Pacific Symposium on Biocomputing 7:53-64 (2002)

A CELLULAR AUTOMATA APPROACH TO DETECTING INTERACTIONS AMONG SINGLE-NUCLEOTIDE POLYMORPHISMS IN COMPLEX MULTIFACTORIAL DISEASES

JASON H. MOORE, Ph.D., LANCE W. HAHN, Ph.D.
Program in Human Genetics, Department of Molecular Physiology and Biophysics, 519 Light Hall, Vanderbilt University Medical School, Nashville, TN 37232-0700, USA
Moore@phg.mc.Vanderbilt.edu

The identification and characterization of susceptibility genes for common complex multifactorial human diseases remains a statistical and computational challenge. Parametric statistical methods such as logistic regression are limited in their ability to identify genes whose effects are dependent solely or partially on interactions with other genes and environmental exposures. We introduce cellular automata (CA) as a novel computational approach for identifying combinations of single-nucleotide polymorphisms (SNPs) associated with clinical endpoints. This alternative approach is nonparametric (i.e. no hypothesis about the value of a statistical parameter is made), is model-free (i.e. assumes no particular inheritance model), and is directly applicable to case-control and discordant sib-pair study designs. We demonstrate using simulated data that the approach has good power for identifying high-order nonlinear interactions (i.e. epistasis) among four SNPs in the absence of independent main effects.

1 Introduction

The idea that epistasis or gene-gene interaction plays an important role in human biology is not new. In fact, Wright[1] emphasized that the relationship between genes and biological endpoints is dependent on dynamic interactive networks of genes and environmental factors. This idea holds true today. Gibson[2] stresses that gene-gene and gene-environment interactions must be ubiquitous given the complexities of intermolecular interactions that are necessary to regulate gene expression and the hierarchical complexity of metabolic networks. Indeed, there is increasing statistical and epidemiological evidence that epistasis is very common[3]. For example, in a study of 200 sporadic breast cancer subjects, Ritchie et al.[4] demonstrated a statistically significant interaction among four polymorphisms in three estrogen metabolism genes in the absence of any independent main effects. Further, Nelson et al.[5] found that epistatic effects of lipid genes on lipid traits was very common.

Despite the importance of epistasis in human biology there are few statistical methods that are capable of identifying interactions among more than two polymorphisms in relatively small sample sizes. For example, logistic regression is a commonly used method for modeling the relationship between discrete predictors such as genotypes and discrete clinical outcomes[6]. However, logistic regression, like most parametric statistical methods, is limited in its ability to deal

with high dimensional data. That is, when high-order interactions are modeled, there are many contingency table cells that have no observations. This can lead to very large coefficient estimates and standard errors[6]. One solution to this problem is to collect very large numbers of samples to allow robust estimation of interaction effects. However, the magnitudes of the sample sizes that are often required are prohibitively expensive. An alternative solution is to develop new statistical and computational methods that have improved power to identify multilocus effects in relatively small sample sizes.

Several groups have addressed the need for new methods by developing data reduction approaches[4,5]. These approaches reduce the dimensionality of the data by pooling multilocus genotypes into a smaller number of groups. For example, the multifactor dimensionality reduction (MDR) method of Ritchie et al.[4] pools multilocus genotypes into high risk and low risk groups effectively reducing the dimensionality of the genotype predictors from *n* dimensions to one dimension. The new one-dimensional multilocus genotype variable is evaluated for its ability to classify and predict disease status using cross-validation and permutation testing. This has been shown to be an effective strategy for identifying gene-gene interactions, however, there is a loss of information in the data reduction step. An alternative strategy is to use pattern recognition that has the advantage of considering the full dimensionality of the data. In this paper, we describe a new pattern recognition approach for identifying gene-gene interactions that takes advantage of the emergent computation features of cellular automata and intelligent search features of parallel genetic algorithms. Using simulated single-nucleotide polymorphisms (SNPs) in a discordant sib-pair study design, we demonstrate that the CA approach has good power for identifying high-order nonlinear interactions with few false-positives.

2 Overview of Cellular Automata

Cellular automata (CA) are discrete dynamic systems that consist of an array of cells, each with a finite number of states. The state of each cell changes in time according to the local neighborhood of cells and their states as defined by a rule table. The simplest CA consist of one-dimensional arrays, although some, such as the Game of Life[7], are implemented in two or more dimensions. Figure 1A illustrates an example of a simple one-dimensional CA iterated through several time steps along with the simple rule table that governs its behavior. In this section, we review how CA can be used to perform computations and then how we exploit this feature to perform multilocus pattern recognition.

2.1 Emergent Computation in Cellular Automata

An intriguing feature of CA is their ability to perform emergent computation[8,9]. That is, the local interactions among spatially distributed cells over time can lead to an output array that contains global information for the problem at hand. For example, Mitchell et al.[9] have used CA to perform density estimation. In that application, a one-dimensional, two-state (1 or 0) CA is given an initial array of states. The goal is to identify a CA rule set such that the CA converges to an array of all 1's if the density of 1's is greater than 0.5 and to an array of all 0's if the density of 1's is less than 0.5. They found that the CA is able to perform this computation through a series of spatially and temporally extended local computations. This emergent computation feature of CA forms the basis of our proposed method for identifying patterns of genotype variations associated with common complex multifactorial diseases.

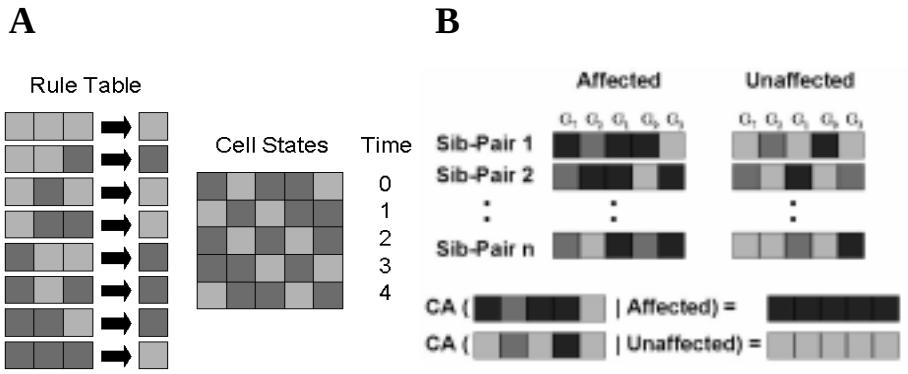

Figure 1. (A) The rule table, cells, and cell states for a simple one-dimensional CA iterated through four time steps. Note that the state of a given cell at time t is determined by the states of that cell and the states of adjacent cells at time t-1 as defined by the rule table. (B) General approach to using CA to perform emergent computation on combinations of genetic variations from affected and unaffected siblings. The goal is to identify a CA model that is able to produce one type of output (e.g. all black cells states) if the genetic variation input is from an affected sibling and another type of output (e.g. all gray cell states) if the input is from an unaffected sibling.

2.2 Description of our Cellular Automata Approach to Pattern Recognition

We begin with a description of the discordant sib-pair study design, a commonly employed design for identifying common complex disease susceptibility genes using single-nucleotide polymorphisms in human populations. With this approach, sib-pairs in which one sibling is affected with the disease and the other is unaffected are ascertained and genetic variations measured in each. The benefits of this study design are twofold. First, many common complex diseases have a late age of onset thus limiting the ability to collect parental samples which are useful for some types of statistical tests. Second, using unaffected sibs as controls instead of unrelated

subjects prevents false-positive results due to population stratification (e.g. by chance you select unaffected controls of a different ethnic/genetic background). This is because, by definition, unaffected sibs come from the same ethnic or genetic background as the affected sib. Traditional statistical methods such as the sibship transmission/disequilibrium test (Sib-TDT)[10] compare observed differences in the frequencies of alleles (i.e. a single genetic variation from one of the two chromosomes) or genotypes (i.e. combination of two alleles) among affected and unaffected sib-pairs with the expected difference of zero under the null hypothesis that the particular genetic variation is not associated with the disease. TDT-type statistics have reasonable power for identifying genetic variations that have moderate to large effects on disease risk. This is evident from studies of linkage disequilibrium among single-nucleotide polymorphisms in and near the *APOE* gene in Alzheimer disease[25]. However, TDT statistics have low power for identifying genetic variations whose effects on disease risk are fully or partially through interaction with other genetic variations. This is because, in its original form, the TDT is a univariate statistic that considers only one genetic variation at a time.

We have developed a CA approach to identifying patterns of genotype variations associated with disease using the discordant sib-pair, or case-control, study design. The general approach is to identify a set of CA operating features that is able to take an array of genotypes as input and produce an output array that can be utilized to classify and predict whether siblings are affected or unaffected (see Figure 1B). In this initial study, we fixed the number of cells in the CA to five. Thus, the CA is presented with a maximum of five unique and/or redundant genotypes. We also allowed 'don't care' or wildcard cells to be included. This permits less than five genotypes to be evaluated. Wildcard cells all have the same state and do not contribute any information for discriminating affected from unaffected sib pairs. Assuming each genetic locus has only three possible genotypes, we used a binary encoding with '01' for the first genotype, '10' for the second genotype, '11' for the third genotype, and '00' for the wildcard. Thus, each array presented to the CA consisted of 10 bits with two bits encoding the state of each of the five cells. We used a simple nearest-neighbor rule table that is implemented by looking at the state of the cell in question and the adjacent cell states as is illustrated in Figure 1A. With three cells forming a rule and four different states per cell, there are 4^3 or 64 possible rule inputs with four possible output states for each. An important feature of CA is the number of time steps or iterations. This will govern the amount of spatial and temporal information processing that can performed. In this study, we allowed a maximum of 128 iterations for each CA. This maximum number of iterations was selected to allow enough time for parallel processing of all the information in an input array without an excessive number of iterations that might complicate interpretation. Thus, there are three essential components to the CA model. First, the correct combination of genetic variations must be selected for initiating the CA cell states. Second, the appropriate rule table that specifies the information processing must be selected. Finally, the number of time steps or

iterations for the CA must be selected. We used the genetic algorithm machine learning methodology to optimize selection of these three model components (described below).

How is the output array of the CA used to perform classification and prediction? We first count the number of 1s in the binary encoded output array of the CA run on each set of genotypes for each affected and each unaffected sib in the sample. A classifier is formed by using a frequency histogram of the number of 1s among affected sibs and unaffected sibs. Each histogram bin is labeled affected or unaffected depending on whether the number of 1s represented by that bin were more frequently observed among affected or unaffected sibs. For example, consider the case where 100 discordant sib pairs were evaluated. Suppose the number of CA output arrays that contained three 1s was 20 for affected sibs and 10 for unaffected sibs. This bin would be labeled affected and thus the 10 unaffected sibs would be misclassified. This would contribute 0.05 to the overall misclassification rate. This is performed for each bin and a total classification error is estimated by summing together the individual rates for each bin.

3 Cellular Automata Optimization using Parallel Genetic Algorithms

3.1 Overview of Parallel Genetic Algorithms

Genetic algorithms (GAs), neural networks, case-based learning, rule induction, and analytic learning are some of the more popular paradigms in machine learning[11]. Genetic algorithms perform a beam or parallel search of the solution space that is analogous to the problem solving abilities of biological populations undergoing evolution by natural selection[12, 13]. With this procedure, a randomly generated 'population' of solutions to a particular problem are generated and then evaluated for their 'fitness' or ability to solve the problem. The highest fit individuals or models in the population are selected and then undergo exchanges of random model pieces, a process that is also referred to as recombination. Recombination generates variability among the solutions and is the key to the success of the beam search, just as it is a key part of evolution by natural selection. Following recombination, the models are reevaluated and the cycle of selection, recombination, and evaluation continues until an optimal solution is identified.

As with any machine learning methodology[11], GAs are not immune to stalling on local optima[14]. To address this issue, distributed or parallel approaches to GAs have been implemented[15]. Here, the GA population is divided into sub-populations or demes. At regular iterative intervals, the best solution obtained by each sub-population is migrated to all other sub-populations. This prevents individual sub-

populations from converging on a locally optimum peak because new highly fit individuals are periodically arriving to increase the population diversity. In biology, it is believed that evolution progresses faster in semi-isolated demes than in a single population of equal size[16]. Indeed, there is some evidence that parallel GAs actually converge to a solution much faster than serial or single-population GAs[15]. This superlinear speedup may be due to additional selection pressure from choosing migrants based on fitness[15].

Genetic algorithms have been applied to microarray data analysis[17, 18] and are ideally suited for selecting polymorphisms and optimizing CA[9].

3.2 Solution Representation and Fitness Determination

The first step in implementing a GA is to represent the solution or model to be optimized as a one-dimensional binary array or chromosome. For the CA, we needed to encode five genetic variations and/or wildcards, the CA rule table, and the number of CA iterations (see Figure 2). Each of the genetic loci and the number of CA iterations were represented using a total of six 32-bit integers with a modulo operation used to constrain the integer to the desired range (0-19 for the genetic loci (described in Section 4) and the wildcard and 0-127 for the number of iterations.) As previously described, each CA cell has four possible states and each rule depends on the state of three cells. Encoding this set of 64 rules, with each rule producing one of four two-bit states as output, requires four 32-bit integers. In total, the GA manipulated 10 32-bit integers for a total chromosome length of 320 bits (see Figure 3). Fitness of a particular CA model is defined as the ability of that model to classify siblings as affected or unaffected. Thus, the goal of the GA is to identify a CA model that minimizes the misclassification error. Implementation using cross validation is described below.

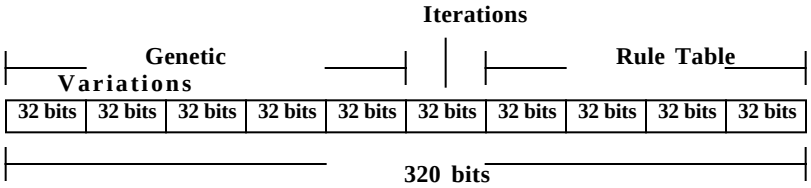

Figure 2. Encoding of the genetic algorithm chromosome. The first five 32-bit segments encode genetic variations and/or wild cards while the sixth 32-bit segment encodes the number of iterations. The last four 32-bit segments encode the CA rule table.

3.3 Parallel Genetic Algorithm Parameters

We implemented a parallel genetic algorithm with two sub-populations or demes undergoing periodic migration. Each GA was run for a total of 200 iterations or generations with migration of the best solutions to each of the other sub-populations every 25 iterations. Sub-population or deme sizes of 10, 50, 100, 200, 500, and 1000 were used for the analysis of all 50 simulated datasets (see Section 4). We used a standard recombination frequency of 0.6 and a standard mutation frequency 0.02[13].

3.4 Hardware and Software

The parallel GA used was a modification of the Parallel Virtual Machine (PVM) version of the Genetic ALgorithm Optimized for Portability and Parallelism System (GALLOPS) package for UNIX[19]. This package was implemented in parallel using message passing on a 110-processor Beowulf-style parallel computer cluster running the Linux operating system. Two processors were used for each separate run. Information about obtaining the CA and GA software can be found at http://phg.mc.vanderbilt.edu/software.

3.5 Implementation

The goal of the GA is to minimize the classification error of the CA model. However, from a genetics perspective, the goal is to identify the correct functional genetic variations. That is, from a pool of many candidate genes, the goal is to find those that play a role in influencing risk of disease. We used a 10-fold cross-validation strategy[24] to identify optimal combinations of genetic variations. Cross-validation has been a successful strategy for evaluating multilocus models in other studies of common complex diseases[4]. Here, we ran the GA on each 9/10 of the data and retained the CA models that minimized the misclassification rate. Across the 10 retained models, we selected the combination of polymorphisms that was observed most frequently. The reasoning is that the functional set of genetic variations should be identified consistently across different subsets of the data[4]. To determine statistical significance of the observed cross-validation consistency, we permuted the data 1,000 times to determine the empirical distribution of cross-validation consistency were the null hypothesis true. We rejected the null hypothesis of no association when the upper-tail Monte Carlo p-value was ≤ 0.05. We also estimated the general prediction error of the best set of retained models using each of the independent 1/10 of the data.

4 Data Simulation

The goal of the simulation study was to generate a series of datasets in which the probability of a sibling being affected is dependent on epistasis or interaction among a set of genetic variations. We first simulated 100 sibling pairs each with 20 unlinked (i.e. on different chromosomes) genetic variations using the Genometric Analysis Simulation Program (GASP)[20]. Each genetic variation had two alleles with frequencies 0.6 and 0.4. Four of the 20 genetic variations served as the functional genetic loci. The remaining 16 genetic variations serve as potential false-positives. Thus, the goal of the CA was to identify the correct four functional genetic variations from the total of 20 candidates.

The epistatic interaction among the four functional genetic variations was accomplished using a Boolean network (Figure 3). Alleles (e.g. A_1 and A_2) were encoded as either 1 for an A allele or 0 for an a allele. The allele combinations at each genetic locus (A-D) that contribute to disease risk were as follows; $A_1 = 1$ and $A_2 = 1$, $B_1 = 1$ and $B_2 = 0$, $C_1 = 0$ and $C_2 = 0$, $D_1 = 0$ and $D_2 = 0$. The logic functions AND, NOR, INHIBITION, and XOR[26] were selected such that the Boolean network would produce a one or affected status if a particular subject had one and only one of the four specified allele combinations. The XOR function has been described as an epistasis model[21] and is often used to evaluate pattern recognition approaches because of its inherent nonlinearity. With this model, the probability of being affected is one if the sibling has one and only one of the allele combinations or genotypes listed above. Inheriting more than one of the allele combinations listed above is protective against disease. The independent main effects of each genetic variation are minimal under this epistasis model.

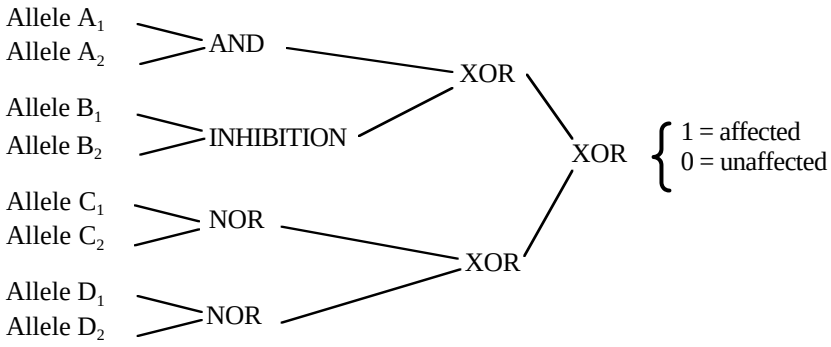

Figure 3. Boolean network used to simulate the epistasis effects on disease risk. The alleles of each single-nucleotide polymorphism combine to form genotypes using the AND, NOR, and INHIBITION functions. The genotypes are then combined using XOR functions that output affected status. Each sibling is at increased risk of being affected if they inherit one, and only one, of the four genotypes. Inheriting more than one of the particular genotypes is considered protective. With this epistasis model, the independent main effects of each genotype are minimized.

5 Data Analysis

The goal of the statistical analysis was to determine whether the cross-validation consistency (see Section 3.5) for a particular combination of genetic variations is expected by chance if the null hypothesis were true. A particular genetic variation was considered statistically significant if it was identified in six or more of the 10-fold cross-validation datasets. As described, this decision rule was determined empirically by permuting the data 1,000 times and selecting the number of 10-fold cross-validation datasets for which loci would be identified less than 5% of the time. This corresponds to a statistical significance level of 0.05. The power of the CA approach is reported as the number of simulated datasets out of 50 in which the correct four functional genetic variations were identified in six or more of the cross-validation datasets as described above.

6 Results

We first analyzed the single-locus effects of each of the 20 simulated SNPs in each dataset of 100 discordant sib-pairs using the traditional Sib-TDT statistic[10]. The power to detect the independent effects of each of the four functional loci was less than 50% while the power to detect each of the false-positive loci was close to the expected false-positive rate of 5%. Thus, a traditional approach would have missed each of the four functional loci more than half the time with occasional false-positives. This indicates that none of the four SNPs have a large independent main effect on disease risk. This is consistent with the simulation strategy used (see Section 4).

Table 1. Power (datasets out of 50) of CA to identify each locus.

Sub-Population Size	Locus A	Locus B	Locus C	Locus D	False Positives
10	4	0	16	1	27
50	50	20	50	34	0
100	50	26	50	33	0
200	50	32	50	37	0
500	50	30	50	40	0
1000	50	34	50	40	0

Table 1 summarizes the number of datasets (out of 50 total) in which each of the functional loci (A-D) were identified using the multi-locus CA approach. For GA deme sizes of 50 or greater, the CA yielded 100% power for identifying loci A and C. That is, these two genetic loci were always found. When the deme sizes

were 1000, the power to detect locus B was 68% while the power to detect locus D was 80%. The power to detect the four functional loci was greatly improved over that of the Sib-TDT by considering combinations of SNPs using the CA approach. It should be noted that a GA deme size of 10 yielded low power and many false-positives. However, when a GA deme size of 50 or greater was used, no false-positives were identified. These results suggest that the CA pattern recognition approach is useful for identifying combinations of genetic variations that influence disease risk primarily through gene-gene interactions.

7 Discussion

We have introduced a cellular automata (CA) approach to identifying patterns of variations in multiple SNPs associated with risk of common complex multifactorial diseases. The development of this CA approach was motivated by the limitations of the generalized linear model for detecting and characterizing gene-gene[3] and gene-environment[22] interactions.

The CA approach shares many of the same advantages of the multifactor dimensionality reduction (MDR) approach[4]. For example, the CA approach is nonparametric. As Ritchie et al.[4] describe, this is an important distinction from traditional parametric statistical methods that rely on the generalized linear model. For example, with logistic regression, as each additional main effect is included in the model, the number of possible interaction terms grows exponentially. The number of orthogonal regression terms needed to describe the interactions among a subset, *k*, of *n* biallelic loci is *n* choose *k* multiplied by 2 raised to the power of k[23]. Thus, for 20 polymorphisms we would need 40 parameters to model the main effects (assuming two dummy variables per biallelic locus), 1,560 parameters to model the two-way interactions, 79,040 parameters to model the three-way interactions, 1,462,240 parameters to model the four-way interactions, etc. Thus, fitting a full model with all interaction terms and then using backward elimination to derive a parsimonious model would not be possible. The CA approach avoids the pitfalls associated with using parametric statistics such as logistic regression for modeling high-order interactions.

An additional advantage of the CA approach is that it assumes no particular genetic model (i.e. model-free). That is, no mode of inheritance needs to be specified. This is important for diseases such as cardiovascular disease and depression in which the mode of inheritance is unknown and likely very complex. In its current form, this approach can be directly applied to case-control and discordant sib-pair study designs. Extension to other family-based control study designs such as triads should also be possible.

As with MDR[4], an advantage of the CA approach is that false-positive results due to multiple testing are minimized. Indeed, we detected no false-positives in the

present study when a GA deme size of 50 or greater was used. This is primarily due to the cross-validation strategy used to select optimal models. Data reduction and pattern recognition approaches are good at identifying complex relationships in data, even when those relationships are due to chance or false-positive variations. However, the real test of any approach is its ability to make predictions in independent data[24]. Cross-validation divides the data into 10 equal parts allowing 9/10 of the data to be used to develop a model and the independent 1/10 of the data used to evaluate the predictive ability of the model[24]. Optimal models are selected solely on their ability to make predictions in independent data. Once a final predictive model has been selected, only then is the null hypothesis of no association tested via permutation testing. It is this combined cross-validation and permutation testing approach that minimizes false-positives due to multiple looks at the data[4].

There are several advantages of the CA approach over data reduction approaches such as MDR. First, there is no loss of information. The CA considers the full dimensionality of the data whereas methods such as MDR seek to reduce the dimensionality of the data and in doing so lose information. Additionally, the CA does not have the same limitation as MDR for making predictions during cross-validation in high-dimensional data[4]. This is because the rule set that is generated is general enough to accept any combination of genotypes to make a prediction of disease risk.

Despite these important advantages, there are also several disadvantages to the CA approach. Most importantly, CA models are very difficult to interpret. Understanding the relationship among the multilocus SNP genotypes requires interpreting the spatial and temporal information processing that is occurring in the CA to produce a predictive output. Although Mitchell et al.[9] have made progress in this area, there is clearly a lot of work remaining. An additional disadvantage over traditional methods is that the CA approach is very computationally intensive. Selection of SNPs and optimization of the CA parameters requires a machine learning strategy such as the parallel GA. Effective implementation of GAs requires at least several workstations that are part of Beowulf-style parallel computer system. However, it should be noted that these systems are very inexpensive to set up since they use commodity-priced off-the-shelf components and freely available software.

In conclusion, we have introduced a new approach to identifying patterns of SNP variations in complex multifactorial diseases. This approach takes advantage of the emergent computation features of one-dimensional CA and the intelligent search features of parallel GAs. We anticipate that the results of this study will open the door for investigations of using CA to identify combinations of SNPs that interact in a non-additive or nonlinear manner to influence risk of common complex multifactorial diseases.

References

1. S. Wright, *Proc. 6th Int. Conf. Genet*. **1**, 356 (1932)
2. G. Gibson, *Theor. Popul. Biol.* **49**, 58 (1996)
3. A.R. Templeton in *Epistasis and Evolutionary Process*, Eds. M. Wade et al (Oxford University Press, New York, 2000)
4. M.D. Ritchie et al, *Am. J. Hum. Genet.* **69**, 138 (2001).
5. M.R. Nelson et al, *Genome Res.* **11**, 458 (2001).
6. D.W. Hosmer and S. Lemeshow, *Applied Logistic Regression* (John Wiley & Sons Inc., New York, 2000)
7. M. Gardner, *Sci. Amer*. **223**, 120 (1970)
8. M. Sipper, *Evolution of Parallel Cellular Machines* (Springer, New York, 1997)
9. M. Mitchell et al, *Physica D*, **75**, 361 (1994)
10. R.S. Spielman and W.J. Ewens, *Am. J. Hum. Genet.* **62**, 450 (1998)
11. P. Langley, *Elements of Machine Learning* (Morgan Kaufmann Publishers, Inc., San Francisco, 1996)
12. J.H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975)
13. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, 1989)
14. W. Banzhaf et al, *Genetic Programming: An Introduction* (Morgan Kaufmann Publishers, San Francisco, 1998)
15. E. Cantu-Paz, *Efficient and Accurate Parallel Genetic Algorithms* (Kluwer Academic Publishers, Boston, 2000)
16. S. Wright S, *Genetics* **28**, 114 (1943)
17. J.H. Moore and J.S. Parker, *Lec. Notes Artificial Intel.* **2167**, 372 (2001)
18. J.H. Moore and J.S. Parker in *Methods of Microarray Data Analysis* (Kluwer Academic Publishers, Boston, in press)
19. http://garage.cps.msu.edu
20. A.F. Wilson et al, *Am. J. Hum. Genet.* **59**, A193 (1996)
21. W. Li and J. Reich, *Hum. Hered.* **50**, 334 (2000)
22. C.D. Schlichting and M. Pigliucci, *Phenotypic Evolution: A Reaction Norm Perspective* (Sinauer Associates, Inc., Sunderland, 1998)
23. M.J. Wade in *Epistasis and Evolutionary Process*, Eds. M. Wade, B. Brodie III, J. Wolf (Oxford University Press, New York, 2000)
24. B.D. Ripley, *Pattern Recognition and Neural Networks* (Cambridge University Press, Cambridge, 1996)
25. E.R. Martin et al, *Am. J. Hum. Genet.* **67**, 383 (2000)
26. D. Kaplan and L. Glass, *Understanding Nonlinear Dynamics* (Springer-Verlag, New York, 1995)