Linkage Disequilibrium in Genetic Association Studies Improves the Performance of Grammatical Evolution Neural Networks

Alison A. Motsinger[1], David M. Reif[2], Theresa J. Fanelli[1], Anna C. Davis[1], and Marylyn D. Ritchie[1]
motsinger@chgr.mc.vanderbilt.edu

[1]Center for Human Genetics Research, Department of
Molecular Physiology & Biophysics, Vanderbilt University,
Nashville, TN, USA 37232

[2]National Center for Computational Toxicology,
Environmental Protection Agency, Research Triangle Park,
NC, USA 27711

Abstract

One of the most important goals in genetic epidemiology is the identification of genetic factors/features that predict complex diseases. The ubiquitous nature of gene-gene interactions in the underlying etiology of common diseases creates an important analytical challenge, spurring the introduction of novel, computational approaches. One such method is a grammatical evolution neural network (GENN) approach. GENN has been shown to have high power to detect such interactions in simulation studies, but previous studies have ignored an important feature of most genetic data: linkage disequilibrium (LD). LD describes the non-random association of alleles not necessarily on the same chromosome. This results in strong correlation between variables in a dataset, which can complicate analysis. In the current study, data simulations with a range of LD patterns are used to assess the impact of such correlated variables on the performance of GENN. Our results show that not only do patterns of strong LD not decrease the power of GENN to detect genetic associations, they actually increase its power.

Keywords
Grammatical evolution, neural networks, linkage disequilibrium, complex disease, gene-gene interactions

I. INTRODUCTION

One of the most important goals of genetic epidemiology is the identification and characterization of genetic factors associated with common, complex diseases [1,2]. The complex etiology assumed to predict such diseases presents an enormous analytical challenge. As genotyping technologies advance and the scale of genetic association studies exponentially increases, variable selection has become a salient problem. From thousands of variables, an analytical approach must identify the appropriate outcome-associated variables (likely involving non-linear interactive effects) and have power to ascribe statistical significance to true-positive genetic models. These analytical demands have prompted the development of a number of novel statistical and computational methods [3-8].

One such computational approach is a neural network (NN). The NN approach is a commonly used pattern recognition technique for data mining. NNs have been successful in a variety of fields, though they have met with mixed success in genetic epidemiology [9-14]. Unsuccessful applications may be attributed to the paramount importance of choosing the correct NN architecture for each individual dataset. The (often *a priori*) arrangement of nodes and their interconnectivity largely determines the success of NN for a specific problem, and inappropriate architecture can doom a NN to failure. In order to evolve appropriate NN architectures for multiple contexts, various machine learning methods have been combined with NNs in many fields [15].

Recently, a Grammatical Evolution neural network (GENN) strategy was introduced to detect single-locus and gene-gene interactions that predict common, complex disease [16]. GENN optimizes inputs from a large pool of variables, synaptic weights between connections, and the architecture of the network for data at hand. In so doing, GENN automatically selects the genetic variables most predictive of the disease under study.

The GENN method has been highly successful thus far. Previous studies compared GENN to other NN applications and found that it outperforms a traditional back-propagation NN strategy and a random search NN strategy [16]. Additionally, GENN has outperformed a genetic programming neural network approach in large datasets [17]. GENN has demonstrated high power to detect gene-gene and gene-environment interactions across a wide range of genetic models [17]. GENN has also replicated the findings of more traditional analytical approaches in detecting epistatic interactions real data applications in the immunogenetics of HIV [17] and age-related macular degeneration [18].

Studies of GENN's evolutionary process have provided insight into the mechanisms of GENN's success in finding purely epistatic models [19]. Given a high-dimensional dataset, GENN builds very large initial models containing both functional and non-functional variables. Over subsequent generations, the noise variables are then pruned out of the model [19].

These initial successes are promising, but previous simulation studies have ignored one important aspect of genetic data: linkage disequilibrium (LD). LD describes the non-random association of alleles. LD is characterized by combinations of genetic markers (normally alleles) that occur more or less frequently in a population than would be expected from a random formation of haplotypes—a statistically associated set of single nucleotide polymorphisms (SNPs) on a single chromatid—from alleles based on their frequencies. The presence of LD can be useful in genetic association studies. If all polymorphisms were independent at the population level, association studies would have to examine every variant. Genetic variants in tight LD allow important savings in terms of time, money, and computation, as fewer variants need to be examined. In small-scale candidate gene studies, SNPs that determine the status of nearby polymorphisms, typically on a short chromosome segment, are chosen to avoid the waste of genotyping resources and eliminate the analytical considerations of highly correlated genotypes. However, the use of tagging SNPs is controversial, as both the size and character of haplotype blocks varies by population, and the current resources for selecting such SNPs are constructed from populations of limited diversity [20].

As the scale of genetic association studies rapidly increases with advances in genotyping technology, the selection of such "tagging" SNPs will no longer be necessary. Instead, full genome screens are becoming commonplace. This means that the inter-correlations between genetic variables due to LD must be considered at the analysis stage. The resultant high-dimensional variable selection problem of evaluating the relative singular and combinatoric/epistatic influences of correlated independent variables (genotypes) on a single dependent variable (disease status) presents a considerable challenge for traditional statistical approaches. The difficulty in dealing with correlated predictor (independent) variables has been recognized in fields such as econometrics and psychology under the label "multicollinearity" [21, 22]. Stated simply, whenever the correlation between two or more variables is high, the sampling error of the partial slopes and partial correlation coefficients will be quite large. As a result there will be a number of different combinations of regression coefficients, and hence partial correlations, which give almost equally good fittings to the empirical data.

The impact of such "multicollinearity" is well-studied for traditional statistics [21, 22], but is less well characterized for novel, machine learning approaches. In the current study, we simulate data with varying patterns of LD to see if increasing inter-correlations between input noise variables affects the power of GENN to detect single-locus and gene-gene interactions associated with disease. We use a complex data simulation strategy to generate realistic patterns of LD, and then insert disease models into these backgrounds. We simulate single-locus and gene-gene interaction models with minimal effect sizes to test the lower limits of GENN to detect genetic associations in the presence of surrounding noise variables in LD. Our results demonstrate that not only is GENN robust to inter-correlation between variables, strong background patterns of LD actually improves the power of the method.

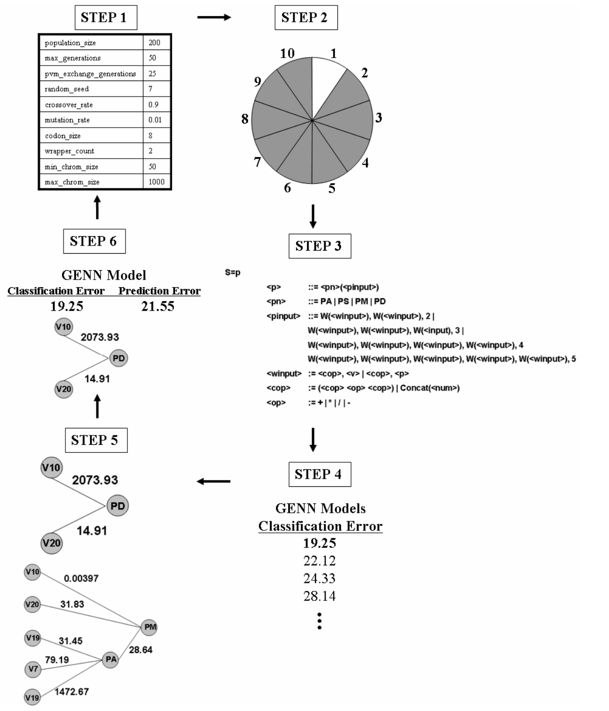
II. Methods

A. Grammatical Evolution Neural Networks (GENN)

Grammatical Evolution (GE) is a type of evolutionary computation that allows the generation of computer programs using grammars [23,24]. Populations are made of linear genomes, where individuals consist of a binary genome divided into codons. Evolutionary operations, such as crossover and mutation, take place at the level of the binary string, much like a typical genetic algorithm (GA). Individual genomes are translated into a functional NN, which can then be evaluated for fitness. Evolutionary operators are then applied to create subsequent generations. GE separates genotype from phenotype by using the grammar to map a NN.

Details of GE can be found in O'Neill and Ryan [23], with only salient features described here. Unlike the functions and terminals used in genetic programming [25], GE uses a Backus-Naur Form (BNF) grammar to generate code [24]. The grammar is used in a genotype to phenotype mapping process which produces a program from the genotypic binary string.

The steps of GENN have been previously described in detail [16,17]. First, GENN parameters must be initialized in the configuration file, including mutation rate, crossover rate, and number of generations. Details of the configuration file can be found in [17]. Second, the data are divided into 10 equal parts for 10-fold cross-validation. 9/10 of the data is used for training, and later the other 1/10 of the data is used to evaluate the predictive ability of the model developed during training. Third, an initial population of random solutions is generated to begin the training process. Sensible initialization is used to guarantee that the initial population contains only functional NN [24]. Fourth, each individual genome is translated into a NN according to the rules of the grammar. Each NN is evaluated on the training set and its fitness recorded. Fifth, the best solutions are selected for crossover and reproduction using user-specified proportions. The new generation (created by a selection technique specified in the configuration file) begins the cycle again. This continues until some stopping criterion is met, a classification error of zero is found, or a limit on the number of generations is reached. An optimal solution is identified after each generation. At the end of GENN evolution, the overall best solution is selected as the optimal NN. Sixth, this best GENN model is tested on the 1/10 of the data left out to estimate the prediction error of the model. Steps two through six are performed ten times using a different 9/10 of



population_size	200
max_generations	50
pvm_exchange_generations	25
random_seed	7
crossover_rate	0.9
mutation_rate	0.01
codon_size	8
wrapper_count	2
min_chrom_size	50
max_chrom_size	1000

```
S=p
<p>        ::= <pn>(<pinput>)
<pn>       ::= PA | PS | PM | PD
<pinput>   ::= W(<winput>), W(<winput>), 2 |
           W(<winput>), W(<winput>), W(<input), 3 |
           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>       := + | * | / | -
```

Figure 1. An overview of the GENN method. The steps correspond to the description of the method in Section 2.1.

the data for training and 1/10 of the data for testing. Figure 1 shows an overview of the GENN algorithm.

GENN uses GE to optimize inputs, architecture, and weights of a NN. The grammar used is available in [17] or from the authors. The GA used to evolve the binary string that is transcribed into a NN has the following parameters in the current implementation: crossover rate = 0.9, mutation = 0.01, population = 200, max generations = 200, codon size = 8, GE wrapping count = 2, min chromosome size (number of codons) = 50, max chromosome size = 1000, selection = tournament, and sensible initialization depth = 10. The island model of parallelization is used, where the best individual is passed to each of the other processes after every 25 generations [26], to prevent stalling in local fitness minima. The genome was derived from GAlib (version 2.4.5), and a typical GA one-point crossover of linear chromosomes is used [27].

GENN is optimized using a training set of data (for each of *k* cross-validation intervals), and a subset of the data is left out as a test set to evaluate the final solution and prevent over-fitting. Classification error simply refers to the number of individuals misclassified by the model in the training set, divided by the total number of individuals in that set. Prediction error refers to the number of samples in the test dataset that are incorrectly classified using the GENN model generated during training. For each cross-validation interval, a best model is chosen based on lowest classification error of all models evaluated for that interval—resulting in 10 models. A classification error and prediction error are recorded for each of the models and a cross-validation consistency is calculated to determine those variables which have a strong disease-association signal across divisions of the data. Cross-validation consistency summarizes the number of times a particular variable(s) is present in the best GENN model for each of the ten cross-validation data splits. The higher the cross-validation consistency is, the stronger the estimated generalizability for the model. The locus/loci with the highest cross-validation consistency is/are chosen as the final model.

B. Data Simulation

For the purposes of the current study, we simulated case-control data with a variety of single-locus and gene-gene interactive disease models where the functional loci (variables) are single nucleotide polymorphisms (SNPs). In order to simulate data with realistic patterns of LD, we first generated genetic backgrounds having different LD patterns and then inserted disease models into these backgrounds.

To generate the background patterns of LD on which to insert disease models, we used a novel data simulation software package called genomeSIM [28]. genomeSIM allows for the simulation of large-scale genomic data in population based case-control samples. It is a forward-time population simulation algorithm that allows the user to specify many evolutionary parameters and control evolutionary processes. The algorithm implemented in genomeSim is described below. Further details of this software can be found in [28].

In the first step, genomeSIM establishes the genome based on the parameters passed to it. The user specifies the number of SNPs per gene and the total number of genes in the genome. The simulator randomly determines the number of SNPs per gene based on the minimum and maximum parameters. The simulator then randomly determines the recombination fraction between adjacent SNPs within each gene based on maximum and minimum recombination fraction parameters. All recombination fractions (for each SNP) are random and independent. SNPs are unlinked across genes. Finally, the allele frequencies are randomly set for each SNP based on preset maximum and minimum allele frequency parameters. When the minimum is set equal to the maximum, the values across the simulated genome will be identical. In step 2 genomeSIM generates an initial population based on the genome established in the previous step. Each individual in the population has two binary chromosomes. For each SNP in the genome, the simulator randomly assigns an allele to each chromosome based on the allele frequencies of the SNP. The dual chromosome representation allows for an efficient representation of the genome and for crossover between chromosomes during the mating process. The genotype at any SNP can be determined simply by adding the values of the two chromosomes at that position. This initial population forms the basis for the second generation in the simulation. For each cross, two individuals are randomly selected with replacement to be the parents. Each parent contributes one haploid genome to a child in the next generation. genomeSIM creates the gametic genotype by recombining the parent's chromosomes. The total number of individuals in each population (across generations) is constant. During "mating", crossover occurs based on the recombination frequencies at each SNP. genomeSIM continues producing generations for the number specified. An overview of the genomeSIM algorithm is shown in Figure 2.

The evolutionary parameters used in the current simulations are shown in Table 1. It is important to note that these parameters only apply to the data simulations, and not to the GENN analysis. For the different backgrounds simulated, all parameters were held constant except for the number of generations. The generations were varied to produce differing patterns of LD. The parameters chosen for the current simulation are based on unpublished optimizations of the genomeSIM software. Generally, as the number of generations increases, the number and size of LD blocks (haplotype blocks) also increases, until a certain point at which the blocks deteriorate. Four numbers of generations were chosen: 1 (as a negative control, generating no LD), 100, 500, and 1000. Haploview [29] software was used to visualize the patterns of LD under each of the four conditions, shown in Figures 2 through 5. The plots can be read similarly to a correlation matrix, where darker (red) shading indicates stronger LD. As these plots show, the single generation background has essentially no LD. After 100 generations, blocks of LD are beginning to form. By 500 generations, there are several strong LD blocks. At 1000 generations, there are fewer LD blocks, but the blocks are larger. Figures 3 through 6 represent the four LD structure "backgrounds" that the simulated disease models were inserted into. Each background contained a total of 100 SNPs, with different patterns of LD generated by genomeSIM.

The underlying etiology of common diseases is presumed to be highly complex—including multiple single-locus risk factors as well as gene-gene and gene-environment interactions (known as epistasis) [30-32].

Disease models were simulated according to penetrance functions, where penetrance defines the probability of disease given a particular genotype combination by modeling the relationship between genetic variations and disease risk. Both single-locus and gene-gene interactive models were simulated. Single-locus effects were simulated under three main types of genetic models: 1. Dominant models, where disease risk was associated with having at least one copy of a dominant risk allele, 2. Recessive models, where the homozygous recessive genotype conferred disease risk, and 3. Additive models, where disease risk increased with an increasing number of risk alleles. Epistatic models were discovered using software described in [33]. These models are purely epistatic, where no one gene exhibits an independent main effect on case-control status. Models lacking such marginal main effects are appropriate for the goals of this study because they challenge the method to find gene-gene interactions in a complex dataset. All penetrance functions used are available from the authors upon request.

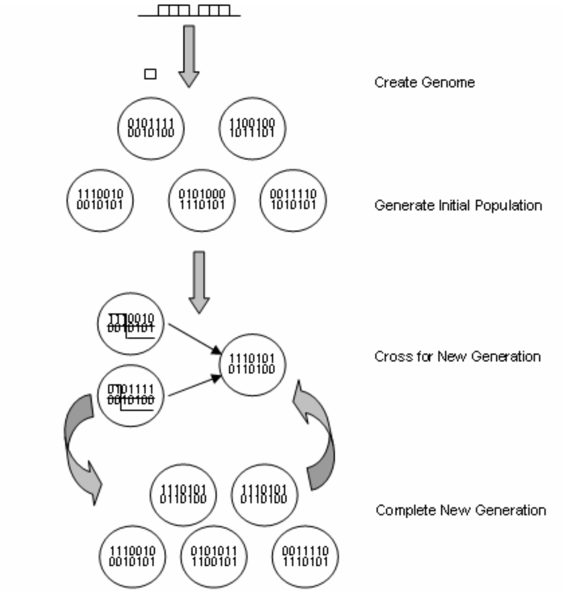


Figure 2. Overview of the genomeSIM algorithm (adapted from [28]). After the genome is constructed, an initial population of individuals is created and individuals cross by contributing one chromosome each to the offspring. These crosses create the next generation and the process repeats until the specified number of generations has occurred. In the last generation, the genotypes for the individual are produced by summing the chromosomes at each position.

TABLE 1.
Parameters for genomeSIM generation of genetic backgrounds

Population size	1000
Total SNPs	1000
Genes	10
SNPs per gene	10
Generations	1, 100, 500, 1000
Minimum recombination between SNPs	0.0001
Maximum recombination between SNPs	0.0001
Minimum minor allele frequency	0.05
Maximum minor allele frequency	0.5

A range of effect sizes were simulated to test the lower limits of GENN to detect disease-associated loci. Effect sizes were measured as odds ratios. For single-locus models (for each genetic model), the following odds ratios were simulated: 1.25, 1.5, 1.75, and 2.0. For the epistatic models, the following odds ratios were simulated: 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, and 3.0. The minor allele frequency for all models was 0.5. For all models with an odds ratio less than or equal to 2.0, the heritability (proportion of the trait due to genetics)

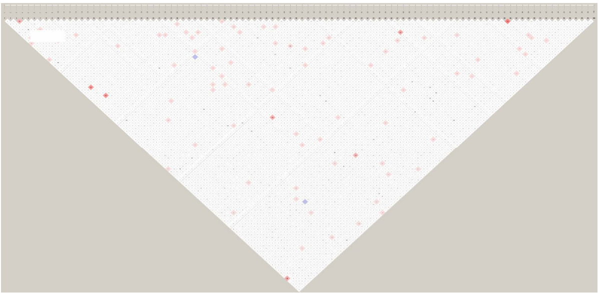


Figure 3. LD plot after 1 generation

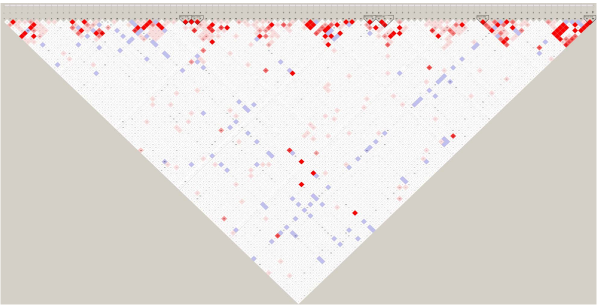


Figure 4. LD plot after 100 generations

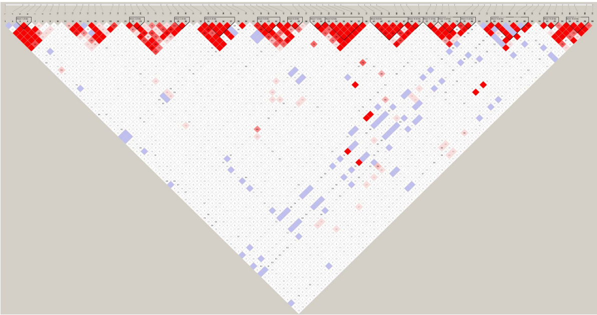


Figure 5. LD plot after 500 generations

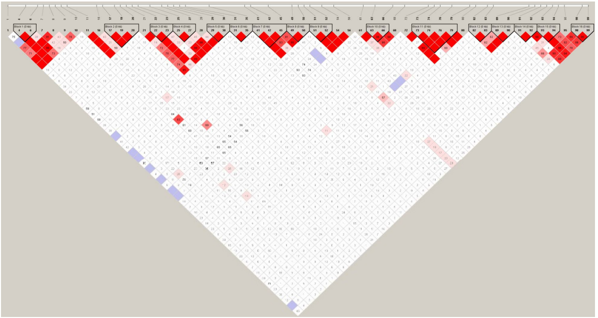


Figure 6. LD Plot after 1000 generation

of that model was ~1%. For odds ratios above 2.0, the heritability was ~5%. The biological relevance of these models is unknown, but in terms of analytical difficulty, they represent "worst case scenarios" in the genetic architecture of common, complex disease.

To create the final datasets having disease models imposed on extant patterns of LD, functional genetic attributes were added to the LD backgrounds and a case-control status was assigned according to the range of models described previously. Details of the simulation procedure can be found in [34]. Functional loci were inserted in the middle of LD blocks, using the same allele frequencies observed in the "background". This maintains all inter-correlation patterns between noise variables, but may break up any correlation between the functional locus/loci and the surrounding SNPs. Because the same allele frequencies were used for the functional variables and the "background" where they were inserted, the functional variable(s) will be in LD with the background by chance.

Each genetic model was inserted into each of the LD backgrounds, resulting in a total of 80 models. The two-stage data simulation approach described in the paper was used because the generation of LD in datasets is stochastic with genomeSIM. GenomeSIM was used to create 4 different genetic backgrounds and then disease models were inserted into those backgrounds. In this way, GENN's performance for disease models could be compared in the context of identical backgrounds of LD. The four different LD backgrounds were generated so that the performance of GENN could be compared for different background patterns of LD in the context of identical disease models. For each model, 100 replicates were produced for a total of 8000 datasets. Each dataset contained 500 cases and 500 controls, with genotype information at 100 SNPs (one or two of which are functional—depending on the particular disease model). Dummy variable encoding was used for each dataset, where *n-1* dummy variables were used for *n* levels [35].

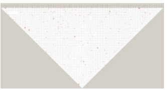
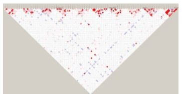
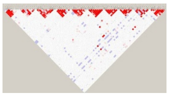
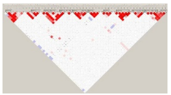
C. *Data Analysis*

GENN was used to analyze all simulated datasets. The configuration parameter settings were as follows: 10 demes, migration every 25 generations, population size of 200 per deme, 200 generations, crossover rate of 0.9, and a reproduction rate of 0.1. These parameter choices have previously been shown to be effective in datasets of this size [17]. Cross-validation consistency was used for final model selection, as described above. For a complete description of GENN configuration parameters, see [17].

Power for all analyses is reported as the number of times the algorithm correctly identified the correct functional loci with no false positive loci over 100 datasets. This strict definition of power is appropriate for the current study because we are interested in the power of GENN to find the associated/functional simulated loci, rather than any in strong LD by chance in the simulation. If either one or both of the dummy variables representing a single SNP was selected, that locus was considered present in the model.

Because GENN has consistently outperformed a random search NN strategy in SNP datasets of the same number of SNPs and number of individuals [16, 19], the current study does not include a comparison to a random search.

TABLE II.
GENN results for all genetic models and LD backgrounds.

Number of Functional Loci	Genetic Model	OR	Power (%)			
			1 Generation 	100 Generations 	500 Generations 	1000 Generations 
1	Dominant	1.25	4	5	21	15
		1.5	25	20	49	36
		1.75	56	43	74	65
		2.0	79	70	97	91
	Recessive	1.25	8	11	21	17
		1.5	48	45	70	70
		1.75	85	73	94	84
		2.0	97	99	100	98
	Additive	1.25	36	29	52	45
		1.5	92	92	97	95
		1.75	99	97	99	98
		2.0	98	96	99	98
2	Epistatic	1.25	5	4	24	19
		1.5	4	3	25	20
		1.75	2	3	22	19
		2.0	21	13	42	29
		2.25	75	84	91	89
		2.5	81	92	99	98
		2.75	100	100	100	100
		3.0	100	100	100	100

III. Results

Table 2 shows the results for all disease models, for all LD backgrounds. Several trends are readily apparent. First, as would be expected, as the effect size increases, so does the power of GENN to detect that effect. Also, as expected, the power to detect a single-locus effect is higher than the power to detect a two-locus effect with the same odds ratio. This is not surprising since it is generally more challenging for any statistical method to detect interactions [1, 2]. Additionally, since GENN relies on a machine learning strategy, in order to detect purely epistatic interactions variables that display no independent main effects must randomly both be included in a model at some point in the search process. After both variables are included, it can prune away noise variables, but this random joint-inclusion is a necessary step. The chances are much higher that a single locus is randomly included in a model than for two loci to be simultaneously included, contributing to the higher power to detect main effect models.

For the purposes of this study, the most important trend is related to the varied LD backgrounds. Across all disease models, GENN has the highest power in the context of the LD simulation run for 500 generations. This LD background has the highest number of strong LD blocks. Generally, across all genetic models, as the number of LD blocks increase in the background, the power increases.

The presence of strong patterns of LD increases the power of GENN to detect genetic associations.

IV. Discussion

The results of this study show that GENN is a promising solution to the analytical issues presented by LD in genetic association studies. The results indicate that strong patterns of LD among noise variables can improve rather than confound GENN.

While these results may be surprising, given that correlated predictor variables often attenuate the significance of the true outcome-associated variables for many analytical methods, careful examination of the GENN learning process provides a potential explanation. In [19], it was shown that GENN models absorb many input features in early generations, with noise variables culled in subsequent evolutionary steps. For the epistatic disease models considered here, wherein both functional variables must be simultaneously identified, stochastic searches have a low probability of hitting two variables jointly in a single iteration within a large combinatoric space. Graphically, such a situation presents a flat fitness landscape punctuated by only the peak representing the joint inclusion of both functional variants. However, any multicollinearity, such as LD, introduces topography into the landscape [36]. For noise variables, LD may introduce valleys in the fitness landscape that assist GENN in efficiently eliminating blocks of such non-informative variables. These valleys in the fitness landscape may also create better resolution between the correct functional variable(s) and the surrounding noise. This resolution may aid in the learning process.

For any functional variable(s) in LD with the genetic background, this topology means that fitness slopes upward along an LD gradient toward the peak including both functional variants. Since the probability of jointly hitting within two outcome-associated *sets* of variants is greater than jointly hitting upon single points, LD can actually help the stochastic search. Once the LD gradient is found, the learning process by which GENN prunes suboptimal variables in subsequent evolutionary steps can eventually arrive at a model representing the optimal fitness peak—including only the functional variants.

Additionally, NN in general may have an advantage in situations with inter-correlated variables over other statistical and machine-learning approaches. NNs are somewhat protected against the problems caused by multicollinearity due to their parallel nature [36-39]. Also, unlike many traditional statistical methods, NN do not assume independence of either individuals in the dataset or input variables. Adjustment of weights between network connections is assumed to correct for variable inter-correlation [38, 39]. These features of NNs may contribute to the robust nature of GENN in the presence of LD.

A crucial next step in the current study is a very precise and quantitative characterization of the LD patterns in the current data. LD patterns need to be assessed at the level of individual datasets to further dissect the performance of GENN in datasets when the functional variables are in LD with noise variables, and when they are not. Additionally, different simulation strategies will be needed to better understand these interesting initial results. Future simulations will need to specifically control patterns of LD surrounding functional variables.

Future studies will address questions about how the distribution of disease-associated polymorphisms with respect to different LD structures affects their identification. Specific questions include the following. How strong an LD "gradient" is necessary for GENN to capitalize on LD for identifying epistatic interactions? Does a variant's placement at the edge or in the center of a physical LD block affect its discovery? Given an LD metric (r^2, D', etc.), can we arrive at thresholds that determine when or if LD will be useful? The answers to these questions should give further insight into the breadth of analytical situations in which GENN would be the most appropriate analytical choice.

ACKNOWLEDGMENTS

This work was supported by National Institutes of Health grants HL65962, GM62758, and AG20135. This paper has been reviewed and approved for publication according to US EPA policy but does not necessarily represent the views of the Agency.

REFERENCES

1. Kardia S, Rozek L, Hahn L, Fingerlin T, Moore J: Identifying multilocus genetic risk profiles: a comparison of the multifactor data reduction method and logistic regression. Genetic Epidemiology 2000.
2. Moore JH, Williams SM: New strategies for identifying gene-gene interactions in hypertension. Ann Med. 34: 88-95. 2002.
3. Fu XJ, Wang LP. A GA-based novel RBF classifier with class-dependent features. Proc 2002 IEEE Congress on Evolutionary Computation 2, 1890-1894. 2002.
4. Fu XJ, Wang LP. Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance. IEEE Transactions System, Man, Cybern, Part B - Cybernetics 33, 399-409. 2003.
5. Hoh J, WAOJ. Trimming, Weighting, and Grouping SNPs in Human Case-Control Association Studies. Genome Res , 2115-2119. 2001.
6. Oh IS, Lee JS, Moon MR. Hybrid genetic algorithms for feature selection. IEEE Transactions Pattern Analysis and Machine Intelligence 26, 1424-1437. 2004.
7. Raymer ML, Punch WF, Goodman ED, Kuhn LA, Jain AK. Dimensionality reduction using genetic algorithms. IEEE Transactions on Evolutionary Computation 4, 164-171. 2000.
8. Ritchie MD, Hahn LW, Roodi N, Bailey LR, Dupont WD, Parl FF et al.: Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am J Hum Genet, 69: 138-147. 2001.
9. Lucek P, Hanke J, Reich J, Solla SA, Ott J: Multi-locus nonparametric linkage analysis of complex trait loci with neural networks. Hum Hered 1998, 48: 275-284.
10. Lucek PR, Ott J: Neural network analysis of complex traits. Genet Epidemiol, 14: 1101-1106. 1994.
11. Marinov M, Weeks D: The complexity of linkage analysis with neural networks. Human Heredity 2001, 51: 169-176.
12. North BV, Curtis D, Cassell PG, Hitman GA, Sham PC: Assessing optimal neural network architecture for identifying disease-associated multi-marker genotypes using a permutation test, and application to calpain 10 polymorphisms associated with diabetes. Ann Hum Genet, 67: 348-356. 2003.
13. Ritchie MD, White BC, Parker JS, Hahn LW, Moore JH: Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. BMC Bioinformatics, 4: 28. 2003.
14. Ritchie MD, Coffey CSMJH: Genetic programming neural networks: A bioinformatics tool for human genetics. Lecture Notes in Computer Science, 3102: 438-448. 2004.
15. Yao X. Evolving Artificial Neural Networks. Proc of the IEEE 97[9], 1423-1447. 1999.
16. Motsinger AA, Dudek SM, Hahn LW, Ritchie MD. Comparison of Neural Network Optimization Approaches for Studies of Human Genetics. Lecture Notes in Computer Science 3907, 103-114. 2006.

17. Motsinger AA, Dudek SM, Hahn LW, Ritchie MD. Grammatical Evolution for the optimization of neural networks for genetic association studies. Bioinformatics in submission. 2006.
18. Motsinger AA, Spencer K, Reif DM, Haines J, Ritchie MD. Grammatical Evolution Neural Networks Detects Genetic and Environmental Predictors of Age-Related Macular Degeneration. in preparation . 2006.
19. Motsinger AA, Rief DM, Dudek SM, Ritchie MD. Understanding the Evolutionary Process of Grammatical Evolution Neural Networks for Feature Selection in Genetic Epidemiology. IEEE Transactions in press. 2006.
20. De La Vega FM, Clark AG, Collins A, Kidd KK. Design and Analysis of Genetic Studies after the HapMap Project: Session Introduction. Pacific Symposium on Biocomputing 11:451-453. 2006.
21. Neeleman D. Multicollinearity in Linear Economic Models. Journal of the American Statistical Association. 69:348, 1049-1050. 1974.
22. Gordon, R A Issues in multiple regression. American Journal of Sociology, 73, 592-616. 1968.
23. O'Neill M, Ryan C. Grammatical Evolution. IEEE Transactions on Evolutionary Computation 5, 349-357. 2001.
24. O'Neill M, Ryan C. Grammatical evolution: Evolutionary automatic programming in an arbitrary language. 2003. Boston, Kluwer Academic Publishers.
25. Koza J. Genetic Programming. Encyclopedia of Computer Science and Technology. 1997.
26. Cantu-Paz E. Efficient and accurate parallel genetic algorithms. 2000. Boston, Kluwer Academic Publishers.
27. Motsinger AA, Hahn L.W., Dudek SM, Ryckman K.K., Ritchie MD. Alternative Cross-Over Strategies and Selection Techniques for Grammatical Evolution Optimized Neural Networks. Proceedings of the Genetic and Evolutionary Algorithm Conference. 1, 947-949. 2006. New York, Association for Computing Machinery Press.
28. Dudek S, Motsinger AA, Velez D, Williams SM, Ritchie MD: Data simulation software for whole-genome association and other studies in human genetics. Pacific Symposium on Biocomputing 11:499-510. 2006.
29. Barrett JC, Fry B, Maller J, Daly MJ: Haploview: analysis and visualization of LD and haplotype maps. Bioinformatics, 21: 263-265. 2005.
30. Culverhouse R, Suarez BK, Lin J, Reich T: A perspective on epistasis: limits of models displaying no main effect. Am J Hum Genet, 70: 461-471. 2005.
31. Moore JH: The ubiquitous nature of epistasis in determining susceptibility to common human diseases. Hum Hered, 56: 73-82. 2003.
32. Moore JH, Williams SM: Traversing the conceptual divide between biological and statistical epistasis: systems biology and a more modern synthesis. Bioessays, 27: 637-646. 2005.
33. Moore J, Hahn L, Ritchie M, Thornton T, White BC. Routine Discovery of High-Order Epistasis Models for Computational Studies in Human Genetics. Applied Soft Computing 4, 79-86. 2004.
34. Reif DM, Motsinger AA, McKinney BA, Crowe JE, Moore JH. Feature selection using Random Forests for the integrated analysis of multiple biological data types. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology In Press.
35. Ott J. Neural networks and disease association. American Journal of Medical Genetics (Neuropsychiatric Genetics) 105[60], 61. 2001.
36. Lyndsay JB, Shang JQ, Rowe RK. Using Complex Permittivity and Artificial Neural Networks for Contaminant Prediction. J. Envir. Engrg. 128, 740. 2002.
37. Smith, M. Neural Networks for Statistical Modeling, 1996. Boston: International Thomson Computer Press, ISBN 1-850-32842-0
38. De Veaux RD, Ungar LH. Multicollinearity: A Tale of two non-parametric regressions. In Selecting Models from Data: AI and Statistics IV, (ed P.Cheeseman and R.W. Oldford), pp.293-302. Springer-Verlag. 1994.
39. Carpio KJ, Hermosilla AY. On Multicollinearity and Neural Networks. Complexity International, Submitted preprint under review, PaperID:hermos01, URL:http://www.csu.edu.au/ci/draft/hermos01/