The Evolution of Emergent Organization in Immune System Gene Libraries

Ron R. Hightower
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
high@cs.unm.edu

Stephanie Forrest
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
forrest@cs.unm.edu

Alan S. Perelson
Theoretical Division
Los Alamos Nat. Lab.
Los Alamos, NM 87545 USA
asp@receptor.lanl.gov

Abstract

A binary model of the immune system is used to study the effects of evolution on the genetic encoding for antibody molecules. One feature of this encoding is that, unlike typical genetic algorithm experiments, not all genes found in the genotype are expressed in the phenotype. We report experiments which show that the evolution of immune system genes, simulated by the genetic algorithm, can induce a high degree of genetic organization even though that organization is not explicitly required by the fitness function. We hypothesize about the nature of this organization and introduce a measure called Hamming Separation to observe its change during the evolution of the immune system.

1 Introduction

How can selection pressures operating only on the phenotype drive evolutionary changes in the genotype? In contrast with typical genetic algorithm representations, in which all genes contribute to the calculation of fitness, the genetic material in natural organisms is not completely expressed. Some aspects of the genotype may not be reflected in the phenotype and are therefore hidden from selection pressure temporarily. The immune system provides a good subject for modeling and experimentation from this point of view—the genotype is never completely expressed in the phenotype, but the mapping from the genotype to the phenotype is simple enough that it can be effectively modeled in a computer simulation.

The immune system has an overall complexity that rivals that of the central nervous system. In this research we only consider the evolution of *antibody* molecules, which are responsible for recognizing foreign cells and molecules, called *antigens*. The first step in antigen recognition occurs when an antibody physically binds to an antigen molecule. This binding requires that the two molecules, antibody and antigen, have complementary shapes. Because the two molecules must "match" in order to bind, it would seem that every antigen requires a corresponding antibody molecule in order to be detected. An undetected antigen could cause infection, illness, or death, so a fit individual should have an immune system that can recognize all possible antigens.

However, there are an almost limitless number of possible antigens, and an individual has only limited genetic resources to allocate to the immune system. Both mice and humans, for example, have fewer than 10^5 genes in their entire genome, but their immune systems can make on the order of 10^{11} different antibody molecules (Berek and Milstein 1988, Darnell et al 1986). The mouse and the human immune systems both use a similar genetic mechanism to produce this large number of different antibody molecules. The genetic material for one antibody molecule is stored in five separate component libraries. Producing one antibody molecule begins with the random selection of a genetic component from each of the libraries. There are many possible combinations of the available components, so the immune system can generate a large number of antibodies, even though the libraries contain a limited amount of genetic information. As discussed later, this combinatorial mechanism is most effective when the variants (the different components in each library) are dissimilar. If all variants were the same there would be little advantage to interchangeability.

Because the components are randomly chosen, only a fraction of the available genetic material is expressed at any one time. Hence the phenotype of the immune system—the expressed antibody molecules—does not completely represent the genotype, which is the total collection of gene segments in all the libraries.

We have defined an abstract model of immune system libraries and used the genetic algorithm to simulate the evolution of individuals. Each individual represents the genetic specification for the antibody libraries of one immune system. In the first set of experiments we

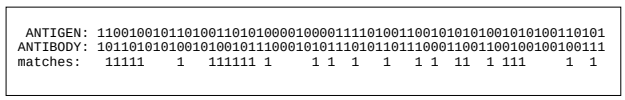

Figure 1: Binding/recognition process for binary molecules

observed the effects of partial gene expression and partial fitness evaluation on the average fitness of the population. Forcing the genetic algorithm to operate on the basis of this incomplete information causes some reduction in progress, but not as much as we might expect. The second set of experiments shows that the entries in the libraries become progressively more dissimilar under evolution, even though dissimilarity is not explicitly required by the fitness function. This organization of the libraries is an "emergent effect" that can be interpreted as a balanced allocation of antibodies to the task of antigen recognition. Sections 2 and 3 describe the artificial immune system model and briefly summarize the experiments that tested the performance of the model over varying rates of antibody expression and antigen exposure. In Sections 4 and 5 we take a closer look at the effects of evolution on the genotype. Section 4 motivates a measure of library organization called Hamming separation, and Section 5 experimentally compares this measure with the fitness of the immune system.

2 Artificial Immune System

In our simplified model of the immune system bit strings are used to represent both the genotype—libraries of gene segments—and the antibody molecules of the phenotype. For a binary molecule, the pattern of the bits represents the shape of the molecule, and the comparison of two binary molecules will determine their ability to bind. In our bit string universe, molecular binding takes place when an antibody bit string and an antigen bit string "match" each other, in the sense that they have complementary shapes (i.e., binary patterns). This reflects the lock-and-key fit of actual molecules during binding. This representation is loosely based on a bit string universe introduced by Farmer et al (1986). Figure 1 shows a binary antigen molecule and a binary antibody molecule.

The binding affinity between two molecules is computed by finding the number of bitwise complementary matches. The bits that match can be used to compute a "match score" in a variety of ways, but for the experiments reported here the match score is simply the sum of the number of matching bits. The match score between the two molecules in Figure 1, for example, is 27.

The bit string representing the genotype of an individual is divided into four equal-size libraries of antibody segments as shown at the top of Figure 2. Within each library there are eight elements, represented as bit strings of length sixteen, so each individual has a total of 512 bits. This structure is a simplified model of the human immune system which has seven libraries, each with a different number of gene segments (Leder 1982).

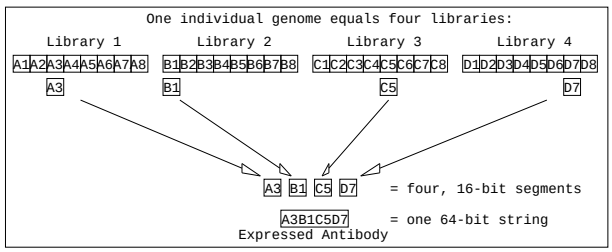

Figure 2: Process of constructing/expressing antibody from genetic library

The process of expressing an antibody is also shown in Figure 2. One segment is chosen randomly from each library and the four selected elements are concatenated into a single bit string that is sixty-four bits in length. We call this bit string an antibody molecule, one of several that will be used to compute the fitness of the individual. The set of antibodies that can be constructed from the libraries is called the *potential antibody repertoire*. Not every antibody from the potential repertoire is present in an individual at a given time. The set of antibodies that are expressed in the phenotype is called the *expressed antibody repertoire*.

The fitness of an individual is determined by its overall ability to recognize antigen molecules. Fitness is evaluated by exposing an individual to a set of antigens and testing how well it recognizes each antigen in that set. The expressed antibodies are responsible for the recognition task. Each antigen receives an *antigen score*, which is the maximum of all the match scores computed between that antigen and the expressed antibodies. The antigen score quantifies how well the immune system recognized that particular antigen. The overall fitness of the individual is found by combining the various antigen scores. The simplest method for computing the fitness, used here, is to average the scores for the different antigens. An alternative method is to use the minimum antigen score for the fitness, with the rationale that the antigen you are least equipped to recognize is the one that best characterizes the fitness of your immune system.

3 Stochastic Gene Expression and Stochastic Fitness Evaluation

In the experiments described in this section, our artificial immune system is evolved using the genetic al-

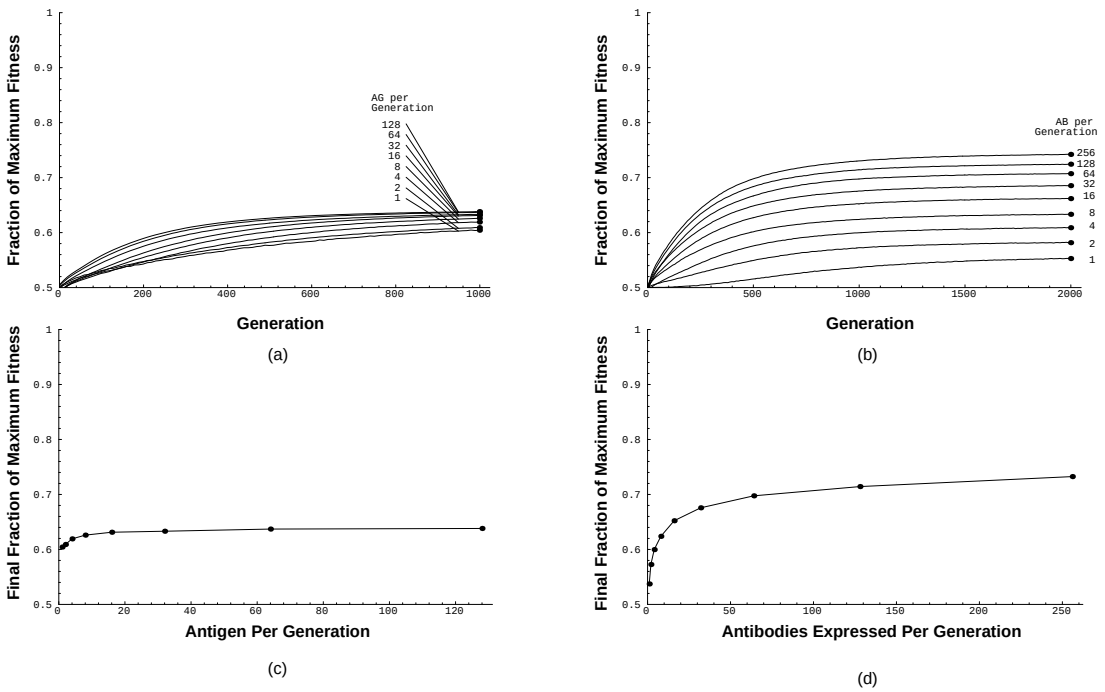

Figure 3: Stochastic Evaluation and Stochastic Expression.

gorithm (Holland 1975, Goldberg 1989, Forrest 1993). Fitness is computed according to an individual's ability to recognize antigen strings. In the initial experiments the fitness of individuals was evaluated on the basis of incomplete information, as is the case with real immune systems. First, each individual was exposed to only a subset of the existing antigens, modeling the fact that real individuals are not exposed to all diseases during their lifetimes. K represents the number of antigens to which an individual is exposed. Second, each individual was only allowed to express a fraction of its potential antibody repertoire. This sampling operation was motivated by the fact that at most 10^7 of the 10^{11} possible antibodies are present in the body, as expressed molecules, at any given time. N represents the number of antibodies expressed by an individual.

The experiments reported here were conducted with a modified version of the genetic algorithm package Genesis 1.2ucsd, (Grefenstette 1984). Population size was 500. The mutation rate was 0.002 and the crossover rate was 0.6. Sigma scaling was used. The population was initialized with all bits set to zero, rather than randomly initializing the population. This reflects the biological hypothesis that the antibody libraries originated through a process of gene duplication, which caused the library elements to be similar until mutation caused diversification. The antigen varying experiments extended for 1000 generations, and the antibody experiments extended for 2000 generations.

Genetic algorithm experiments were performed for various antigen exposure rates and antibody expression rates (i.e. varying K and N). Figure 3a shows the fitness trajectories of genetic algorithm experiments with antigen exposure, K, varying between 1 and 128, while the number of expressed antibodies, N, was held constant at 8. (Antigen bit strings were drawn from a predefined set called the "antigen universe.")[1] Figure 3b shows the fitness trajectories of genetic algorithm experiments with antibody expression, N, varying between 1 and 256, while the number of antigens exposed, K, was held constant at 8. Both antigens and antibodies were sampled with replacement. The plotted results are the mean values from 30 runs of the genetic algorithm.

Figure 3c compares the population averages at the end of the genetic algorithm experiments with various values for antigen exposure, and Figure 3d makes the same comparison for values of antibody expression. These experiments demonstrated that the genetic algorithm was capable of improving the fitness of the population, even when given sparse and incomplete information about the performance of individuals. The

[1] Increasing the size of the antigen universe makes the recognition task more difficult, and also increases the runtime of the simulation. We determined empirically (data not shown) that an antigen universe with 32 antigen bit strings was nearly as difficult to recognize as larger ones, yet made the experiments tractable.

next step was to explore what was happening within the components of the gene libraries and to determine the internal structure of a succesful immune system.

4 Coverage of Antigen Space

In this section we take a closer look at the nature of the antigen recognition task and discuss the type of solutions the genetic algorithm is finding. One perspective on the antigen recognition task is to consider the set of all possible antigens as a space of points, where antigen molecules with similar shape occupy neighboring points in the space. We call this *antigen space*. Because antigen molecules in the binary model are 64 bits in length, the total number of unique antigens is $2^{64} = 1.8 \times 10^{19}$, which is the size of this antigen space.

A given antibody molecule recognizes some set of antigens and therefore covers some portion of antigen space. The amount of coverage provided by one antibody is determined by the acceptable matching error. If no error is allowed during matching an antibody can only recognize the antigen that is its exact complement. If, however, the immune system is allowed to make a one-bit error during matching then each antibody can cover 65 antigens: the one antigen it matches exactly and the 64 antigens created by changing one of its 64 bits. The *error radius,* r, is the number of bits that may be in error during matching. The number of antigens covered by one antibody within a given error radius is:

$$coverage = \sum_{i=0}^{r} \left(\begin{array}{c} l \\ i \end{array} \right)$$

where l equals 64, the length of the bit strings. An error radius of two bits, for example, allows one antibody to cover $1 + 64 + 2016 = 2081$ antigens, while an error radius of 25 bits lets one antibody cover 9.5×10^{17} antigens, which is roughly 5 percent of antigen space. Figure 4a shows a stylized image of antigen space being covered by antibody molecules. The crosses denote the location of antigen molecules and the black dots are antibody molecules. The circles around the antibodies show the coverage each one provides for a given error radius. If the error radius were reduced then each antibody would provide less coverage.

Figure 4 can be used to discuss some important aspects of the immune system libraries, although both real antigen space and our model have a much higher dimensionality than the two-dimensional picture shows. First, note that every antibody is associated with a unique location in antigen space—the location of the antigen that has an exactly complementary shape. Second, the distance between two molecules in antigen space is equal to the Hamming distance between the two bit strings. Because the distance between two similar antibody molecules is small, such molecules would recognize many of the same antigens. Similar

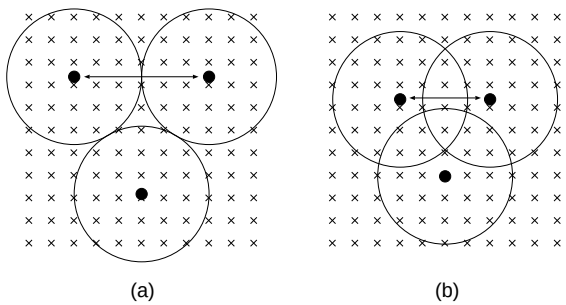

Figure 4: Coverage of antigen space by antibodies. X's represent a projection of antigens in antigen space. Circles represent the portion of antigen space recogized by the corresponding antibodies (black dots). In (a) the antibodies are far apart (dissimilar) and their areas of coverage do not overlap. This leads to gaps in the coverage. In (b) the antibodies are closer to each other, eliminating the intervening gaps, but leading to redundant coverage.

molecules would therefore have overlapping coverage in antigen space. Overlapping coverage is redundant and potentially reduces antibodies' capacity for recognizing antigen. Because the immune system only has a limited number of antibodies, it is desirable to reduce redundant coverage by arranging antibodies as far from each other as possible. This provides a possible way of indirectly measuring coverage, as discussed in Section 5.

Figure 4a suggests that if the Hamming distance between all antibodies is greater than or equal to the error radius, then gaps of coverage might exist. On the other hand, Figure 4b shows that if the Hamming distance between antibodies is less than twice the error radius (the radius of the circles in the figure) then their coverage will overlap.

Randomly generated 64-bit antibodies have an average Hamming distance of 32 bits. (Given one antibody, any other randomly chosen antibody will have a 50% chance of having the same value for any particular bit, so the two bit strings will differ on average in half their bits.) A set of randomly generated antibodies will tend to be an average of 32 bits from one another. We have found empirically that randomly generated antibodies provide a good coverage of antigen space (shown in 6).

5 Hamming Separation and True Fitness

Section 4 explained that similar antibodies have a small Hamming distance between them and this corresponds to an overlapping coverage of antigen space. As the antibodies become increasingly close together, the redundant coverage increases and their overall com-

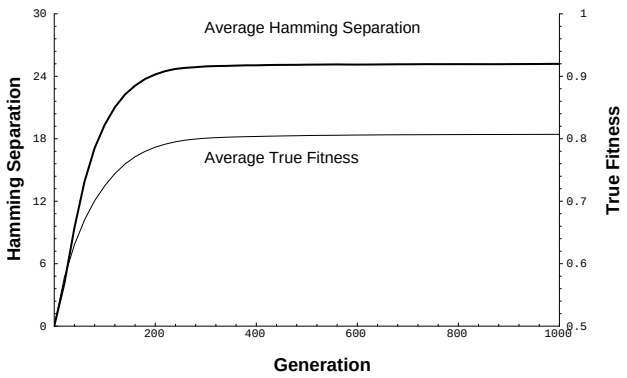

Figure 5: Hamming Separation for 30 Experiments

bined coverage is reduced. Compare Figure 4a with Figure 4b to see this effect.

As antibodies become increasingly further apart that the redundant coverage is reduced. We might predict that the maximum coverage of antigen space would be achieved when the antibodies were maximally distant. The amount of overlapping coverage would be minimized. This suggests an indirect method for measuring the efficiency of antigen coverage and a direct method of measuring genetic organization. The Hamming distance between pairs of antibodies, averaged over all pairs of antibodies in the repertoire, might have a high correlation with the fitness of the immune system. We call this measure of genetic organization *Hamming separation.*

Hamming separation compares all pairs of antibody bit strings, requiring N^2 comparisons, which can be computationally expensive for the size of the antibody repertoire, $N = 4096$, However, the gene libraries can be considered independently in this computation, so N^2 comparisons only need be made over the $N = 8$ elements in each library, times four libraries. So Hamming separation is computed by finding the average Hamming distance between all pairs of gene segments in each of the four libraries and summing the result. (In our model there are 8 elements in a library and each library elements is 16 bits in length. Recall, that the average Hamming distance between elements, for random libraries, is 50% of the bit string length, or 8 bits. For four libraries, then, the average distance would be 32 bits.)

The hypothesis is that the fitness of an immune system correlates with Hamming separation of its gene segment libraries. This was tested by running 30 experiments like those described in Section 3 with the additional computation of the Hamming separation and the true fitness of each individual. (True fitness is a complete evaluation of the genotype and considers all antibodies as well as all antigens.) For all 30 experiments each phenotype consisted of $N = 8$ randomly expressed antibodies, which was then exposed to $K = 8$ randomly chosen antigens. Figure 5 shows the results of this experiment. The graph shows that the Hamming separation gradually improves as the genetic algorithm progresses, as does the fitness trajectory of the various experiments shown in Figure 3. What is the relationship between Hamming separation and the true fitness of the immune system?

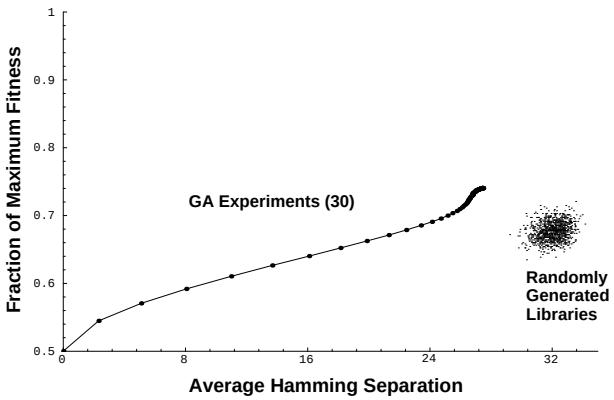

Figure 6: True Fitness versus Hamming Separation. Average trajectory for 30 genetic algorithm experiments. Compare with the region of randomly generated individuals.

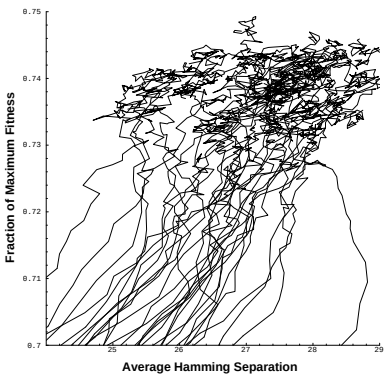

Figure 7: Expanded view of the attractor for the genetic algorithm experiments.

A comparison of Hamming separation with average true fitness is shown in Figure 6. The curve shows the average trajectory of 30 genetic algorithm experiments through this fitness/separation space. (The experiments were sampled every 20 generations, shown as black points along the curve.) As mentioned previously the experiments begin with the individuals in the population initialized to all zeros. This means that all antibodies are initially zero, so the Hamming separation begins at zero and the initial fitness is only fifty percent. Both the fitness measure and Hamming

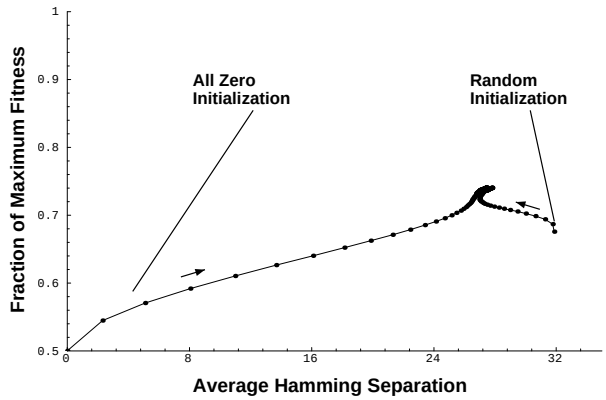

Figure 8: Trajectories for genetic algorithm experiments with individuals initialized to all zeros compared with randomly initialized populations.

separation improved steadily during the experiment, ending with a true fitness around 0.74 and a Hamming separation near 27.5.

Figure 7 shows an enlarged view of the region where the genetic algorithm experiments terminate. Each path in the graph is a trajectory followed by one of the 30 experiments. This graph shows that the genetic algorithm repeatedly converges to a particular region of the fitness/separation space. In all 30 experiments the genetic algorithm reaches this region by generation 300 and never leaves, even after generation 2000 (Figure 5 also shows that the experiments have converged).

The relationship between true fitness and the Hamming separation measure is very nearly linear. So our hypothesis appeared to be correct: the antibodies libraries evolve towards a good coverage of antigen space and the degree of this coverage can be quantified using the Hamming separation measure.

However, there was a problem reconciling these results with what we knew about randomly generated individuals. We knew from earlier results that the genetic algorithm was finding solutions with fitness values above those of randomly generated individuals. But we also knew that the expected Hamming separation for such randomly generated individuals is 32 bits, while the Hamming separation of evolved individuals was converging to a value near 27.5. For comparison we generated random genomes for five hundred individuals, and computed their true fitnesses and Hamming separation values. The results are shown in Figure 6, as a cloud of points in the lower right corner, far from the trajectories of the genetic algorithm experiments. The average fitness of the randomly generated individuals is about 0.67 of the maximum true fitness.

Although Hamming separation and true fitness are highly correlated for the individuals from the genetic algorithm experiments, the relationship is not as strong with respect to randomly generated individuals. Additional experiments show that genetic algorithm runs starting with randomly generated individuals follow the trajectory shown in Figure 8. This suggests that the optimal solution to the antigen recognition task is not simply a maximization of the Hamming separation measure, as originally hypothesized. Rather, an optimal solution requires a balance between the conflicting issues of coverage redundancy and coverage gaps, discussed earlier.

6 Conclusions

The artificial immune system model uses a binary representation for both molecular interaction and the genetic encoding of individuals. The interaction between antigen and antibody molecules in this representation is sufficiently complex to exhibit interesting behavior, without being so complex as to be computationally intractable. The library mechanism for storing antibody components is a simplified version of the real immune system and exhibits a non-trivial mapping from genotype to phenotype. This binary model allows us to study concepts like the coverage of antigen space and genetic organization with a manageable amount of complexity.

The genetic algorithm experiments with the artificial immune system show that the genetic algorithm can optimize complex genetic information. In fact the genetic algorithm has been able to organize the complex structure of the antibody libraries acting on the basis of incomplete fitness information. We have also shown that selection pressure operating on the phenotype as a whole can translate to selection pressure acting on individual genes, even though not all genes are expressed in the phenotype.

By considering the antigen recognition task in terms of a spatial coverage problem we were able to devise a measure of organization for the immune system libraries. The Hamming separation measure provided a simple preliminary tool for observing the effects of evolution on the genotype of the immune system. This measure was shown to have a high correlation with the true fitness of the population, verifying the "space coverage" hypothesis, and providing additional evidence that the genetic information is undergoing implicit organization not directly required by the fitness function.

Acknowledgments

We thank the Center for Nonlinear Studies, Los Alamos National Laboratory and the Santa Fe Institute for ongoing support of this project. Forrest also acknowledges the support of the National Science Foundation (grant IRI-9157644). Perelson acknowl-

edges the support of the National Institutes of Health (grants AI28433)

References

[1] C. Berek and C. Milstein. The dynamic nature of the antibody repertoire. *Immunological Reviews*, **105**:5–26, 1988.

[2] J. Darnell, H. Lodish, and D. Baltimore. *Molecular Cell Biology.* Scientific American Books, NY, 1986.

[3] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. In D. Farmer, A. Lapedes, N. Packard, and B. Wendroff, editors, *Evolution, Games and Learning.* North–Holland, Amsterdam, 1986. (Reprinted from *Physica* **22D**:187–204, 1986).

[4] P. Leder. The Genetics of Antibody Diversity. In W.E. Paul, editor, *Immunology: Recognition and Response.* W. H. Freeman, New York, 1991.

[5] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, MI, 1975.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley, Reading, MA, 1989.

[7] S. Forrest. Genetic Algorithms: Principles of Adaptation Applied to Computation. *Science*, **261**:872-878, 1993.

[8] J. J. Grefenstette. *GENESIS: A System for Using Genetic Search Procedures.* In *Proceedings of a Conference on Intelligent Systems and Machines.* 161–165, 1984.