

# The Evolution of Secondary Organization in Immune System Gene Libraries

Ron Hightower  
Dept. of Comp. Sci.  
University of New Mexico  
Albuquerque, NM 87131 USA  
high@unmvax.cs.unm.edu

Stephanie Forrest  
Dept. of Comp. Sci.  
University of New Mexico  
Albuquerque, NM 87131 USA  
forrest@unmvax.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. 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. This secondary organization is related to the *true fitness* of an individual, in contrast to the *sampled fitness* which is the explicit fitness measure used to drive the process of evolution.

Keywords: immune system, genetic algorithm, V-region gene libraries

## 1 Introduction

The interplay between concrete actions at a local level and emergent behaviors at the global level is one of the major themes of artificial life. In the context of evolution, an important question is how selection pressures operating only at the global, phenotypic level can produce appropriate low-level, genetic structures. This question is most interesting when the connection between phenotype and genotype is more than a simple, direct mapping. The immune system provides a good subject for experimentation from this point of view—the phenotype is not a direct mapping from the genotype but the connection is simple enough that it can be studied.

In order to defend against foreign cells and molecules, called antigens, an immune system must first be able to recognize them. Antibody molecules are one of the agents responsible for antigen recognition. Recognition is achieved when an antibody physically binds to an antigen molecule. Molecular 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.

There are, however, an almost limitless number of antigens to recognize, 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 [1, 2]. Both the mouse and the human immune system use a collection of gene libraries to code for components of antibody molecules. Because the components can be combined in a large number of ways to produce an antibody, these immune systems can generate a large number of antibodies, even though the libraries contain only a small amount of genetic information.

Each antibody molecule, for example, is composed of two types of polypeptide chains: the heavy chain (H) and the light chain (L). If the immune system could construct  $10^4$  different light chains and  $10^4$  heavy chains, then the random combination of light and heavy chains would allow the construction of  $10^8$  different antibodies. The chains themselves are constructed from interchangeable components. The heavy and light chains both contain a variable (V) region of about 100–110 amino acids that differ from one antibody to the next. The structure of the antibody V-region is encoded by multiple gene segments, whereas most biological molecules are encoded by a single contiguous length of DNA. The V-region of the heavy chain, for example, is encoded using three gene segments, each of which has a number of different variants. Every combination of gene segments produces a unique V-region, so the large number of genetic combinations makes it possible to construct large number of different V-regions. All the interchangeable variants of a gene segment are stored in a library of gene segments. (Immunologists call these libraries multigene families.)

The gene segments are combined together before translation to an amino acid sequence takes place. For example, the variable region of the heavy chain is constructed by selecting one gene segment from each of three libraries, combining the three segments into a single piece of DNA, and then constructing from that strand the amino acid sequence which is the final heavy chain. The V-region of the light chain is made in an analogous way but it is constructed from only two gene segments, each with their own libraries. The V-regions for an antibody, then, are encoded by five different gene segments, each drawn from a separate gene library. When gene segments are combined, nucleotides can be added or deleted at the junctions adding a another level of diversity called junctional diversity. This additional mechanism for achieving diversity will not be considered in the model presented here.

By constructing antibodies from separate gene segments, each of which has a number of possible variants, the immune system leverages a small amount of genetic material to create a large number of antibody molecules. As will be argued later, this combinatorial mechanism is most effective when the variants (referred to as entries in the library) are dissimilar. If all variants were the same there would be little advantage to interchangeability. To study this effect, we have defined a simplified model of an immune system, and used the genetic algorithm to evolve individuals (each individual represents the genetic specification for one immune system). Our experiments show that the entries in the libraries become progressively more dissimilar under evolution, even though dissimilarity is not directly required by the fitness function. This organization of the libraries is a "secondary effect" that can be interpreted as a balanced partitioning of the antigen recognition task.

The organization of the libraries is a genotypic effect that is caused by selection pressure on the phenotype. The organization is implicit while the selection process is an explicit action. This distinction between phenotype (the aggregate level at which selection takes place) and

genotype (the level at which variation takes place) is a hallmark of artificial life systems. Our immune system model illustrates the explicit/implicit theme in two ways. First, the secondary organization of the immune system libraries is necessary because antibodies must *collectively* be able to recognize all antigens. The interdependence among components is a secondary organization, and is not measured directly. Rather, an individual's fitness is evaluated according to how well it matches randomly selected antigens. Secondly, only a small fraction of an individual's possible antibodies are *expressed* at any one time, yet an individual's fitness is determined by how well its expressed antibodies match the presented antigens. In some cases, an individual may be "unlucky" in the sense that it has the genetic material to match an antigen, but that material was unexpressed at the time the antigen was presented. As we will see, our model shows a separation between genotype and phenotype in a highly simplified setting, which allows us to quantify the effect.

In Sections 2 and 3, we review our artificial immune model and summarize earlier experiments which tested the performance of the model on various antigen recognition tasks. These experiments demonstrated the capability of a library mechanism for encoding antibody genes and showed that the genetic algorithm could optimize the antigen recognition capability of the model. However, it was not clear from these experiments exactly how the system evolved as well as it did. In the extended experiments described in Section 4 we explore the behavior of the model more carefully. Specifically, we study the relationship between *sampled fitness* (an incomplete testing of an individual's fitness that guides the selection process) and a complete measure of the individual's fitness which we call *true fitness*. Finally, in Sections 5 and 6, the effects of evolution on the genome are considered. Section 5 motivates a measure of library organization called Hamming separation, and Section 6 experimentally compares this measure with true fitness.

## 2 Artificial Immune System

Our simplified model of the immune system uses bitstrings to represent molecules and the gene segment libraries. The patterns of the bits represent the shapes of molecules and determine their ability to bind with other molecules. This representation is loosely based on a bitstring universe introduced by Farmer et al [3]. In our bitstring universe, molecular binding takes place when an antibody bitstring and an antigen bitstring "match" each other. A match occurs when the antigen and antibody have complementary shapes (i.e., binary patterns), which reflects the lock-and-key fit of actual molecules during binding. Figure 1 shows a binary antigen molecule and a binary antibody molecule. The binding affinity between real antigens and real antibodies is primarily determined by molecular shape and physical properties such as electrostatic surface charge, both of which are complementary when the molecules have a high affinity.

Matching is not required to take place perfectly along the entire length of the molecules. The exclusive-or operator (XOR) is used to compute which bits are complementary matches between the two molecules. The bits that match can be used to compute a "match score" in a number of different ways. In the experiments described here, the match score is simply

the sum of the number of matching bits. In the figure, for example, the XOR operator has found 27 bits that are complementary between the antigen and the antibody, so the match score is 27.

Antigen:	1100100101101001101010000100001111010011001010101001010100110101
Antibody:	0011010101001010010111000101011101011011100011001100100100100111
XOR:	11111 1 111111 1 1 1 1 1 1 1 1 1 1 1 1

Figure 1: Binding/recognition process for binary molecules

## 2.1 Antibody Libraries

Each individual in the simulated population contains four equal-size libraries of antibody segments as shown at the top of Figure 2. Within each library there are eight elements, represented as bitstrings 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<sup>1</sup> libraries, each with a different number of gene segments [4].

The *expression* of an antibody is also shown in Figure 2. One segment from each library is chosen, usually at random, and the four selected elements are concatenated into a single bitstring that is sixty-four bits in length. We call this bitstring 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 have currently been expressed 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

<sup>1</sup>The are two types of light chains,  $\lambda$  and  $\kappa$ , each of which has two V-region libraries. So while a given antibody is the product of gene segments from only five libraries, each cell contains a total of seven V-region libraries.

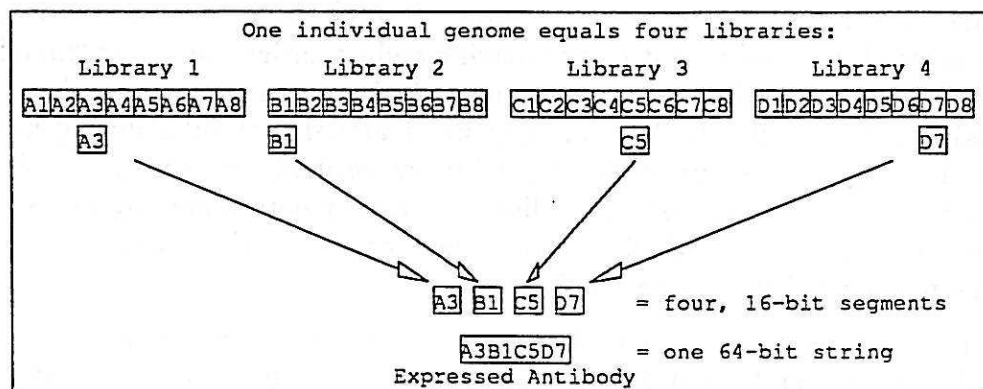


Figure 2: Process of constructing/expressing antibody from genetic library



recognizes each antigen in that set. The expressed antibodies are used to do the recognition. 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.

## 2.2 The Genetic Algorithm

The effects of evolution are simulated on the binary immune system by using the genetic algorithm, a computational model of genetic evolution [5, 6]. A population of individuals is represented in the computer as bitstrings. At each generation the population is evaluated according to some measure of fitness. A new population is formed from the evaluated population, where the individuals with higher fitness have more offspring than the less fit individuals. This cycle of evaluation and reproduction continues, and through time the average fitness of the population increases. Two genetic operators, crossover and mutation, modify the contents of the population as the genetic algorithm progresses. Crossover combines the binary patterns of two individuals into a new individual, whereas mutation changes the bits of an individual with some small probability. A discussion of genetic algorithm methodology is found in [6]. The experiments reported here were conducted with Genesis 1.2ucsd, which is a genetic algorithm tool written in C [11].

## 3 Previous Results

In earlier experiments the artificial immune system was used to test whether the genetic algorithm could evolve the gene libraries effectively [9],[10]. Preliminary experiments showed that the genetic algorithm could easily evolve an immune system (one using gene libraries) that recognized 100 percent of all possible antigens. Thus, even though the genetic representation of antibodies was complex, it was possible to optimize the antigen recognition task. This first experiment, however, was based on perfect information from the environment, so the recognition task was not as difficult as that faced by the real immune system.

In the next set of experiments the evaluation of individual fitness was subjected to two types of sampling noise, simulating the incomplete information available to real immune systems. First, each individual was exposed to a only subset of the existing antigens, modeling the fact that real individuals are not exposed to all diseases during their lifetimes. Second, each individual was only allowed to express a fraction of their 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.

Genetic algorithm experiments were performed for various antigen exposure rates and antibody expression rates. This type of partial evaluation of the fitness, due to sampling noise, reduces the efficiency of the selection process and the rate of evolution is slowed. Both sets of experiments showed that even with sparse and incomplete information, the immune system

libraries could evolve and make continued improvement in overall fitness. This result holds across a wide range of sampling rates, with the implication that the mechanism for gene libraries is robust and not a fragile construct.

The question arose as to how well the libraries that were evolved in the later experiments compared with the first libraries that were evolved using perfect information. That question is the basis for this paper, where the sampled fitness, based on incomplete information, is compared with the true fitness that is based on perfect information.

## 4 True Fitness vs. Sampled Fitness

In general, *true fitness* can be defined as an individual's fitness when evaluated in all possible conditions. Within the context of the immune system model, true fitness is an individual's ability to recognize all possible antigens using its entire potential antibody repertoire. As the name suggests, *sampled fitness* measures an individual's fitness for only a sample of possible environmental conditions. In the context of the immune system model sampled fitness is an individual's ability to recognize those antigens it stochastically encounters, using only that portion of the antibody repertoire it happens to express (the expressed antibody repertoire). Thus, sampled fitness is only an approximation of true fitness. We would like to know how well this approximation works when combined with the processes of evolution.

In the real world, true fitness is clearly a fiction. An individual would have to relive its life many times in all possible circumstances so that its fitness could be completely tested. While this is impossible in the real world it is feasible for the artificial immune system. True fitness is computed by expressing the entire potential antibody repertoire and using the highest match score found for each antigen being recognized.

In the experiment described in this section, our artificial immune system is evolved using the genetic algorithm. Fitness is computed according to an individual's ability to recognize antigen strings. An individual expresses a small subset of antibodies from its potential repertoire of antibody molecules. Then for each antigen presented to it, the individual selects the expressed antibody that best recognizes the antigen, and receives an antigen match score. The antigen match score, averaged over the set of antigens it encounters, becomes the individual's sampled fitness. The genetic algorithm determines an individual's reproductive future based on the sampled fitness.

The experiments used a population size of 500 individuals, and all experiments were run for one thousand generations. Instead of initializing the population with random bitstrings, as is common practice for GA experiments, the population began as all zero-valued bits (the reason for this is given in the next section).

We compare sampled fitness with the true fitness and the results are shown in Figure 3. For this experiment, sampled fitness is computed by expressing only eight antibodies of the 4096 in the potential repertoire. Figure 3 shows the population average for true fitness and the population average for sampled fitness. These curves have been averaged over thirty experiments. Initially the population contains individuals that are all zero bits, so the fitness

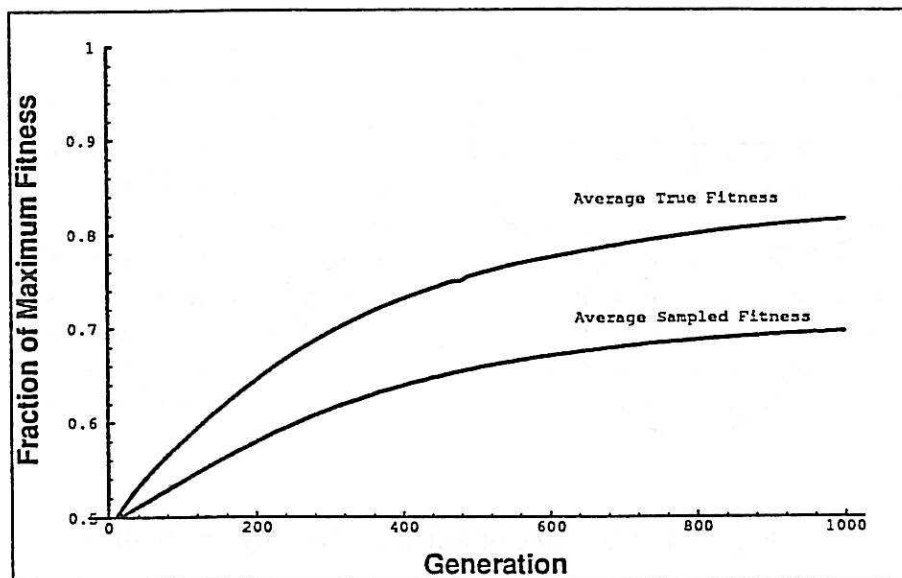


Figure 3: Sampled Fitness Curves and True Fitness Curves

of the population begins at 50 percent, i.e. no better than fair guessing.

In general, the true fitness will be higher than the sampled fitness, as is shown by the experiment. When computing the sampled fitness, the best antibody for recognizing a given antigen will not always be expressed, so with some fixed probability a less appropriate antibody will be used instead. For the true fitness, however, all antibodies are always expressed from the potential repertoire, so the best antibody is always available. Note that the ratio of sampled fitness to true fitness remains almost constant at 0.615 throughout the experiment (ratio taken with respect to the 50% fitness level:  $ratio = (sample\ fitness - 0.5) / (true\ fitness - 0.5)$ ).

## 5 Coverage of Antigen Space

The set of all possible antigens is called *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 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

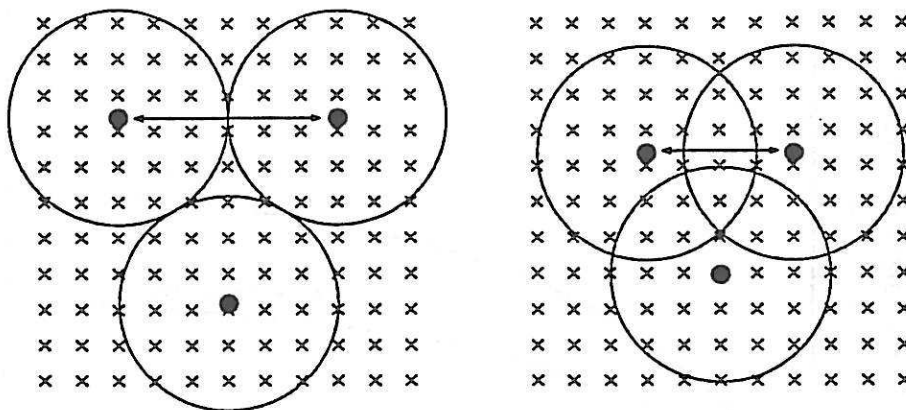


Figure 4: Coverage of antigen space by antibodies

number of antigens covered by one antibody within a given error radius is:

$$\text{coverage} = \sum_{i=0}^r \binom{l}{i}$$

where  $l$  equals 64, the length of the bitstrings.

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  $9.5 \times 10^{17}$  antigens, which is roughly 5 percent of antigen space. Figure 4a shows an image of antigen space being covered by antibody molecules. The crosses are antigen molecules while 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. 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 number of bits by which they differ. This is called Hamming distance.

Now, because the distance between two similar antibody molecules is small, such molecules would recognize many of the same antigens. Similar molecules would therefore have overlapping coverage in antigen space. Overlapping coverage is redundant and reduces the usefulness of an antibody. 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 6.

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



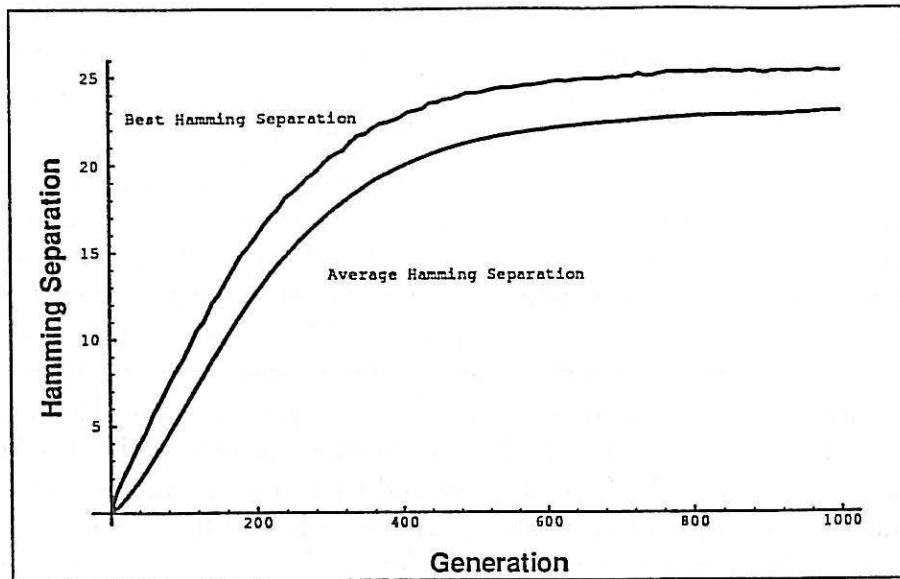


Figure 5: Hamming Separation for Thirty Experiments

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 bitstrings will be different on an average of half their bits.) Now, if a set of antibodies are randomly generated they will tend to be an average of 32 bits from one another, which represents a good of antigen space. So because randomized individuals provide an unfair head start towards covering antigen space, the experiments described in Section 4 do not begin with randomly generated individuals. Another motivation for initializing the population to all zero-valued individuals is that real immune system libraries probably developed through a process of gene duplication. Because all the genes may have derived from the same parent copy the immune system would have had an initial homogeneity across gene segments. This is duplicated in the model by using all zero-valued bits in the initial population.

## 6 Hamming Separation vs. True Fitness

Section 5 explained that similar antibodies have a small Hamming distance between them and this corresponds to an overlapping coverage of antigen space. If the antibodies become increasing closer together the redundant coverage increases and their overall combined coverage is reduced. Compare Figure 4a with Figure 4b to see this effect.

If the converse were true, the maximum coverage of antigen space would be achieved when the antibodies were maximally distant. The antibodies would be far apart and 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 true fitness.

This new measure of genetic organization was called the *Hamming separation*. In general the Hamming separation would be computationally expensive as a way of measuring organization because it requires  $N^2$  comparisons and the size of the antibody repertoire is  $N = 4096$ . However, the library encoding of antibody components makes it possible to compute Hamming separation with fewer computations. The elements of the libraries contribute independently to the overall coverage so they can be independently analyzed (see Figure 2). Although this still entails  $N^2$  comparisons,  $N$  is now equal to 8, the size of the 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 bitstring length, or 8 bits. For four libraries, then, the average distance would be 32 bits.)

The hypothesis that true fitness correlates with Hamming separation was tested by running the same thirty experiments described in Section 4 with the additional computation of the Hamming separation. 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 shown in Figure 3. What is the relationship between Hamming separation and the true fitness?

Figure 6 shows a graph comparing Hamming separation with average true fitness. Each line in the graph is one of the thirty experiments. (The experiments are shown separately to convey the distribution.) 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 separation improved steadily during the experiment, ending with a true fitness around 0.8 and a Hamming separation near 22.

The relationship between true fitness and the Hamming separation measure appears to be nearly linear for the GA experiments. Do randomly generated individuals also fall on this nearly linear curve? To test this question five hundred individuals were randomly generated and tested for true fitness and Hamming separation. The results are also shown in Figure 6, as a cloud of points in the lower right corner. The average fitness of these random individuals seems to be around 0.67 and the average Hamming Separation is 32.

The data from the random individuals shows that the near linearity of the relationship is not absolute as suggested by the data from the GA experiments. The cloud of points is well separated from the trajectory of the GA data points. However, the correlation between the true fitness and the Hamming separation measure is still high, thus validating, at least partially, the hypothesis that fitness is related to separation of the antibodies. Also, the fact that not all data points formed a perfectly linear relationship may reflect on the particular method for measuring Hamming separation, rather than on the hypothesis itself. It may be the case that simply averaging the the Hamming distances between the elements of a library is not a sufficiently accurate description of antibody separation.

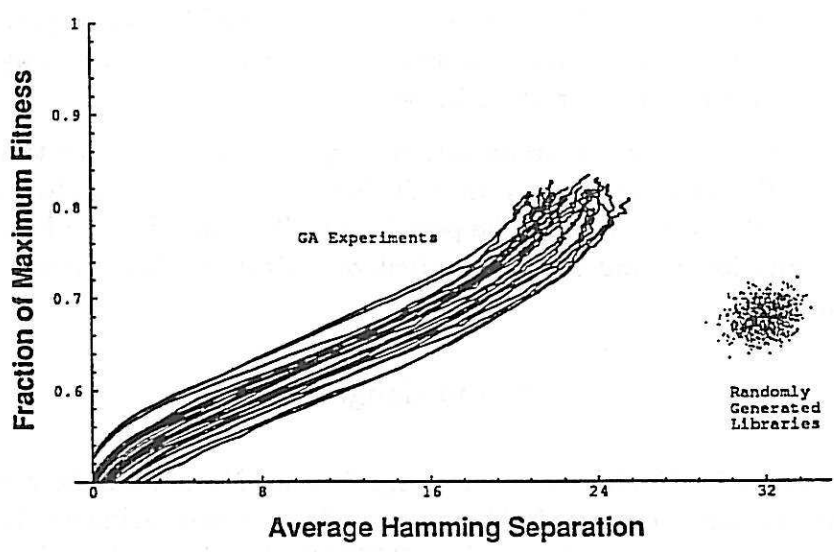


Figure 6: True Fitness versus Hamming Separation

The comparison of true fitness and Hamming separation shows organization taking place at the genetic level as a result of the selection process taking place at the phenotypic level. Hamming separation is a genetic measure, while true fitness is an evaluation of the phenotype. Organization at the genetic level is a secondary effect in the evolution of the artificial immune system, one that is not explicitly required by the fitness function.

## 7 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 GA experiments with the artificial immune system show that 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 only on the basis of sampled fitness. The organization of genetic material was shown in two ways. First, a distinction was made between true fitness and sampled fitness, showing that the genetic algorithm was operating at one level and producing results at a second level. The distinction between true fitness and sampled fitness becomes important for the evolution of complex systems acting in complex environments, such as in most Artificial Life models. True fitness could be a useful tool for monitoring the progress of more complex GA experiments. However, true fitness is

computationally expensive and in general would be infeasible to compute. One advantage of the artificial immune system model is that it is simple enough to study true fitness, but just complex enough to have interesting behavior.

The second way of observing the organization of genetic material was through the use of a special measure called Hamming separation. This measure was shown to improve in a steady fashion along with the true fitness of the population. This provides additional evidence that the genetic information is undergoing implicit organization than 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 acknowledges the support of the National Institutes of Health (grants AI28433 and RR06555).

### 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] A. S. Perelson and G. F. Oster. Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-non-self discrimination. *J. Theoret. Biol.* 81:645-670, 1979.
- [8] A. S. Perelson. Theoretical immunology. In E. Jen, editor, *1989 Lectures in Complex Systems, SFI Studies in the Sciences of Complexity, Lect. Vol. II*. Addison-Wesley, Reading, MA, pp. 465-499, 1990.



- [9] R. Hightower, S. Forrest, and A. S. Perelson. The Evolution of Cooperation in Immune System Gene Libraries. Technical Report CS92-14, Univ. of New Mexico, Dept. of Computer Science, University of New Mexico, Albuquerque, Oct. 1992. Submitted to the International Conference on Distributed Computing Systems.
- [10] R. Hightower, S. Forrest, and A. S. Perelson. Evolution of Multi-Gene Families in a Binary Immune System. Manuscript in preparation, 1993.
- [11] J. J. Grefenstette. *GENESIS: A System for Using Genetic Search Procedures*. In *Proceedings of a Conference on Intelligent Systems and Machines*. 161-165, 1984.