Systems Biology Thought Experiments in Human Genetics Using Artificial Life and Grammatical Evolution

Bill C. White and Jason H. Moore

Center for Human Genetics Research, Vanderbilt University, Nashville, TN 37323-0700
{bwhite, moore}@chgr.mc.vanderbilt.edu

Abstract

A goal of systems biology and human genetics is to understand how DNA sequence variations impact human health through a hierarchy of biochemical, metabolic, and physiological systems. We present here a proof-of-principle study that demonstrates how artificial life in the form of agent-based simulation can be used to generate hypothetical systems biology models that are consistent with pre-defined genetic models of disease susceptibility. Here, an evolutionary computing strategy called grammatical evolution is utilized to discover artificial life models. The goal of these studies is to perform thought experiments about the nature of complex biological systems that are consistent with genetic models of disease susceptibility. It is anticipated that the utility of this approach will be the generation of biological hypotheses that can then be tested using experimental systems.

Introduction

Human genetics is largely concerned with the relationship between DNA sequence variations and measures of human health. Genotypes influence phenotypes through a hierarchical network of biochemical, metabolic, and physiological systems in the context of environmental exposure. Systems biology is an emerging discipline focused on developing comprehensive laboratory and analytical strategies for understanding the complex biological systems that complete the genotype to phenotype mapping relationship (Ideker et al. 2001). The promise of a joint human genetics and systems biology approach is improved human health through advances in disease diagnosis, prevention, and treatment.

Understanding the genetic architecture of common human diseases such as essential hypertension requires a research strategy that embraces, rather than ignores, complexity due to nonlinear gene-gene interactions or epistasis (Moore and Williams 2002). The biological definition of epistasis is one gene standing upon or masking the effects of another gene (Bateson 1909) while the statistical definition is a deviation from additivity in a linear model (Fisher 1918). We have presented the working hypothesis that epistasis is a ubiquitous component of the genetic architecture of common human diseases (Moore 2003).

There are several general systems biology strategies that can be utilized to study the genetics of susceptibility to common human diseases. One strategy relies on the collection and integrated analysis of genetic, genomic, and proteomic data from complex biological systems (e.g. Reif et al. 2004). A second strategy relies on perturbation of complex biological systems in model organisms (Jansen 2003). A third strategy relies on computer simulations for carrying out thought experiments that can be used to generate testable hypotheses (e.g. Di Paolo et al. 2000). In the present study, we explore the utility of using artificial life in the form of agent-based modeling for carrying out systems biology thought experiments for generating hypotheses about the mapping relationship between genotype and susceptibility to common disease.

Our previous work in this area has focused on the use of a discrete dynamical systems modeling tool called Petri nets. Petri nets are a type of directed graph (Desel and Juhas 2001) that have been used to model biochemical systems (Goss and Pecoud 1998). We utilized an evolutionary computing strategy called grammatical evolution to discover Petri net models of hypothetical biochemical systems that are consistent with a fixed genetic model of disease susceptibility (Moore and Hahn 2003a, 2003b, 2004a, 2004b). These proof-of-principle studies provided evidence that it is possible to routinely generate discrete dynamic systems models that are consistent with genetic models in which disease susceptibility is dependent on nonlinear interactions among genotypes from two or three DNA sequence variations.

While the Petri net approach has been very successful, it is our conjecture that agent-based modeling will provide more flexibility for carrying out thought experiments. That is, we anticipate a system comprised of agents and rules for their physical interaction will provide a wider range of possible system behaviors than that afforded by the Petri nets. To this end, we provide here an initial proof-of-principle study that demonstrates an artificial life modeling strategy is capable of generating systems biology models that are consistent with a defined genetic model. This interdisciplinary study brings together concepts from several disparate fields that are summarized in the next several sections. We first review the nonlinear gene-gene interaction models from which the systems biology models are derived. Next, we review agent-based simulation and then our agent-based modeling strategy. Finally, we review our evolutionary computing strategy for model discovery that is based on grammatical evolution. The

final sections of the paper present the research results and a discussion with future directions.

The Nonlinear Gene-Gene Interaction Models

Our two high-order, nonlinear, gene-gene interaction (i.e. epistasis) models are based on penetrance functions. Penetrance functions represent one approach to modeling the relationship between genetic variations and risk of disease. Penetrance is simply the probability (P) of disease (D) given a particular combination of genotypes (G) that was inherited (i.e. P[D|G]). Figure 1 illustrates the penetrance functions used for Models 1 and 2, respectively. Each model has been described previously (Frankel and Schork 1996; Li and Reich 2000; Moore et al. 2002). What makes these models interesting is that disease risk is dependent on each particular combination of all three genotypes inherited. Each single genotype has effectively no main effect on disease risk.

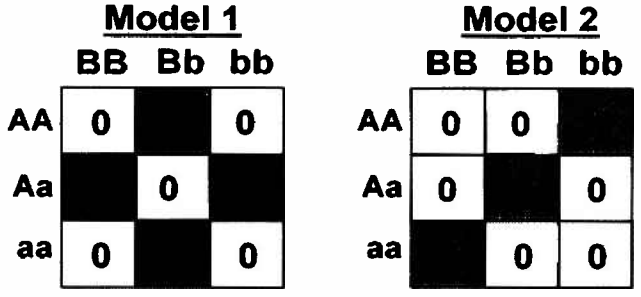

Figure 1: Gene-gene interaction Models 1 and 2.

Agent Based Simulation

Agent based simulation (ABS) refers to a branch of artificial intelligence in computer science that is concerned with multiple, autonomous, interacting computing elements and their emergent behavior (d'Inverno 2001, Woolridge 2002). While the idea of what an agent is and does is not well defined, agents used in this study follow the notion of "weak agency" as put forth by Wooldridge and Jennings (1995) in which agents are defined as being autonomous, interacting with other agents, reacting to their environment and pursuing their own goals in self interest.

Agents in our simulations move and collide with each other on a grid of fixed size. Boundary conditions are handled by allowing wraparound at the grid edges, that is, the world is a toroidial grid. Agents begin in a random spatial configuration with predefined move and collision behaviors and end in a final state after a specified number of time steps. Some agents move in fixed ways, while others are dependent on global environmental conditions specified at the beginning of the simulation, specifically whether agent behavior is dependent on characteristics of the system being modeled. Agent interaction (therefore communication) happens via collisions. Agents can sense whether the location they move to has an agent already occupying that location. If so, the agent is said to have collided with the existing agent, and it reacts using its collision rule. Agents are allowed to occupy the same space; therefore, infinite move-collision recursions are avoided. In cases where no collision is detected, agents continue to move according to their move rules, which define their goals. The expected emergent behavior relates agent interaction dynamics to the system being modeled.

Our Agent Based Simulation Modeling Strategy

Moore and Hahn (2003a, 2003b, 2004a, 2004b) developed a strategy for identifying discrete, dynamic models of biochemical systems that are consistent with observed gene-gene interactions that define disease susceptibility. The specific systems used to model the biochemical pathways were Petri nets with time (Merlin 1974, Ramchandani 1974).

The goal of identifying discrete models of biochemical systems that are consistent with observed population-level gene-gene interactions is accomplished here by developing agent based simulations that are dependent on specific genotypes from two DNA sequence variations. We allow movement and collision behavior to be genotype-dependent; therefore, simulations can yield different ending configurations. Each agent based simulation model is related to the genetic model using a discrete version of the threshold model from population genetics (Falconer and Mackay, 1996). With a classic threshold or liability model, it is the concentration of a biochemical or environmental substance that is related to the risk of disease, under the hypothesis that risk of disease is greatly increased once a particular substance exceeds some threshold concentration. Conversely, the risk of disease may increase in the absence of a particular factor or with any significant deviation from a reference level. In such cases, high or low levels are associated with high risk while an intermediate level is associated with low risk. Here, we use a discrete version of this model for our deterministic ABS. For each model, the number of agents at a particular space of the simulation world is recorded, and if they exceed a certain threshold, the appropriate risk assignment is made. If the number of agents does not exceed the threshold, the alternative risk assignment is made. The high-risk and low-risk assignments made by the discrete threshold from the output of the ABS can then be compared to the high-risk and low-risk genotypes from the genetic model. A perfect match indicates the ABS model is consistent with the gene-gene interactions observed in the genetic model. The ABS then becomes a model that relates the DNA sequence variations to risk of disease through an intermediate biochemical network.

Identifying ABS models that are consistent with the genotype-dependent distribution of risk is challenging by trial and error. Therefore, we developed an evolutionary computing approach to the discovery of ABS models. This approach is described in the next section.

A Grammatical Evolution Approach to Discovering Agent Based Simulation Models

Overview of Grammatical Evolution

Evolutionary computation arose from early work on evolutionary programming (Fogel 1962, Fogel et al. 1966) and evolution strategies (Rechenberg, 1964, Schwefel 1965) that used simulated evolution for artificial intelligence. The focus on representations at the genotypic level lead to the development of genetic algorithms by Holland (1962, 1975) and others. Genetic algorithms have become a popular machine intelligence strategy because they can be effective for implementing parallel searches of rugged fitness landscapes (Goldberg 1989). Briefly, this is accomplished by generating a random population of models or solutions, evaluating their ability to solve the problem at hand, selecting the best models or solutions, and generating variability in these models by exchanging model components among different models. The process of selecting models and introducing variability is iterated until an optimal model is identified or some termination criteria are satisfied. Koza (1992) developed a variation on genetic algorithms called genetic programming where the models or solutions are represented by tree structures that are in turn executed as computer programs. Koza (2001) and others (Kitagawa 2003) have applied genetic programming to modeling metabolic networks.

Grammatical evolution has been described by O'Neill and Ryan (2001, 2003) as a variation on genetic programming. Here, a grammar is specified that allows a computer program or model to be constructed by a simple genetic algorithm operating on an array of numbers called "codons." The evolved codons select grammar elements in the derivation of a valid sentence in the language specified by the grammar. For our purposes the language L is all valid ABS configurations. This approach is appealing because only a text file specifying the grammar needs to be altered for different simulations, that is, as long as the grammar specifies valid sentences of the language L(G). There is no need to modify and recompile source code during development once the fitness function is specified. The end result is a decrease in development time and an increase in computational flexibility.

A Grammar for Agent Based Simulation Models in Backus-Naur Form

Backus-Naur Form (BNF) is a formal notation for describing the syntax of a context-free grammar as a set of production rules that consist of terminals and nonterminals (Hopcroft and Ulman 1979). Nonterminals form the left-hand side of production rules while both terminals and nonterminals form the right-hand side. A terminal is a simulation/model element or parameter, and a nonterminal is the name of a production rule. Use of nonterminals in the right-hand side of production rules allows for recursion, deriving more complex sentences, that is, simulations, by expanding these nonterminals recursively. For the ABS models, the terminal set includes, for example, the basic building blocks of an ABS: agents and their movement and collision behavior rules. The nonterminal set includes the names of production rules that construct the ABS. For example, a nonterminal might name a production rule for determining whether an agent has movement and collision behavior that is fixed or genotype-dependent. We show in (1) below the production rule that is the start symbol to begin the derivation and thus the ABS configuration. In the grammar rules, variables shown in all capital letters represent terminals and variables contained within angle brackets represent nonterminals.

<simulation>	::=	NUM_TIMESTEPS = <constant > GRID_SIZE = <constant> <constant> <statement_list>	(1)

The <simulation> production rule specifies that a valid simulation configuration must have a number of timesteps, a grid/world size (fixed here to 12 by 12) and a statement list. The nonterminal <statement_list> is a production rule that allows the ABS to grow. The production rule for <statement_list> is shown below in (2).

<statement_list>	::=	<statement> | <statement> <statement_list>	(2)

Here, the BNF symbol "|" separates choices in the substitution for <statement_list> in the derivation of valid ABS configurations. The specific choice is determined by applying the modulus operator to the current genetic algorithm codon. In the case of <statement_list>, the genetic algorithm codon would be applied modulo two, making the choice equally probable. If the second of the alternatives is chosen, the derived configuration is extended by substituting an instance of the <statement_list> nonterminal itself. This process can repeat recursively until the first alternative is chosen and a <statement> nonterminal ends the recursion.

A <statement>, shown in (3) below, defines an agent in the ABS; therefore, rule (2) above allows the simulation to have one or more agents.

<statement>	::=	AGENT MOVE <move_rule> COLLIDE <collision_rule>	(3)

Agent move and collision behavior is then defined by the grammar to allow for many types of movement and interaction, including becoming stuck, that is, an agent can become an obstacle. The grammar for all behaviors is too

large to reproduce here. A full grammar can be obtained from the authors upon request.

The Fitness Function

Once an ABS model is constructed using the BNF grammar, as dictated by the genetic algorithm chromosome (vector of codons), the model fitness is determined. Similar to the Petri net approach described by Moore and Hahn (2003a, 2003b, 2004a, 2004b), this is carried out by executing the ABS model for each combination of genotypes in the genetic model and comparing the final agent counts at a defined quadrant of the grid world to a threshold constant to determine the risk assignment. Let G be the set of $i = 1$ to n possible genotype combinations where n = 9 when there are two DNA sequence variations, each with three genotypes. Let Z_i be the final number of agents from the designated ABS quadrant for the ith genotype combination and let c be the threshold constant. Let $d(G_i)$ be the risk assignment for the ith genotype combination in the genetic model and let $f(G_i)$ be the risk assignment made by the ABS. If $Z_i \geq c$ then $f(G_i)$ = "high risk" else if $Z_i < c$ then $f(G_i)$ = "low risk". The dichotomous risk assignment is consistent with epidemiological study designs in which subjects with the disease (cases) and subjects without the disease (controls) are used to identify genetic risk factors. Genotypes more common in cases than controls can be thought of as high risk. Fitness (E) of the ABS model is determined by comparing the high risk and low risk assignments made by the ABS to those from the given nonlinear gene-gene interaction model. Calculation of the fitness value, E, is given by the classification error function E shown in (4) below. In the present study, max(E) = 9 and min(E) = 0. The goal is to minimize E.

$$E = \sum_{i=1}^{|G|} e_i \quad e_i = 0 \text{ if } f(G_i) = d(G_i)$$

$$e_i = 1 \text{ if } f(G_i) \neq d(G_i) \qquad (4)$$

Genetic Algorithm Parameters

Grammatical evolution works by decoding genetic algorithm chromosomes. Our GA chromosome consisted of 250 integer codons. In implementing the grammar, it is possible to reach the end of a chromosome with an incomplete instantiation of the grammar. To complete the instance, chromosome wraparound was used (O'Neill and Ryan 2001, 2003). In other words, the instance of the grammar was completed be reusing the chromosome. Wraparound imperfectly solves this problem by simply returning to the beginning of the chromosome to acquire more codons to expand the production rules. The solution is imperfect because there is no guarantee that the grammar is finite; therefore, it is possible for the derivation to recurse endlessly, so there must be a stopping point. In the present study, we used a stopping point of 10 wraparounds.

Genetic algorithms require the setting of many parameters. Table 1 below summarizes the genetic algorithm parameter settings used in this study. These initial settings were selected based on Goldberg's simple GA (Goldberg 1989) and our previous experience in this domain (Moore and Hahn 2003a, 2003b, 2004a, 2004b). We ran the genetic algorithm a total of 100 times with different random seeds for each gene-gene interaction model. The genetic algorithm was stopped when a model with a classification error of zero was discovered (i.e. E = 0) or when 90% of the population converged, where convergence was measured by the genetic algorithm library used, GAlib (Wall 2003). We used a parallel search strategy (Cantu-Paz 2000) of ten demes, each with a population size of 500, for a total population size of 5000. A best chromosome migrated from each deme to all other demes every 25 generations.

Number of runs	100
Stopping criteria	Classification error = 0 or population convergence at 90% as defined by GAlib
Population size	5000
Number of demes	10
Generations (max.)	n/a
Selection	Roulette wheel
Crossover	Single point
Crossover probability	0.90
Mutation probability	0.01

Table 1: Summary of the genetic algorithm parameters used.

Results

The grammatical evolution algorithm was run a total of 100 times for each of the two nonlinear gene-gene interaction models. For both Model 1 and Model 2 (see Figure 1), the grammatical evolution strategy yielded an ABS model that was perfectly consistent with the high-risk and low-risk assignments for each combination of genotypes. Thus, the ABS model discovery method routinely found perfect models.

Figure 2 below illustrates the starting and ending configurations for an ABS that is consistent with genetic Model 2. Note that the ABS for genotypes *AAbb*, *AaBb*, and *aaBB* all have at least three agents in the upper left quadrant while there are less than three agents in the upperleft quadrant for all the other genotype combinations. This pattern of high-risk and low-risk agent counts is perfectly consistent with the high-risk and low-risk genotype combinations in genetic Model 2 and thus has a maximum fitness. This ABS was executed for 91 timesteps and had nine total agents. Six agents had movements that were dependent on genotype while five had collisions that were genotype-dependent. The agents

that finished in the upper left quadrant were all dependent on genotype.

Tables 2 and 3 below summarize the mode (i.e. most common) and range of the number of agents, collisions and types of moves and collision behaviors that define the genotype-dependencies of the elements in the best ABS models found across the 100 runs for each model. Agents are broken into categories according to their movement and collision behaviors. Locus independent (LI) and locus dependent (LD) move (M) and collision (C) behaviors are shown. As expected, every ABS had at least two agents that had moves or collisions that were locus dependent. For Model 1 and Model 2, the most frequent numbers of agents were six and five, respectively.

	Agents	LIM	LDM	LIC	LDC	C
Avg	4.5	1.3	3.2	2.3	2.2	49.3
Min	2	0	1	0	0	2
Max	12	5	9	9	6	533
Mode	6	0	2	2	2	2

Table 2: Summary of results from 100 runs for Model 1.

	Agents	LIM	LDM	LIC	LDC	C
Avg	5.1	1.3	3.7	2.5	2.5	67.3
Min	2	0	2	0	0	4
Max	13	5	8	8	7	521
Mode	5	0	2	3	3	6

Table 3: Summary of results from 100 runs for Model 2.

Discussion

The primary conclusion of this study is that it is possible to use artificial life in the form of agent-based simulation (ABS) to generate hypothetical systems biology models that are consistent with genetic models of disease susceptibility. This study represent the first step towards the use of artificial life to carry out thought experiments about the nature of the genotype to phenotype mapping relationship in the context of human health and disease. We anticipate a combined computational approach to both systems biology and human genetics will lead to a better understanding of human health which in turn will lead to better disease diagnosis, prevention, and treatment strategies.

We acknowledge that the approach presented here is only the first step and that additional changes to the overall modeling strategy will be needed before this prototype will be useful for routine thought experiments. For example, it will be important to incorporate measures of agent dynamics into the system. The current approach measures a static agent endpoint. It will also be important to incorporate measures of systems complexity such as entropy (reviewed by Adami 1998). A wider range of agent behaviors such as birth and death will need to be explored in addition to optimization of the grammatical evolution strategy for higher-order genetic models as has been done for the Petri net approach (Moore and Hahn 2004b). Further, higher-level measures of the system will need to be implemented to capture more interesting patterns of behavior such as dynamical hierarchies (e.g. Dorin and McCormack 2003). Finally, interpretation of ABS models will be necessary if useful biological hypotheses are to be generated from these thought experiments. We anticipate that this study will provide a useful starting point for those hoping to use artificial life models as hypothesis-generating thought experiments.

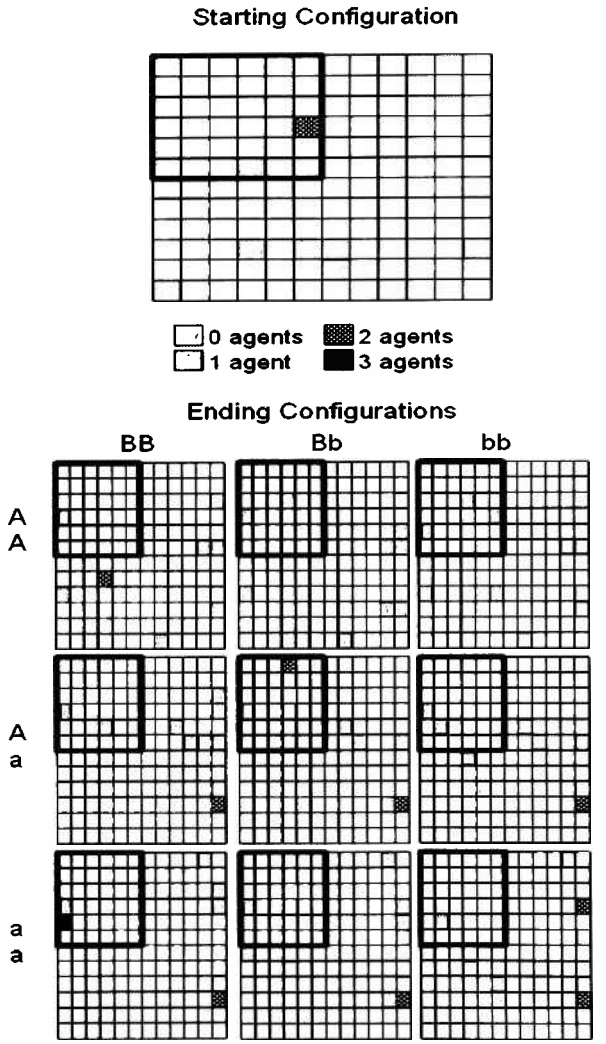

Figure 2: Starting (top) and ending (bottom) configurations for an ABS that is consistent with Model 2. In this ABS, each square in the 12x12 grid is occupied by either zero (white), one (grey), two (checkerboard), or three (black) agents. For simplicity, the types of agents are not illustrated. The upper left quadrant of the grid outlined in black is used to determine risk.

Acknowledgements

This work was supported by National Institutes of Health grants HL65234, HL65962, GM31304, AG19085, and AG20135.

References

Adami, C. 1998. *Introduction to Artificial Life*. TELOS Springer-Verlag, Santa Clara.

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

Cantu-Paz, E., 2000. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers.

Desel, J., Juhas, G. 2001. What is a Petri net? Informal answers for the informed reader. In H. Ehrig, G. Juhas, J. Padberg, and G. Rozenberg (Eds.), *Unifying Petri Nets, Lecture Notes in Computer Science* 2128, pp 1-27. Springer.

D'Inverno, M. and Luck, M. 2001. *Understanding Agent Systems*. Berlin, Springer.

Di Paolo, E.A., Noble, J., Bullock, S. 2000. Simulation models as opaque thought experiments. In: Dedau, M.A. et al. (Eds.), *Artificial Life VI*. The MIT Press, Cambridge.

Dorin, A., McCormack, J. Self-assembling dynamical hierarchies. In: Standish, R.K., Bedau, M.A., Abbass, H.A. (Eds.), *Artificial Life VIII*. The MIT Press, Cambridge.

Falconer, D.S., Mackay, T.F.C. 1996. *Introduction to Quantitative Genetics*. Longman, Essex.

Fisher, R.A. 1918. The correlation between relatives on the supposition of Mendelian inheritance. *Transactions of the Royal Society of Edinburgh* 52, 399-433.

Fogel, L.J. 1962. Autonomous automata. *Industrial Research* 4, 14-19.

Fogel, L.J., Owens, A.J., Walsh, M.J. 1966. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York.

Frankel, W.N., Schork, N.J. 1996. Who's afraid of epistasis. *Nature Genetics* 14, 371-3.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Goss, P.J., Peccoud, J. 1998. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proceedings of the National Academy of Sciences USA* 95, 6750-5.

Holland, J.H. 1969. Adaptive plans optimal for payoff-only environments. In: *Proceedings of the 2nd Hawaii International Conference on Systems Sciences*, pp 917-920.

Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Hopcroft, J. and Ullman, J. 1979. *Introduction to Automata Theory, Languages and Computation*. Reading, MA: Addison-Wesley.

Ideker, T., Galitski, T., Hood, L. 2001. A new approach to decoding life: systems biology. *Annual Review of Genomics and Human Genetics* 2, 343-72.

Jansen, R.C. 2003. Studying complex biological systems using multifactorial perturbation. *Nature Reviews Genetics* 4 , 145-51.

Kitagawa, J., Iba, H. 2003. Identifying metabolic pathways and gene regulation networks with evolutionary algorithms. In: Fogel, G.B., Corne, D.W. (Eds.), *Evolutionary Computation and Bioinformatics*. Morgan Kaufmann Publishers.

Koza, J.R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press.

Koza, J.R., Mydlowec, W., Lanza, G., Yu, J., Keane, M.A. 2001. Reverse engineering of metabolic pathways from observed data using genetic programming. *Pacific Symposium on Biocomputing* 6, 434-45.

Li, W., Reich, J. 2000. A complete enumeration and classification of two-locus disease models. Human Heredity 50, 334-49.

Merlin, P. 1974. *A Study of the Recoverability of Computer Systems*. Ph.D. Thesis, University of California, Irvine (1974).

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

Moore, J.H., Hahn, L.W. 2003a. Grammatical evolution for the discovery of Petri net models of complex genetic systems. In: Cantu-Paz et al. (Eds.), *Genetic and Evolutionary Computation – GECCO 2003. Lecture Notes in Computer Science* 2724, 2412-13.

Moore, J.H., Hahn, L.W. 2003b. Petri net modeling of high-order genetic systems using grammatical evolution. *BioSystems* 72, 177-86.

Moore, J.H., Hahn, L.W. 2004a. Evaluation of a discrete dynamic systems approach for modeling the hierarchical relationship between genes, biochemistry, and disease susceptibility. *Discrete and Continuous Dynamical Systems: Series B* 4, 275-87.

Moore, J.H., Hahn, L.W. 2004b. An improved grammatical evolution strategy for Petri net modeling of complex genetic systems. In: *Applications of Evolutionary Computing, Lecture Notes in Computer Science*, in press.

Moore, J.H., Hahn, L.W., Ritchie, M.D., Thornton, T.A., White, B.C. 2002. Application of genetic algorithms to the discovery of complex genetic models for simulation studies in human genetics. In: Langdon, W.B., et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann Publishers, San Francisco.

Moore, J.H., Williams, S.M. 2002. New strategies for identifying gene-gene interactions in hypertension. *Annals of Medicine* 34, 88-95.

O'Neill, M., Ryan, C. 2001. Grammatical evolution. *IEEE Transactions on Evolutionary Computation* 5, 349-358.

O'Neill, M., Ryan, C. 2003. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, Boston.

Rechenberg, I. 1965. *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment, Farnborough, U.K., Library Translation No. 1122, August.

Reif, D.M., White, B.C., Moore, J.H. 2004. Integrated analysis of genetic, genomic, and proteomic data. *Expert Review of Proteomics*, in press.

Schwefel, H.-P. 1965. *Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungstechnik*. Diploma Thesis, Technical University of Berlin.

Wall, M. 2003. GAlib. http://lancet.mit.edu/ga/

Woolridge, M. 2002. *An Introduction to MultiAgent Systems*. Sussex, England: Wiley.

Wooldridge, M., Jennings, N.R. 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review* 10, 115-52.