaluated according to a specified optimization function. After fitness is evaluated, evolutionary operators are applied. GE is unique compared to other EC algorithms as it separates "genotype" from "phenotype" in the evolutionary process and allows greater genetic diversity within the population than other evolutionary algorithms [13, 14]. This process is analogous to the translation of RNA into a protein according to the amino acid code.

2.2 Neural Networks

Neural networks (NN) are a highly successful class of pattern recognition algorithms developed to model the basic functional unit of the brain, the neuron [15]. NNs were developed to model and capitalize on the parallel architecture of the human brain. This parallel architecture is a advantage over conventional computer programs, as they traditionally process data sequentially [16].

To model this capability, NN are used to construct a collection of simple analog processors in parallel to take an input pattern and generate an output signal [16]. In the case of genetic association studies, the inputs are genetic variants, and the output is a phenotypic value (such as disease status or a quantitative clinical variable like blood pressure, insulin levels, etc.) NN are a specific type of directed graph [17], consisting of nodes that represent processing elements, arcs that represent the connections of the nodes, and directionality on the arcs that represent the flow of information [18]. The nodes are arranged in layers such that the input layer receives the external pattern that is to be processed by the NN. Each node in the input layer is connected to one or more nodes in a hidden layer. In turn, nodes in the hidden layer are connected to either additional hidden layers or to an output node. The number and organization of inputs, nodes, and arcs is referred to as the architecture of a NN. Each connection within the NN has a weight associated with it. The external signal is conducted from the input layer, though the hidden layer(s) to the output layer, which is typically a single node. This output node then generates an output signal that can then be used to classify the original input pattern [18]. Figure 1 demonstrates the structure of a general NN.

NN are a highly appropriate model for human genetics studies for several reasons. Importantly, as the scale of data is rapidly expanding, NN are able to handle large quantities of data in reasonable computation time. Additionally, NN are universal function approximators so they should be able to approximate any type of genetic etiology that underlies phenotypic values. Finally, they are model-free in that no assumption has to be made about the genetic architecture that results in a particular phenotype, which is important when performing data-mining explorations of high dimensional data when little biology is known *a priori*.

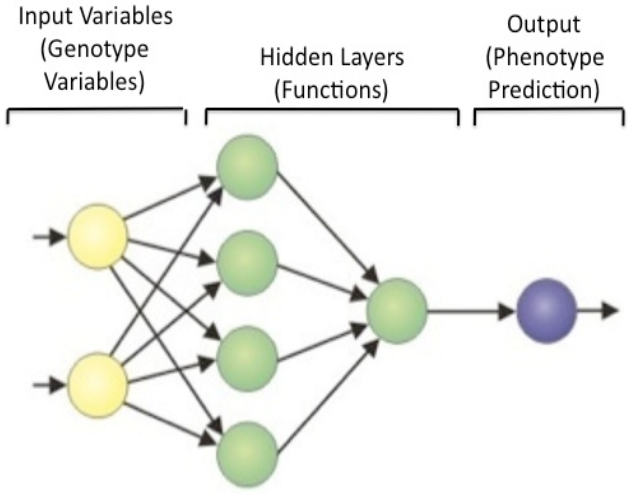

Figure 1. A feed-forward neural network with one input layer consisting of five nodes, two hidden layers with four and one node respectively, and one output layer (O). The connections between layers have associated connection strengths or weights.

Traditional application of NN models involve the implementation of a user-specified architecture (including variable selection), and

backwards selection optimization of weights between nodes. This traditional approach is highly reliant on these arbitrary parameter choices. This is a major concern for genetic association studies, as the proper variables and appropriate NN architecture is unclear [19]. To overcome these limitations, the application of evolutionary computation algorithms has emerged as a promising solution for evolving such models [20]. This approach has been successful in many fields [20], and has been successfully applied in association studies as well [21-23].

2.3 Grammatical Evolution Neural Networks (GENN)

An application of GE to evolve NN for genetic association data has previously been described in detail ([24]). This Grammatical Evolution Neural Network (GENN) approach was originally designed to select from a large number of potential input variables (typically categorical genetic variants, or single nucleotide polymorphisms (SNPs)) and associate them to a binary outcome variable (like case-control status).

Before analysis, the original genotype data (input variables) is recoded into dummy variables as proposed by [24]. This is done to remove assumptions of additivity in the genetic model by using a numeric encoding. Single-locus genotypes, encoded as 0, 1, 2 (representing three genotypes derived from a biallelic genetic locus), are re-encoded as two-variable linear contrasts, such that 0 becomes -1 and –1, 1 becomes 0 and 2, and 2 becomes 1 and -1.

GENN used GE to evolve every aspect of NN analysis. Details of the GENN process have been previously described [24]. The basic steps are as follows, shown in Figure 2.

Briefly, the data is split into ten different pieces, for ten-fold cross-validation (CV). 9/10 of the data are used to train the neural networks (NNs), and the remainder is used to test the final NNs. For each of the ten CV splits, the following analysis is carried out.

Step 1: Random bit strings are generated corresponding to valid neural networks (NNs) using sensible initialization [13, 14], to create an initial population of NNs.

Step 2: These bit strings are translated into NNs using the grammar.

Step 3: Each NN is evaluated on the training data, comparing the actual contents of the training data to the values predicted by the NN using the fitness metric.

Step 4: NNs that performed 'best' according to the fitness metric are chosen, and a new generation of solutions is constructed via reproduction, including random crossover and mutation of the bit strings.

Steps 2 through 4 are iterated until either fitness (one of several measures of classification accuracy for classical GENN) reaches 100%, or the maximum number of generations has been reached. At this point, the final population of NNs is evaluated using the fitness metric against the remaining 1/10 of the data held out as the testing set. The best performing of these NNs is used as the final model.

In an attempt to escape local minima in the search space, as well as take advantage of parallel computing facilities, steps 1 through 4 take place in multiple populations of NNs running on separate processors. The best individual from each population is periodically replicated to all other populations.

After this process is completed for all 10 cross-validation replicates, the variables in the final model of each run are inspected. Variables with the highest cross-validation consistency (i.e., they appear in the final models from most or all of the CV runs, more often than the other variables) are chosen for inclusion in the final model.

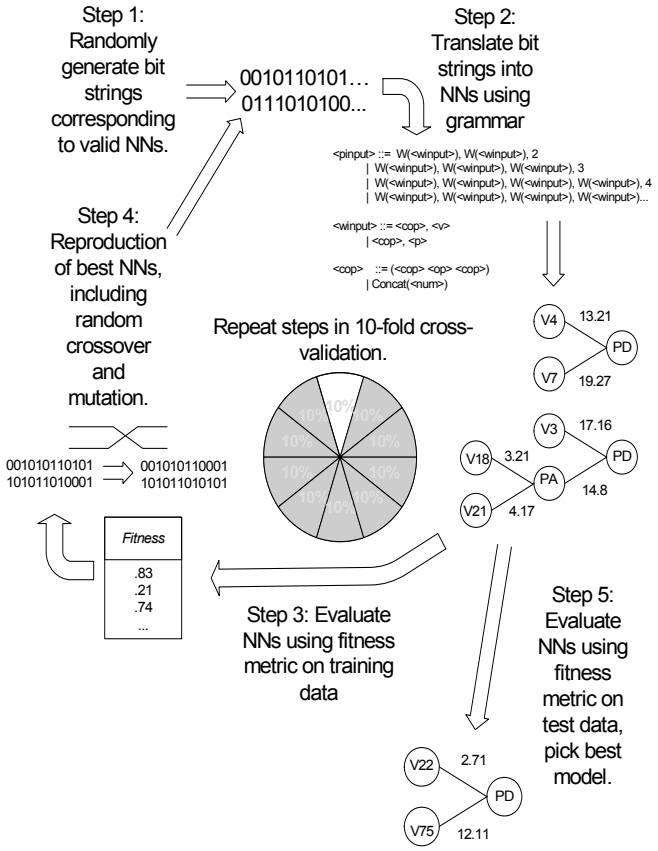

Figure 2. An overview of the GENN process that shows the six-step process of initialization, cross-validation, training, fitness evaluation using balanced error, natural selection (tournament) and testing – evaluating prediction error.

2.4 Quantitative GENN (QTGENN)

For this study, the standard balanced error fitness metric implemented for case-control studies was replaced with the R^2 value, calculated as the square of the correlation,

$$r = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\sqrt{[\Sigma(x-x)^2\,\Sigma(y-y)^2]}} \qquad (1)$$

where x_i and y_i are the actual and predicted trait values respectively for individual i. The predicted values y_i are the output of a neural network constructed from each individual's bit string according to the grammar.

This grammar modification allows for the evaluation of quantitative traits with GENN (QTGENN). Since there is not a

restriction of the NN models to binary outcomes, this extension of the grammar to quantitative traits is a natural extension.

The grammar's language contains symbols for arithmetic operators such as addition, subtraction, multiplication, and division, numeric constants, and more complex symbols allowing these to be joined together. It also includes input variables, which are read from the input data for each sample, and consist of genotypes (and possible environmental covariates). The exact grammar used in this implementation is explicitly included in the Appendix.

2.5 Linear Regression

To compare QTGENN against a widely used technique for genetic association, we used R (r-project.org [25]) to perform forward stepwise linear regression on each of the datasets. In this technique, linear regression models are constructed by starting from a null model, adding each locus to the model in turn, and re-evaluating. Decisions on whether to keep each locus or remove it at each step are based on the selection criterion – in this case, the Bayesian Information Criterion (BIC) [26]. Power for linear regression was defined as the percentage of replicates for each model/effect size for which analysis identified all of the causal loci.

As a positive control, explicit regression was used to determine how much genetic signal each model contained. In explicit regression, models are constructed which contain only the causal loci, their interactions, and the intercept. These models are then tested to determine the average r-squared value across all replicates, as well as the percentage of replicates for which each effect is significantly different from zero. If the explicit regression finds little to no significance in the known actual model, it would be unsurprising for other methods to perform poorly at discovering the correct model de novo.

2.6 Data Simulations

To test and compare the power of the QTGENN approach, a range of genetic models and effect sizes were simulated. We use penetrance functions to represent epistatic genetic models, where penetrance typically defines the probability of disease given a particular genotype combination by modeling the relationship between genetic variations and disease risk.

Since we are interested in quantitative traits instead of binary traits, penetrance functions were used not to assign disease status, but to assign individuals into two different phenotype distributions. Instead of assigning "case" status, the penetrance function is used to define the probability of an individual's assignment to the higher/upper trait distribution. The genetic variations modeled are single-nucleotide polymorphisms (SNPs) with 2 alleles (A and a) that result in three genotypes per SNP (AA, Aa, aa). Genotypes were generated according to Hardy-Weinberg proportions (in both models, p=q=0.5) [27, 28].

Both single-locus and multi-locus interaction models were simulated. Single-locus models with different genetic modes of inheritance were simulated (dominant and recessive). In the case of the dominant model, allele A is "dominant" to allele a, such that genotypes AA and Aa have the same probability of assignment to the upper (higher) trait distribution, and the genotype aa has a zero probability of being assigned to the upper distribution. In the case of the recessive model, the aa genotype has risk of being assigned to the upper distribution, and the AA and Aa genotypes are assigned to the lower distribution.

Canonical examples of 2-locus epistatic models, XOR and ZZ (described by [29]), were used to evaluate the potential of the method to detect "purely" epistatic effects, with no marginal main effects. Additional 2- and 3-locus epistatic models were also generated using the SimPen software [30]. SimPen uses a genetic algorithm to find penetrance functions with minimal main effects, based on user specified effect sizes (in terms of heritability and odds ratio) and controlling the minor allele frequencies of causal genes. These models were chosen such that the heritability of which distribution was chosen was relatively low (0.05 or 0.10), and with odds ratios that varied from 2 to 5. Heritability was calculated as discussed in [31].

Example penetrance functions for the XOR and ZZ functions are shown below in Table 1. The XOR model demonstrates an interaction effect in which high risk of assignment to the upper trait distribution is dependent on inheriting a heterozygous genotype (Aa) from one locus or a heterozygous genotype (Bb) from a second locus, but not both. In the ZZ model, high risk of assignment to the upper trait distribution is dependent on inheriting exactly two high-risk alleles (A and/or B) from two different loci. The additional penetrance functions used in the current simulation study are available on the following website: www4.stat.ncsu.edu/~motsinger, or directly from the authors by request.

Table 1. Example Epistatic Penetrance Functions, where the probability of assignment to the higher quantitative trait distribution is listed in the cells (the multilocus genotype combinations for two genes, A and B). Table A demonstrates the XOR function, while B demonstrates the ZZ model.

A. XOR Model, Heritability=5.26%

	BB	Bb	bb
AA	0	0.1	0
Aa	0.1	0	0.1
aa	0	0.1	0

B. ZZ Model I, Heritability=5.13%

	BB	Bb	bb
AA	0	0	0.1
Aa	0	0.05	0
aa	0.1	0	0

The final trait distribution in all datasets was thus a mixture of two normal distributions: one sampled from N(0,1), and the other from N(μ, 1), where μ took the value of 0.25, 0.5, 1.0, 2.0, or 3.0. This leads to a more easily interpretable range of varying model difficulty, from obvious (an effect size of 3) to almost non-existent (an effect size of 0.25), as can be seen by Figure 3. These extremely low effect sizes were used as a negative control – on datasets with so little signal, any analytical methods can be expected to perform poorly at determining the causal loci. This simulation approach has been described in detail in [32], and is outlined in Figure 3.

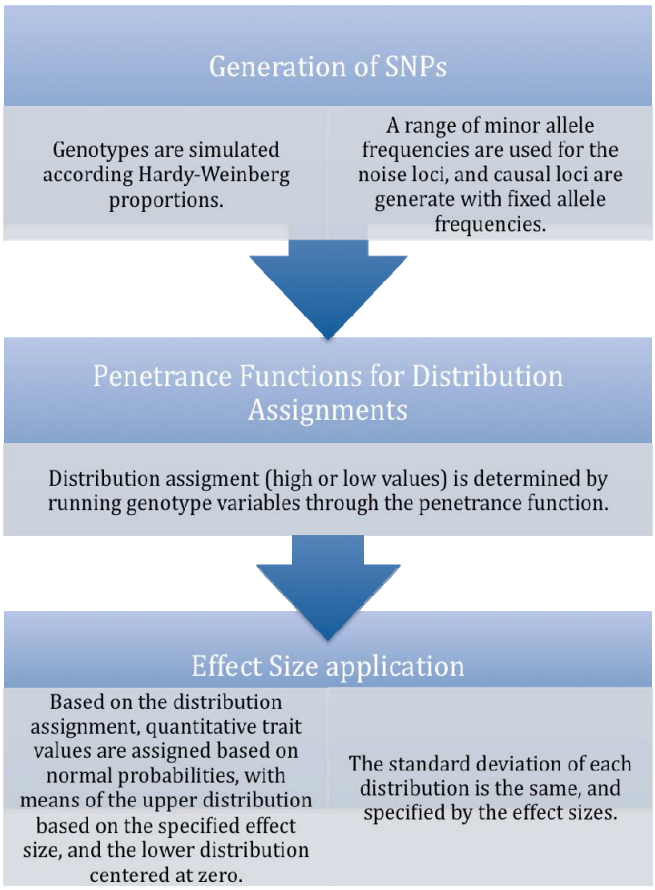

Figure 3. Workflow of the Data Simulation Process.

In total, sixteen different genetic models were simulated, with 5 effect sizes (shifts in means for the two distributions) for each. The models simulated are listed in Table 2. Each dataset contained 500 individuals sampled from the low distribution and 500 from the high distribution. Each dataset had 100 total loci, with one, two, or three of these being causal (depending on the genetic model) and the rest being unlinked to the trait (or each other). Allele frequencies were 0.5 for all loci. Each combination of genetic model and effect size was replicated 100 times, each of which was analyzed separately. Datasets with 100 loci are certainly not representative of the genome-wide data available today, but a power study using GWAS data is currently computationally infeasible. Figure 4 shows histograms of example trait distributions for two of the effect sizes simulated.

2.7 Data Analysis

QTGENN was executed with 10-fold cross-validation, 4 demes per run. Each deme was allowed to run for a maximum of 5000 generations, with each deme exchanging their best NN every 25 generations. Demes had a population of 250. Standard crossover was used, with a crossover rate of 0.9, and the mutation rate was 0.2. Minimum chromosome size was 50 codons, and maximum was 500. Tournament selection was used – in this method, individual NNs are chosen at random in groups of 3, and the best-performing one is selected for breeding in the EC process.

These values were chosen after a previous, smaller-scale simulation study (smaller in terms of number of models simulated,

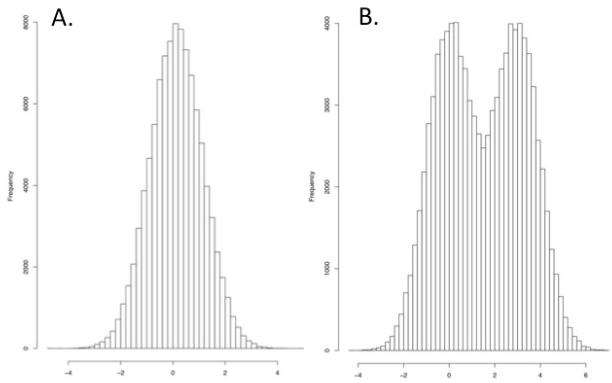

Figure 4. Examples of mixed trait distributions (frequency histograms) of model A, with effect sizes of 0.25 (A) and 3 (B).

Table 2. Summary of simulated models.
For special genetic models, the odds ratio is undefined because of a zero value in the penetrance function.

# Causal Loci	h^2	Odds Ratio (Special Genetic Model)
1	0.05	Undefined (Dominant)
1	0.05	Undefined (Recessive)
2	0.05	2
2	0.05	2.5
2	0.05	3
2	0.05	Undefined (XOR)
2	0.05	Undefined (ZZ)
2	0.10	3
2	0.10	4
2	0.10	5
3	0.05	2
3	0.05	2.5
3	0.05	3
3	0.10	3
3	0.10	4
3	0.10	5

not number of SNPs or individuals) parameter sweep experiment was performed on similar simulated data (results not shown) – observing varying performance of the method as different parameter values were tried. The fitness criterion used to evaluate each NN was the R^2 calculation described above. These parameters resulted in the highest power of the QTGENN method in these parameter sweeps.

QTGENN was implemented in C++, compiled using GCC 3.4.6 for 32-bit Linux, and executed on cluster nodes with Intel Xeon processors running RHEL 4. Average execution time of each cross-validation replicate on this system was 24 minutes.

Instructions for acquiring the source code can be found through the following website: http://www4.stat.ncsu.edu/~motsinger/.

Power for QTGENN was defined as the percentage of datasets (of the 100 replicates simulated for each model) analyzed in which the final model contained all causative loci at a higher cross-validation consistency (CVC) than any non-causal locus (with no false positive or false negative loci). For the step-wise linear regression analysis, power was defined as the percentage of times across the 100 replicates for each simulated model that regression selected the causal variable(s) in the final model (with no false positive and no false negative loci).

3. RESULTS

Table 3 contains a summary of QTGENN power over a range of disease models. QTGENN has increasing power to detect all causal loci in models with larger effect sizes, heritabilities, and odds ratios. Conversely, QTGENN has less power to detect models with increasing numbers of purely epistatic loci, with 3-locus models having relatively low power. Only when very large effect sizes are reached does the method show power above zero at all.

Table 3. QTGENN Power (%) (ES = Effect Size)

Model			ES 0.25	ES 0.5	ES 1.0	ES 2.0	ES 3.0
Loci	h^2	Odds Ratio (Special Genetic Model)					
1	0.05	Undefined (Dominant)	0.02	0.13	0.78	0.92	0.95
1	0.05	Undefined (Recessive)	0.22	0.99	1.0	1.0	1.0
2	0.05	2	0	0.04	0.51	1.0	0.99
2	0.05	2.5	0	0	0.08	0.52	0.61
2	0.05	3	0	0	0.24	0.73	0.85
2	0.05	Undefined (XOR)	0.03	0.47	1.0	1.0	1.0
2	0.05	Undefined (ZZ)	1.0	0.54	1.0	1.0	1.0
2	0.10	3	0	0.03	0.39	0.98	1.0
2	0.10	4	0	0.04	0.40	0.99	1.0
2	0.10	5	0	0.04	0.63	1.0	1.0
3	0.10	2	0	0	0	0	0
3	0.10	2.5	0	0	0	0.01	0.01
3	0.10	3	0	0	0	0	0
3	0.10	3	0	0	0	0.01	0.02
3	0.10	4	0	0	0	0.21	0.24
3	0.10	5	0	0	0	0.06	0.21

The results of the step-wise linear regression analyses are shown below in Table 4. These results show that the power of step-wise linear regression is higher than that of QTGENN for the single locus models, but is much lower for the epistatic models. Unsurprisingly, given the hierarchical nature of the step-wise variable selection procedure, step-wise linear regression has powers of zero for all the interactive models.

Because a multistep approach was taken for the data simulations, we calculated R^2 values for all the simulated models to have better context for the results of the QTGENN and step-wise regression results. Table 5 lists the average R^2 across the 100 datasets for each model. The correlation coefficients quantify the true amount of signal in the datasets generated.

Table 4. Stepwise Linear Regression Power (%) (ES = Effect Size)

Model			ES 0.25	ES 0.5	ES 1.0	ES 2.0	ES 3.0
Loci	h^2	Odds Ratio (Special Genetic Model)					
1	0.05	Undefined (Dominant)	0.02	0.26	0.99	1	1
1	0.05	Undefined (Recessive)	0.32	1	1	1	1
2	0.05	2	0	0	0	0	0
2	0.05	2.5	0	0	0	0	0
2	0.05	3	0	0	0	0	0
2	0.05	Undefined (XOR)	0	0	0	0	0
2	0.05	Undefined (ZZ)	0	0	0	0	0
2	0.10	3	0	0	0	0	0
2	0.10	4	0	0	0	0	0
2	0.10	5	0	0	0	0	0
3	0.05	2	0	0	0	0	0
3	0.05	2.5	0	0	0	0	0
3	0.05	3	0	0	0	0	0
3	0.10	3	0	0	0	0	0
3	0.10	4	0	0	0	0	0
3	0.10	5	0	0	0	0	0

4. DISCUSSION

The results of our simulations show several trends. For both QTGENN and regression, power increases with effect size and odds ratios. This is not surprising, since these cases represent models with stronger genetic effects that should be easier to find. We are encouraged by relatively high power of QTGENN to detect models for which explicit regression finds small R^2 values (0.05 - 0.10). We believe this indicates the promise of the method to detect epistatic models even in the absence of main effects, significant or otherwise.

The results of this study demonstrate that the QTGENN approach is promising for detecting genetic risk factors in association studies, and provides initial results to understand the power of this analytical approach in a wide range of models. While these results are exciting, there are many more questions that should be followed up in the continual development of this method.

There are many aspects of the implementation itself that might be improved. For example, the grammar used in this paper allowed the NNs to extrapolate from the data – i.e., trait values outside the observed range could be predicted. Although this provides maximum generality, it may reduce QTGENN power, or increase computational time required for convergence. In follow up studies we would like to experiment with additional model

components, including new protected operators which limit predictive NN values to be in the same range as the training dataset. More complex non-terminals could also be envisioned, such as logarithms, exponentials, and trigonometric functions. Although the current grammar allows approximations of these functions to evolve, having them available as building blocks of the NN may allow more complex behavior to evolve in fewer generations.

Table 5. Variance explained by Causal Loci when Explicitly Modeled (R^2). (ES = Effect Size)

Model			ES 0.25	ES 0.5	ES 1.0	ES 2.0	ES 3.0
Loci	h^2	Odds Ratio (Special Genetic Model)					
1	0.05	Undefined (Dominant)	0	0.01	0.03	0.07	0.1
1	0.05	Undefined (Recessive)	0.01	0.03	0.12	0.3	0.43
2	0.05	2	0	0.01	0.02	0.06	0.08
2	0.05	2.5	0	0.01	0.02	0.05	0.06
2	0.05	3	0	0.01	0.02	0.05	0.07
2	0.05	Undefined (XOR)	0.01	0.02	0.07	0.18	0.25
2	0.05	Undefined (ZZ)	0.01	0.03	0.1	0.24	0.34
2	0.10	3	0.01	0.01	0.03	0.08	0.1
2	0.10	4	0.01	0.01	0.04	0.09	0.12
2	0.10	5	0.01	0.02	0.04	0.1	0.13
3	0.05	2	0.02	0.02	0.04	0.05	0.07
3	0.05	2.5	0.02	0.02	0.04	0.06	0.08
3	0.05	3	0.02	0.03	0.04	0.07	0.09
3	0.10	3	0.02	0.03	0.05	0.08	0.11
3	0.10	4	0.02	0.03	0.05	0.1	0.14
3	0.10	5	0.02	0.03	0.06	0.11	0.15

Optimization of the implementation itself will also be important. While the methodology is promising on these smaller scale datasets, in thinking towards scaling the methodology to real data, with hundreds of thousands of variables, the implementation will need to be improved.

In addition, we plan to follow up these results with comparisons to other computational methods that are designed to detect gene-gene interactions in human genetic association studies, such as the Recursive Partitioning Method (RPM) [32] and generalized Multifactor Dimensionality Reduction (GMDR) [33]. Such comparisons would be an important next step in understanding the niche of GE optimized neural networks for detecting quantitative traits.

Finally, because of the flexibility of NNs as classifiers, the method could readily be expanable to quantitative inputs as well as outputs. This would allow for the application of the QTGENN method to a wider range of genomic, environmental, and clinical input variable types, such as gene expression data, etc.

5. ACKNOWLEDGMENTS

The research presented in this work was funded by NIEHS 2 T32 ES007329.

The authors would like to thank others that have contributed to the development of the QTGENN approach, including Marylyn Ritchie, Scott Dudek, Lance Hahn, and Bill White.

6. APPENDIX

The grammar used in the QTGENN implementation is below, where N describes the nonterminals, T describes the terminals, and S represents the start codon.

N = {p,pn,pinput,wt,winput,cop,op,v,num, dig}

T = [W,*,+,/,-,Concat,.,V1-200]

S = p

<p> ::= <op>(<pinput>)

<pn> ::= PA
| PS
| PM
| PD

<pinput> ::= W(<winput>), W(<winput>), 2
| W(<winput>), W(<winput>), W(<winput>), 37.
| W(<winput>), W(<winput>), W(<winput>), W(<winput>), 4
| W(<winput>), W(<winput>), W(<winput>), W(<winput>), W(<winput>), 5

<winput> ::= <cop>, <v>
| <cop>, <p>

<cop> ::= (<cop> <op> <cop>)
| Concat(<num>)

<op> ::= +
| -
| *
| /

<num> ::= . 0 0 <dig> <dig> 5

<dig> ::= 0
| 1
| 2
| 3
| 4
| 5
| 6
| 7
| 8
| 9

<v> ::= V1-200

8. REFERENCES

[1]. Altshuler, D., M.J. Daly, and E.S. Lander, *Genetic mapping in human disease.* Science, 2008. 322(5903): p. 881-8.

[2] Eichler, E.E., et al., *Missing heritability and strategies for finding the underlying causes of complex disease.* Nat Rev Genet, 2010. 11(6): p. 446-50.

[3] Moore, J.H., *The ubiquitous nature of epistasis in determining susceptibility to common human diseases.* Hum Hered, 2003. 56(1-3): p. 73-82.

[4] Cordell, H.J., *Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans.* Hum Mol Genet, 2002. 11(20): p. 2463-8.

[5] Moore, J.H. and M.D. Ritchie, *STUDENTJAMA. The challenges of whole-genome approaches to common diseases.* JAMA, 2004. 291(13): p. 1642-3.

[6] Motsinger, A.A., M.D. Ritchie, and D.M. Reif, *Novel methods for detecting epistasis in pharmacogenomics studies.* Pharmacogenomics, 2007. 8(9): p. 1229-41.

[7] Hocking, R.R., *The Analysis and Selection of Variables in Linear Regression.* Biometrics, 1976. 32.

[8] Moore, J.H. and S.M. Williams, *New strategies for identifying gene-gene interactions in hypertension.* Ann Med, 2002. 34(2): p. 88-95.

[9] Motsinger-Reif, A.A., et al., *Comparison of approaches for machine-learning optimization of neural networks for detecting gene-gene interactions in genetic epidemiology.* Genet Epidemiol, 2008.

[10] Holzinger, E.R., et al., *Initialization Parameter Sweep in ATHENA: Optimizing Neural Networks for Detecting Gene-Gene Interactions in the Presence of Small Main Effects.* Genet Evol Comput Conf, 2010. 12: p. 203-210.

[11] Turner, S.D., S.M. Dudek, and M.D. Ritchie, *ATHENA: A knowledge-based hybrid backpropagation-grammatical evolution neural network algorithm for discovering epistasis among quantitative trait Loci.* BioData Min, 2010. 3(1): p. 5.

[12] Quinlan, J.R., *Programs for Machine Learning.* 1993: Morgan Kaufmann Publishers.

[13] O'Neill, M. and C. Ryan, *Grammatical Evolution*. 2001, Boston: Kluwer Academic Publishers.

[14] O'Neill, M. and C. Ryan, *Grammatical Evolution: Evolutionary automatic programming in an arbitrary language.* 2003, Boston: Kluwer Academic Publishers.

[15] Mandic, D. and J. Chambers, *Recurrent Neural Networks for Prediction: Architectures, Learning algorithms and Stability.* 2001: Wiley.

[16] Reilly, D.L., L.N. Cooper, and C. Elbaum, *A Neural Model for Category Learning.* Biological Cybernetics, 1982. 45: p. 6.

[17] Minsky, M. and S. Papert, *An Introduction to Computational Geometry.* 1969: MIT Press.

[18] Skapura, D., *Building Neural Networks.* 1995: Wiley.

[19] Motsinger-Reif, A.A. and M.D. Ritchie, *Neural networks for genetic epidemiology: past, present, and future.* BioData Min, 2008. 1(1): p. 3.

[20] Vonk, E., L.C. Jain, and R.P. Johnson, *Automatic Generation of Neural Network Architecture Using Evolutionary Computation*. Advances in Fuzzy Systems - Application and Theory. Vol. 14. 1997: World Scientific.

[21] Ritchie, M.D., et al., *Genetic Programming Neural Networks: A Powerful Bioinformatics Tool for Human Genetics.* Appl Soft Comput, 2007. 7(1): p. 471-479.

[22] Motsinger, A.A., et al., *Understanding the Evolutionary Process of Grammatical Evolution Neural Networks for Feature Selection in Genetic Epidemiology.* Proc IEEE Symp Comput Intell Bioinforma Comput Biol, 2006. 2006: p. 1-8.

[23] Motsinger, A.A., et al., *GPNN: power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease.* BMC Bioinformatics, 2006. 7: p. 39.

[24] Motsinger, A.A., et al., *Comparison of Neural Network Optimization Approaches for Studies of Human Genetics.* Lect Notes Comput Sci, 2006. 3907: p. 103-114.

[25] R Development Team. *R: A language and environment for statistical computing, reference index version 2.2.1.* 2005; Available from: URL http://www.R-project.org.

[26] Schwarz, G.E., *Estimating the dimension of a model.* Annals of Statistics, 1978. 6(2): p. 4.

[27] Hardy, G.H., *Mendelian Proportions in a Mixed Population.* Science, 1908. 28(706): p. 2.

[28] Weinberg, W., *Über den Nachweis der Vererbung beim Menschen.* Jahreshefte des Vereins für vaterländische Naturkunde in Württemberg, 1908. 64(368): p. 14.

[29] Li, W. and J. Reich, *A complete enumeration and classification of two-locus disease models.* Hum Hered, 2000. 50(6): p. 334-49.

[30] Moore, J.H., et al. *Application of genetic algorithms to the discovery of complex models for simulation studies in human genetics.* . in *Genetic and Evolutionary Algorithm Conference.* 2002: Morgan Kaufman Publishers.

[31] Culverhouse, R., et al., *A perspective on epistasis: limits of models displaying no main effect.* Am J Hum Genet, 2002. 70(2): p. 461-71.

[32] Culverhouse, R., T. Klein, and W. Shannon, *Detecting epistatic interactions contributing to quantitative traits.* Genet Epidemiol, 2004. 27(2): p. 141-52.

[33] Lou, X.Y., et al., *A combinatorial approach to detecting gene-gene and gene-environment interactions in family studies.* Am J Hum Genet, 2008. 83(4): p. 457-67.