

# Original Investigations

## JAMIA

Research Paper ■

## The Immune System as a Model for Pattern Recognition and Classification

JEROME H. CARTER, MD

**Abstract Objective:** To design a pattern recognition engine based on concepts derived from mammalian immune systems.

**Design:** A supervised learning system (Immunos-81) was created using software abstractions of T cells, B cells, antibodies, and their interactions. Artificial T cells control the creation of B-cell populations (clones), which compete for recognition of “unknowns.” The B-cell clone with the “simple highest avidity” (SHA) or “relative highest avidity” (RHA) is considered to have successfully classified the unknown.

**Measurement:** Two standard machine learning data sets, consisting of eight nominal and six continuous variables, were used to test the recognition capabilities of Immunos-81. The first set (Cleveland), consisting of 303 cases of patients with suspected coronary artery disease, was used to perform a ten-way cross-validation. After completing the validation runs, the Cleveland data set was used as a training set prior to presentation of the second data set, consisting of 200 unknown cases.

**Results:** For cross-validation runs, correct recognition using SHA ranged from a high of 96 percent to a low of 63.2 percent. The average correct classification for all runs was 83.2 percent. Using the RHA metric, 11.2 percent were labeled “too close to determine” and no further attempt was made to classify them. Of the remaining cases, 85.5 percent were correctly classified. When the second data set was presented, correct classification occurred in 73.5 percent of cases when SHA was used and in 80.3 percent of cases when RHA was used.

**Conclusions:** The immune system offers a viable paradigm for the design of pattern recognition systems. Additional research is required to fully exploit the nuances of immune computation.

■ JAMIA. 2000;7:28–41.

Affiliation of the author: University of Alabama–Birmingham, Birmingham, Alabama.

Correspondence and reprints: Jerome H. Carter, MD, Director, Section of Medical Informatics, Division of General Internal Medicine, University of Alabama–Birmingham, 614 Medical

Education Building, 1813 6th Avenue South, Birmingham, AL 35294.

e-mail: (jcarter@gim.dom.uab.edu).

Received for publication: 2/22/99; accepted for publication: 9/20/99.

The immune system, much like the central nervous system, is capable of performing complex information processing tasks. At the very highest level it recognizes foreign molecules, referred to as antigens, and clears them from the organism. While performing this essential task, it learns the structure of the antigen and retains a memory of its amino acid sequence for future encounters. Over the last 15 years, a good deal of work has been done developing computer models that mimic various aspects of the immune system and are capable of performing practical information processing tasks. Even with the progress that has been made building artificial immune systems, much remains to be done to fully exploit this rich paradigm. In this paper a type of artificial immune system (Immunos-81) is proposed, which has powerful pattern recognition and classification features yet is easy to build and train. Prior to a discussion of design and performance of Immunos-81, a review of pertinent features of mammalian immune systems is offered along with an overview of existing artificial immune system implementations. For a more extensive discussion of the immune system, see Benjamin et al.<sup>1</sup>

## Background

### Overview of the Immune System

The main information processing activities of the immune system occur as the result primarily of the interactions of two types of cells—antigen-presenting cells (APCs) and lymphocytes. Lymphocytes exist in two forms, B cells and T cells. T cells (so called because they develop to maturity in the thymus gland) direct the response to protein antigens. The creation of mature T cells is a fairly random process. Progenitor cells migrate to the thymus from the bone marrow. There they undergo a two-stage selection process. The first, positive selection, acts on the entire maturing T-cell population, and only cells that demonstrate functional surface receptors capable of recognizing major histocompatibility (MHC) molecules continue maturing; the remaining cells undergo apoptosis (programmed cell death). Those that survive this process are then exposed to “self” antigens. Cells that demonstrate a response to self antigens are then negatively selected and permitted to die. As many as 98 percent of T cells do not make it through this two-stage selection process. Still, approximately  $10^6$  mature T cells are released into the circulation every day. Each T cell carries a unique set of surface receptors that are capable of recognizing antigens.

B cells mature in the bone marrow and carry immunoglobulin (IgM and IgD) on their surfaces. All im-

munoglobulins on a given cell are identical and are capable of binding antigen. B cells are produced at a rate similar to T cells. The two types of lymphocytes are distributed differently in the body. Ninety percent of T cells circulate, whereas 90 percent of B cells are found in secondary lymph tissue (e.g., lymph nodes, spleen). Antigen-presenting cells (e.g., macrophages, dendritic cells, Kupffer cells) are found throughout the body—some are fixed; others, such as macrophages, roam freely.

Before T cells can respond to an antigen, the antigen must be presented in a very specific manner. Antigen-presenting cells perform this vital function. They ingest foreign proteins, break them down and, as a final step, attach them to MHC molecules prior to displaying the antigen-MHC complex on the cell's surface. There are two types of MHC molecules. The MHC I molecules are found on all cells and attract the CD8+ (killer) subpopulation of T cells. Cells that express MHC I molecules with foreign (e.g., virus-infected) proteins attached are destroyed by the responding T cells. Antigens bound to MHC II molecules attract CD4+ (helper) T cells, and it is this subpopulation that interacts with B cells to create a full-blown immune response. Notice that B cells express MHC II molecules on their surfaces along with immunoglobulins and other important cell surface receptors.

When a B cell or T cell encounters an antigen that has sufficient affinity for its surface receptors, the cells become activated. However, antigen binding is a necessary but not sufficient condition for promulgation of the immune response. In addition, a B cell and T cell that recognize the *same* antigen must undergo direct physical contact with each another. This contact involves multiple receptors on the surfaces of both cells. Once this occurs, the B cell and T cell divide and the next phase of the response begins. B and T cells, which interact as previously described, produce progeny, referred to as clones, which express surface receptors identical to those of their parents. This response seems to be proportional to the degree of affinity of the original receptor for the provoking antigen. Thus, those B- and T-cell pairs that most strongly bind antigen produce a greater number of progeny that survive than do those with weaker binding affinities. B cells thus stimulated undergo an additional, remarkable transformation, referred to as “affinity maturation.” Affinity maturation describes the process by which some progeny of activated cells produce antibodies (via point mutations on DNA coding for antigen-binding sites) with an even higher affinity for the provoking antigen than that present on the parent cell. Thus, as the immune response pro-

gresses, the system learns how to better recognize and thereby clear invading proteins. The final portion of the response is denoted by the formation of memory cells. These are both B cells and T cells that remain in the system for months or years and are capable of responding to previously recognized antigens in a much shorter time and with greater vigor than are newly activated cells. The set of all B cells and their attendant immunoglobulins is referred to as a repertoire.

A few more words concerning antigens are in order. The antigens that elicit the greatest immune response are made of proteins. Proteins are large molecules that are made of smaller units referred to as amino acids. Proteins exhibit three types of organization, referred to as primary, secondary, and tertiary structures. Only primary and tertiary structures are germane to the current discussion. The actual linear, physical sequence of amino acids that make up the protein constitutes the primary structure. The three-dimensional structure of the protein with all its folds and twists is what is known as the tertiary structure. Think of a strand of pearls. Each pearl is an amino acid. The primary structure is simply the linear order of the pearls on the strand. Now twist the strand around your finger. The coil that is formed is the strand's tertiary structure (of course, natural proteins have far more complex tertiary structures). B cells recognize the three-dimensional aspects of the antigen, whereas T cells learn the primary structure or amino acid sequence. The entire process is an example of an extraordinary adaptive process.

### Abstracting Information-processing Ideas from the Immune System

This brief discussion of the immune system recounts a number of remarkable features that would seem to be useful in information processing. Currently available artificial immune systems tend to focus on a particular subset of the features found in natural immune systems, many of which are not incorporated into the design of Immunos-81. This feature subset will be reviewed before the design of Immunos-81 is presented.

Perhaps the most fundamental feature of the immune system is its ability to distinguish self from nonself. This ability is possible because of two very important factors—the fact that self does not change over time, and the two-step selection process to which maturing T cells are subjected. Applying self-recognition as an information concept requires finding a suitable problem in the real world. Here the value of the “unchanging” aspect of self becomes obvious. Thus, one possible application of self-recognition is detection of perturbations or changes in a stable system. This fea-

ture has been studied at length by Forrest et al.<sup>2</sup> using a negative-selection algorithm for change detection to provide more robust security for computer operating systems. Dasgupta and Attoh-Okine<sup>3</sup> describe a variety of projects, including work with Forrest, in which they experimented with this concept as a mechanism for anomaly detection in time series data.

B-cell activation and the subsequent creation of high-affinity clones via affinity maturation have also received a good deal of attention from artificial immune system designers.<sup>4–7</sup> Genetic algorithms, as initially described by Holland,<sup>8</sup> have done much to promote the use of this feature of the immune system. In artificial immune systems that make use of the concept of affinity maturation, a population of artificial antibodies/B cells is created with the amino acid sequences of antigen-binding sites represented by binary strings. Antigens are likewise represented as binary strings. A matching algorithm is then used to determine the degree of affinity between an antibody's binding site (paratope) and the binding site of an antigen (epitope). Using the principles of genetic algorithms, a population of antibodies is evolved that attains a higher affinity for the antigen with each new generation. The entire population of antibodies/B cells that results (i.e., the repertoire) now reflects the features of the presented antigen. Using public domain genetic algorithm software, Forrest et al.<sup>4</sup> have demonstrated that a population of artificial antibodies could be evolved that would recognize a diverse artificial antigen population while maintaining the stability of the antibody population.

### Idiotypic Network Theory

A final feature commonly found in current artificial immune systems (especially in combination with genetic-algorithm-influenced antibody population growth models) is the incorporation of some aspects of Jerne's idiotypic network model.<sup>9–11</sup> Jerne's model is seen as providing a viable theoretic underpinning for explaining the dynamics of interactions between the various components of artificial immune systems. Jerne postulated that the immune system is regulated by antibody–antibody and antibody–lymphocyte interactions. Idiotypic network theory stipulates that, in the absence of antigen, the immune system is in a state of dynamic equilibrium. Specifically, binding sites of immunoglobulins (paratopes) or T-cell immune receptors are capable of provoking a type of autoimmune response in which the immune system creates antibodies against them. The immunogenic sites of immunoglobulins and T-cell receptors are referred to as idiotypes. When antigen is introduced, a subset of B and T cells becomes stimulated and pro-

liferates, leading to clearing of the offending antigen. The introduction of clones with new affinities provokes a counter idiotypic response, and a cascade of anti-idiotypic, anti-anti-idiotypic, and anti-anti-anti-idiotypic interactions occur until the system once again reaches a state of equilibrium.

A good deal of work has been done to create mathematical and, later, computer-based models of idiotypic networks.<sup>10–20</sup> However, there is still no firm evidence that this is the ultimate control mechanism for natural immune systems. The most obvious effects of Jerne's theory on the design of artificial immune systems has been the emphasis on accounting for the number and types of antibodies extant in a system and the provision of a mechanism for population subsets to change in size and strength because of some type of competitive mechanism. For example, Hunt and Cooke<sup>5</sup> offer the following description of learning in their artificial immune system:

During the learning phase, input data [are] inserted into the cell network. The cells in the vicinity of the insertion point are presented with the data. An immune-based matching algorithm is used to establish the match between the data and the cell. If the match value exceeds a threshold, the cell becomes stimulated and produces other cells whose pattern matching element can mutate, which may produce better matches for the input data. These cells join the network of cells. This network reinforces the stimulation level of the better cells and represses poorer cells via a feedback mechanism. The size of the network and the links within the network are dynamically generated by the interaction of the cells.

Farmer et al.<sup>6</sup> demonstrated the similarities between classifier systems introduced by Holland and an idiotypic network immune model system that used genetic operators. Ishida and Mizessyn<sup>21</sup> applied the immune paradigm to sensor monitoring for fault diagnosis in chemical plants.

Finally, Gilbert and Routen<sup>7</sup> relate their failed attempt to build an artificial immune system designed to exhibit associative memory. They cite the inability to find a stable network configuration (i.e., one that would retain a memory of the antigen and maintain a stable antibody population) as the main reason for their lack of success.

In summary, most artificial immune system designers have relied very heavily on three principal concepts—self/nonself discrimination, generation of cell/antibody populations via genetic operators, and idiotypic network theory as the basis for controlling interactions between various artificial immune system

components. These are obviously sound architectural principles, as evidenced by the existence of working applications; however, a number of immune system features that may provide additional rich examples for building intelligent systems have been overlooked.

## An Alternative Approach to Building an Artificial Immune System

A remarkable trait of all published accounts of artificial immune system applications is their very strong adherence to biological principles gleaned from actual immunologic research as well as theoretic computer models of the immune system. Witness the failure of Gilbert and Routen to build a stable learning system. They were very careful to use equations derived from theoretic studies of the immune system<sup>6,18,19,20</sup> in building their model, which proved to be unworkable in an actual application. Building systems that adhere closely to what is known of natural immune systems with all their innate complexity, while admirable, may in the long run prove to be less than ideal from a computational standpoint. Consider the explosion in neural network activity with the introduction of the back-propagation algorithm.<sup>22</sup> Very capable applications have been built using this algorithm, yet one would be hard pressed to demonstrate such a mechanism in real nervous systems.

The recognition of the value of simple processing units connected in parallel was the most important design hint taken from the central nervous system. Imagine how little progress would have been made in building useful applications if neural net applications had to adhere strictly to all the precepts of neurophysiology. Similarly, with immune-based systems, perhaps a wider variety of applications could be developed if more systems were based on higher-level immune system features with less regard to actual biological detail. For example, natural immune systems create millions of cells daily, most of which die without encountering an antigen. This is necessary in a biological system, which cannot predict when or to what it may be called on to react. In an application such as bad loan detection, however, such redundancy would serve no useful purpose, since all input types (but not patterns) are defined in advance. In creating Immunos-81, I made a few very specific design decisions, perhaps the most important being that immune system concepts would be reduced to their most fundamental level before they are incorporated into the prototype. For example, B cells/antibodies are not randomly generated with a range of binding affinities. Instead, B-cell/antibody generation is under the control solely of entered data.

Table 1 ■

Immunos-81 Design Goals

Ability to perform classification/pattern recognition tasks:
■ Easily understood internal representation
■ Ability to generalize from input data
■ Predictable training times
■ Online learning
■ Potential to act as an associative memory
■ Acceptance of continuous and nominal variables
■ Capacity to learn and recall large numbers of patterns
■ Experience-based learning
■ Supervised learning

Table 2 ■

Immune System Concepts and Their Adaptation for Immunos-81

T-cell concepts:
T cells control the activities in the immune system.
T cells learn the primary structure of antigens (i.e., the sequence of amino acids).
T cells decide what is interesting (not based on self/nonself).
B-cell concepts:
B cells recognize surface features of antigens.
Clones of a particular B cell, because of affinity maturation, have a range of affinities for the same antigen.
IgG recognition is more specific than IgM recognition.
Learning:
When new antigens are introduced, the behavior of clone population group is adjusted only during learning (primary response).
Ability to recognize an antigen on secondary response results in no adjustment in the influence of clone groups.
Learning occurs in two steps—T-cell recognition followed by B-cell creation or clone expansion, or both.
Antigen concentration is accounted for during learning
Recognition:
The total avidity for an epitope determines the winner.
T cells screen unknowns prior to B-cell presentation.
Concepts not explicitly used:
No attempt is made to explicitly represent a concept of self.
No preformed T or B cells are present in the system.
No idiotypic interactions are stimulated.

Design Goals for Immunos-81

The immune system was chosen as a design paradigm because at its most fundamental level it may be viewed as a learning system that readily accepts new input patterns of arbitrary length, maintains a database of previously encountered patterns, and on reintroduction recognizes learned patterns efficaciously. These are the characteristics of the immune system that Immunos-81 attempts to emulate. A list of the specific design goals of Immunos-81 may be found in

Table 1. Immunos-81, while not intended as a strict implementation of a natural immune system, does make use of many information processing techniques used by natural systems. Table 2 lists concepts that were borrowed to create Immunos-81 components.

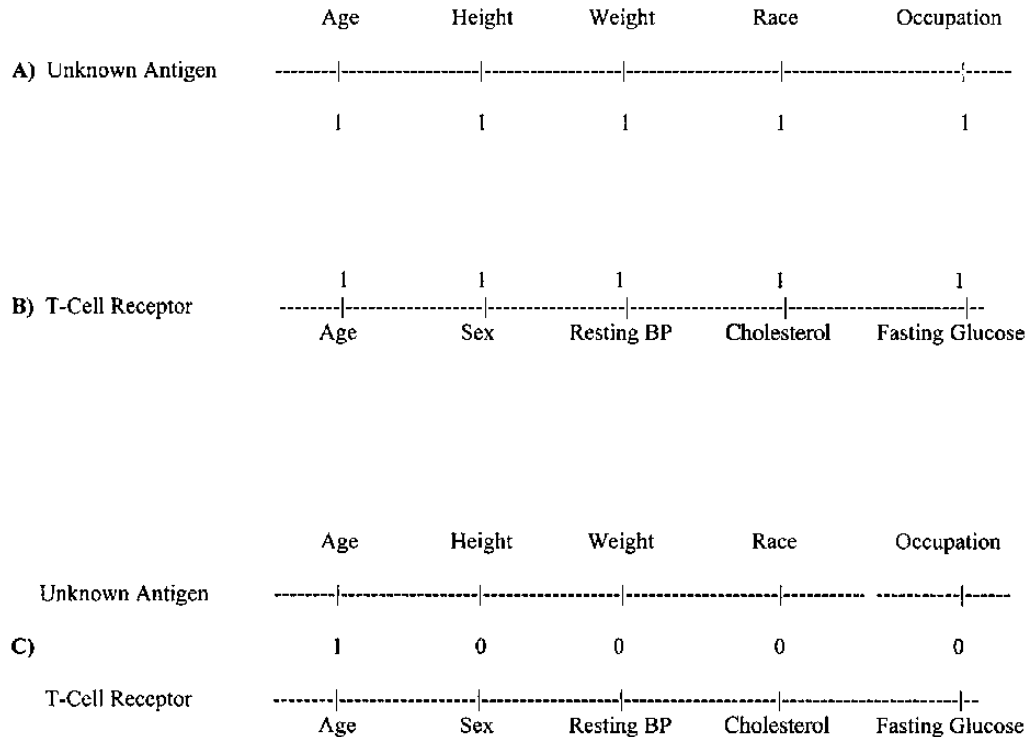
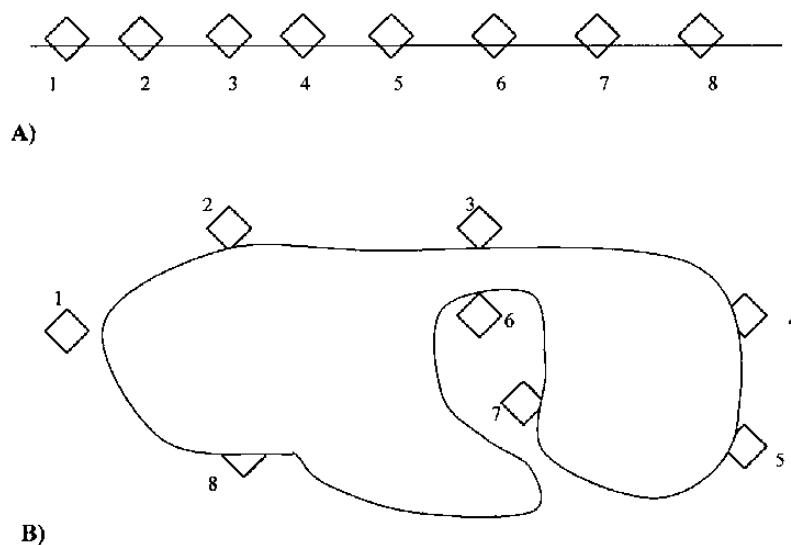
Artificial T Cells

T cells are the central control point for Immunos-81. Antigens are presented to T Cells for both learning and recognition. T cells are concerned mainly with the primary structure/sequence of antigens (Figure 1A). For the current coronary artery disease data set, the physical sequence of the fields in each record constitute the primary sequence or structure of the antigen. Consequently, record fields are the Immunos-81 equivalent of amino acids in natural antigens. This sequence is recorded by T cells. In the simplest form of T-cell learning or recognition, an antigen might be recognized by matching its primary sequence to that of one stored in a T cell. For example, a T cell that recognizes a sequence of “age, weight, height, race” would have a low binding affinity for a sequence of “age, cholesterol, resting blood pressure, fasting blood glucose.” More complex recognition might be based on patterns embedded in sequences (e.g., temporal, repeating groups). In Immunos-81, antigens may be of arbitrary length and variables may consist of a variety of data types.

If an antigen class has been previously encountered, the T cell presented with an unknown antigen of that class presents it in turn to a specific subpopulation of B cells (clones) for recognition. If the antigen is a member of a class not previously encountered, an attempt is made to learn its primary structure (amino acid sequence), and then a B-cell population is created that can recognize special features of the antigen (tertiary structure) (Figure 1B). T-cell receptors are represented as arrays with binary paratopes. A direct result of this design is that T cells in Immunos-81 perform a type of “class-level” recognition.

During recognition, unknowns are initially presented to T cells, which compete to find one that perfectly matches the primary sequence of the unknown—that is, all variables in the unknown match those in the T cell in both type and sequence (Figure 2). If a perfect match is found, then the B-cell population controlled by the winning T cell is presented with the unknown to determine which instance of the class the unknown represents. Although not implemented completely in the current version of Immunos-81, partial matching of unknowns may be used to determine degrees of membership in a particular class. The ability to determine the degree of membership in a class permits the

**Figure 1** (A), Generic antigen with eight potential binding sites—primary structure (T-cell recognition). (B), Generic natural antigen with eight potential binding sites—tertiary structure (B-cell recognition). B cells recognize instances of a class by looking for special features. In natural antigens, these may be thought of as surface features. In the figure, sites 6 and 7 are folded into the body of the antigen, making them relatively inaccessible to B cells/antibodies. Therefore, B cells/antibodies directed at this antigen would tend to notice sites 1 through 5 and site 8. In Immunos-81, special features are denoted by paratopic affinities. In clone populations (groups of B cells), the greater the affinity value of a paratopic site, the more likely that site represents a special feature of the antigen. For example, if “male gender” has an affinity of 0.34 and “female gender” an affinity of 0.51, then being female is a stronger indicator than being male of membership in the antigen class recognized by this clone.



**Figure 2** T-cell Paratope and unknown. T cells and antigen binding sites are binary. (A) and (B), Antigens and T cells show a value of 1 at a binding site if the feature is present in the represented class, and a 0 if the feature is absent. Class membership may be determined by matching the T-cell receptor sites to those of the unknown. (C), In the present case a match occurs only at the “Age” paratope/epitope pair (one of five sites). Thus, the antigen is unlikely to be a member of the class recognized by this T cell. During recognition, feature sequence is very important. No attempt is made to reorder features to improve match results. Thus, the feature order and the presence or absence of a feature affects recognition. Partial matches may be used to assign degrees of membership to a class, permitting a form of generalization. Partial matching is not implemented in the current version of Immunos-81.

Table 3 ■

## Immunos-81 Components

**Antigen:** A data grouping from the real world that may consist of multiple variables of any type.

**T cell:** A control agent that represents a particular class and determines the sequence and types of variable within an antigen; whether an antigen has been previously encountered; and which B-cell clone will decide the identity of an antigen.

**B cell:** An entity that represents an instance of a particular class during learning. After learning has occurred, no explicit representations of B cells are stored by Immunos-81.

**Amino Acid Library:** All variables encountered by Immunos-81, which are each assigned a name and type. This information is contained in an ordered pick-list and may be used to construct new antigen sequences. This ensures a consistent definition for all variables. Each entry has an Amino Acid Number, Amino Acid Name, and Data Type. Typical entries for Immunos-81 are: 1, Age, numeric; 2, Sex, nominal, etc.

**Clone:** A recognition agent—mathematical representation of a population of B-Cells

system to perform a type of generalization (i.e., if no perfect match is found, then the T cell with the highest match total wins and presents the unknown to its B cells). To ensure that sequences are always compared properly, a library of epitopes (i.e., variables) (Table 3) is maintained and is referred to as the amino acid library (AALib). Each entry in the library consists of a unique numerical identifier (Amino Acid Number), unique name (Amino Acid Name), and data type (nominal, numeric, ordinal). When new antigens are presented to the system, their record fields (i.e., amino acids or epitopes) are decoded and placed in the AALib. T cells use the information in this library to construct their receptors and those of B cells. During recognition, AALib data are used to determine whether T-cell paratopes match the epitopes of the unknown. Once the primary structure has been decoded, a B cell is created that matches the antigenic sites (epitopes) on the antigen.

## Artificial B Cells

Antibodies, which in natural immune systems are B-cell products and serve as B-cell surface receptors, do not exist as discrete entities in Immunos-81. Since B-cell surface receptors are identical to the antibodies produced by the B cell, each B cell represents all its potential antibody progeny. The notation Ab is used to denote the antibody/B-cell surface receptor duality. B cells are created that represent an “immune response” to instances of a particular class found in a data set (antigens).

For example, the data used to test the recognition capabilities of Immunos-81 consist of cases of patients evaluated for coronary artery disease. Two categories of patients are represented in this group (referred to here as a “class”)—those with coronary disease (CAD+) and those without (CAD−). An antigen would consist of a complete data set for each patient, containing one value for each possible trait (age, sex, systolic blood pressure, cholesterol level, etc.). T cells, which represent and monitor the “immune response” at the class level, are created for each distinct class. CAD+ and CAD−, being two instances of the same class, would lead to the creation of a single CAD T cell. B cells perform “instance-level” recognition and, in this example, one B cell would be created for each CAD+ and CAD− antigen set. In a sufficiently large data set, it could reasonably be expected that more than one record would exist that represents an instance of the CAD+ or CAD− state. The entire population of CAD+ B cells would then be grouped to form a “clone” (as would the CAD− B cells), capturing the features of CAD+ antigens. The clone is the basic unit of recognition in Immunos-81.

The relationship between B cells and clones is as follows (Figure 3): If a B-cell (Ab) receptor’s paratopic site is represented by  $P$ , then B-cell “X” ( $Ab_x$ ), with ten potential binding sites, would be denoted  $Ab_x(P_{10})$  and the  $i$ -th paratope  $P_i$ . The “affinity” (binding strength) of a paratopic site for a particular epitope is denoted  $P_i.a$ . Thus, the affinity of the eighth paratopic site on B cell “X” for an antigenic epitope would be denoted ( $Ab_x P_{8a}$ ). In Immunos-81, at the antibody/B-cell level, paratopic affinity is binary (i.e., if the feature is present in the antigen, then the corresponding site on the antibody/B cell is stored as 1; if absent, as 0).

During learning in Immunos-81, presentation of an unknown results in the creation of a T cell, which then directs the creation of B cells whose receptors recognize the epitopes of the unknown. In natural immune systems the magnitude of the immune response (and size of the resulting clone) is a function of the antigenicity of the foreign material and the concentration of that material in the serum. This concept is approximated in Immunos-81 by the use of “antigenic units” (au). An antigenic unit is the basic unit of “foreignness” and is defined as the amount of antigen required to cause the creation of a single B cell. The concentration of a given antigen,  $q$ , is represented in Immunos-81 as  $[q]^{au}$ .

Most natural antigens are complex proteins that are processed into smaller units by APCs. Therefore, any given antigen may have multiple epitopes, and the B-cell response will constitute a response to the entire

**Figure 3** Clone population. *A*, Each clone consists of a series of B cells that recognize the same antigen class instance. *B*, Once an antigen class instance is learned, a clone that recognizes the future instances of that class is created. Notice that, once a clone is formed, individual B cells no longer exist, only their pooled representation in the form of the clone. Also, notice that, at the level of the clone, affinities ( $Cl_x P_{na}$ ) are continuous variables.

**A)**  $Ab_1(P_1, P_2, P_3, P_4, P_5, \dots, P_n)$   
 $Ab_2(P_1, P_2, P_3, P_4, P_5, \dots, P_n)$   
 $Ab_3(P_1, P_2, P_3, P_4, P_5, \dots, P_n)$   
 $Ab_4(P_1, P_2, P_3, P_4, P_5, \dots, P_n)$   
 $\dots$   
 $\dots$   
 $Ab_n(P_1, P_2, P_3, P_4, P_5, \dots, P_n)$

**B)**  $Cl_x((0.005, P_{1a}), (0.1, P_{2a}), (0.023, P_{3a}), (0.034, P_{4a}), (0.001, P_{5a}), \dots, (Cl_a, P_{na}))$

epitope population. All the subparticles (and their epitopes) that are created as a result of APC processing belong to the same parent antigen (i.e., class), and each is capable of invoking an immune response. Immunos-81 assumes (during supervised learning) that all antigens presented as a group belong to a same class or class instance. And each instance of the class is equal to one antigenic unit. If 350 instances (cases) are presented for learning purposes, then  $[q]^{350}$  represents the concentration of antigen  $q$ .

### Clone Characteristics

Immunos-81 creates one B cell for each antigen presented, resulting in the creation of a clone (Cl) population. As in natural immune systems, antigens of the same class will cause the generation of an antibody clone population with varying affinities for a given epitopic site. This principle was used in designing the recognition engine of Immunos-81. Recalling the coronary artery disease example, assume that the problem is learning to classify patients into two groups—those with coronary artery disease (CAD+) and those without (CAD−). If 100 cases with 13 variables per case are presented to the system, then one T cell will be created (there is one class with two instances) and a clone population of antibodies/B cells will be created, one for each class instance. When an unknown case is offered for recognition, the clones representing each instance (CAD+ and CAD−) will compete. Recognition is decided by clonal avidity, not at the level of individual antibodies/B cells. In other words, the clone population that has the optimum mixture of high receptor affinity and high concentration will be declared the winner.

Clone-level affinity is a function of the affinities of the antibodies/B cells it contains. Clone-level affinity of clone “X” ( $Cl_x$ ) at paratope site 8 may be represented as:

$$Cl_x(P_{8a}) = k \sum_{j=1}^{j=1} (Ab_j P_{8a})$$

(clone-level affinity)

where  $j$  indexes all antibodies/B cells in the clone,  $N$  is the number of Ab in the clone with value 1 at  $P_{8a}$ , and  $k$  is the scaling factor.

The size of a clone ( $S$ ), as stated earlier, is proportional to the concentration of the antigen that provoked its creation:

$$S(Cl_x) = k[q]^v \quad (\text{clonal size})$$

The avidity ( $A$ ) of a clone is the combined affinities for all paratopic sites. Given this definition, it is obvious that the avidity of a clone for a particular antigen will vary according to the epitopes present for binding on the antigen (i.e., the number of features present). Also, the greater the number of antibodies/B cells that the clone comprises, the more opportunities it has for antigen binding. This very basic principle, which plays a key role in natural immune systems, is emulated in Immunos-81 by inclusion of a clone-level scaling factor ( $k_2$ ), which ensures that the relative avidity of a clone is adjusted in accordance with the number of clones present for a particular antigenic class. The total avidity for paratopic site 8 would denoted as:

$$Cl_x A(P_8) = k_2 [Cl_x(P_{8a}) \cdot S(Cl_x)]$$

(total clonal affinity at paratope 8)

where  $k_2$  is the scaling factor that adjusts for the total number of clones.

The total avidity for clone X would then become:

$$Cl_x A(P) = k_2 \left[ \sum_{i=1}^{i=1} (Cl_x(P_{ia})) \right] \cdot S(Cl_x)$$

(total clonal avidity)

where  $i$  indexes clonal paratopes in the clone,  $N$  is the number of paratopes in the clone with value 1 at  $P_{ia}$ , and  $k_2$  is the scaling factor that adjusts for the total number of clones.



65	M	Angina	160	225	F	Norma	135	True	2	flat	3	reverse
AGE	SEX	CHEST_PAIN	RESTING_BP	CHOL	GLU	REST_MAX_HRT_RA_EKG	EXERCISE	ST_DEPRESS	ST_SLOPE	MAJOR_VESS	THALLIUM	

**Figure 4** Coronary artery disease antigen/input record. Antigens are presented to Immunos-81 as a list of ordered fields (they may be database records, or arrays). Fields may be of numeric, nominal, or ordinal type.

Once a clone population has been created, the antigen is considered learned. A consequence of this definition of clonal avidity is that the influence of any particular clone is subject to change as new antigen classes are added or as additional instances of a particular class are presented to the system.

### Special Learning Features

An interesting feature of Immunos-81, which is currently being investigated, is its potential ability to learn "on line." In this setting online learning refers to the ability to add new classes or class instances to the artificial immune system without having to rebuild the entire system. For example, the current artificial immune system contains one T-cell (CAD) and two antibody/B-cell clones, (CAD+) and (CAD-). The presentation of a data set containing information on valvular heart disease would result in the creation of a T cell for that class and series of antibody/B-cell clones representing class instances (e.g., mitral stenosis, aortic regurgitation). The new T cell and clones would become active parts of the artificial immune system, leaving the CAD components undisturbed. Consider the effects of making similar adjustments to a back-propagation neural network: New output states would require a complete retraining of the network and possibly the addition of more hidden layer neurons. In addition to the ability to accept new classes without disruption of current classes, new instances of current classes may be added, with retraining taking place only in the affected classes. Thus, clone size and total affinity (and therefore avidity) may change at any time.

The dynamic nature of clonal avidity permits Immunos-81 to mimic forgetting, which is handled as follows. Using our coronary artery disease example, in which 14 variables are provided to the system and in which one variable is age, the effect of adding cases of CAD+ Medicare patients to the database would be to increase the affinity for older ages in the CAD+ clone subpopulation, while that in the CAD-subpopulation would decrease. This ability to adjust clonal avidities on the basis of experience improves the ability of Immunos-81 to adjust to its local environment.

The final recognition step involves the determination of the avidity difference between the two clones that demonstrate the highest avidities. Two selectable strategies are available—simple highest avidity (SHA), in which the winning clone has the highest total avidity; and relative highest avidity (RHA), in which the winning clone has the highest total avidity and that avidity is also at least 5 percent greater than the avidity of the closest competitor.

### Methods

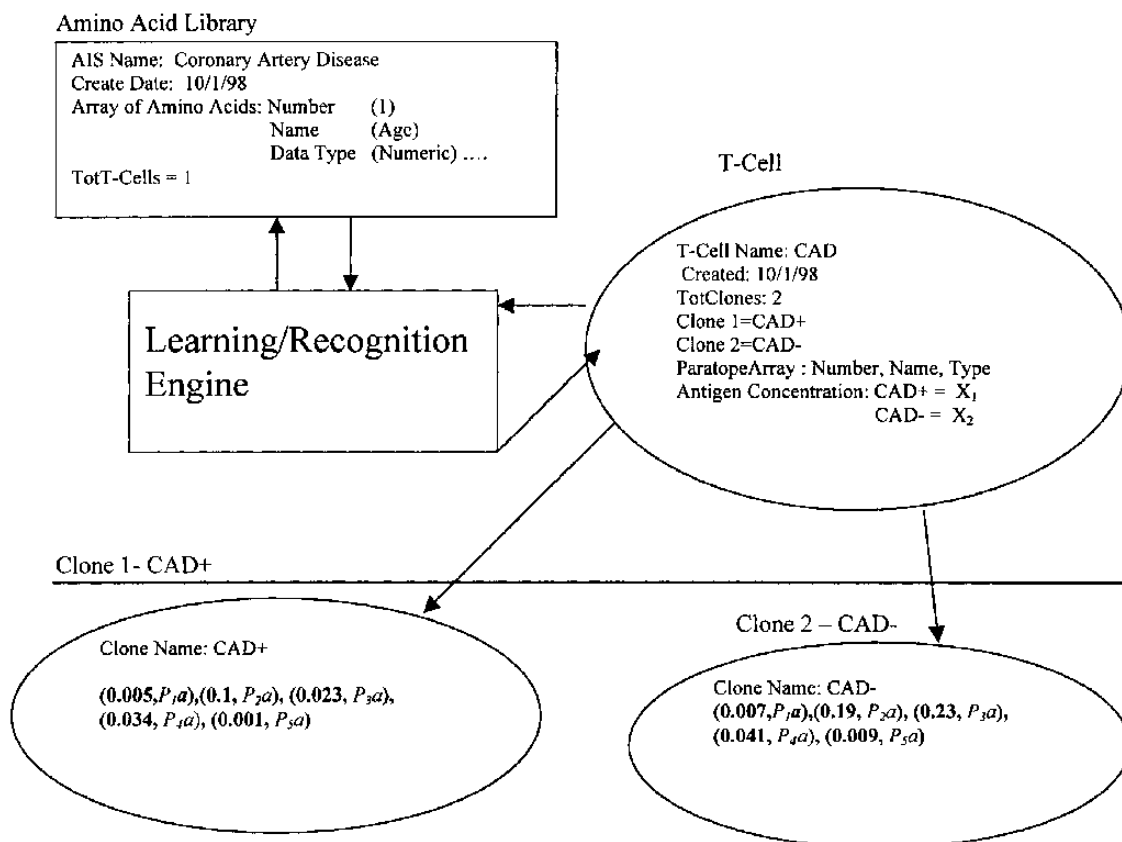
Testing of Immunos-81 was carried out using two standard databases available from the Machine Learning Group of the University of California—Irvine.<sup>23</sup> The Cleveland database was used for training and cross-validation, and the Long Beach database for test unknowns. Both databases were provided by Detrano et al.<sup>24</sup> The Cleveland database consists of 303 cases of 14 variables (eight nominal, six continuous) each, derived from patients referred for coronary angiography at the Cleveland Clinic from 1981 until 1984. Patients with a history of myocardial infarction, valvular disease, or cardiomyopathy were excluded. Each case includes the following variables—age, sex, chest pain type, resting systolic blood pressure,

```

Read Primary Sequence (record columns)
  Determine all variable names and data types
  IF Amino Acid Library exists
    Add new Names and Data Types
  ELSE
    Create Amino Acid Library
    Add new Names and Data Types
  ENDIF
Create T-Cell (s) for each Antigen Class
Determine Number of Class. Instances to be learned
Identify all antigen records for each class instance
  Read values (epitopes) for each antigen
  For each epitopic site in each record, determine value
    For Numeric values
      calculate
      Mean, Hi, Low
    For Nominal Values
      Determine frequency
Calculate affinity at each paratopic site
Create Clone with assigned affinity for each paratopic site
Perform scaling to set final avidity values for paratopes of each clone

```

**Figure 5** Learning algorithm.



**Figure 6** Components of the coronary artery disease Immunos-81 artificial immune system (AIS). The amino acid library and T cells are the highest-level components—all information entering the system must be registered with both. The clones are created using information stored in T cells. The clones represented above have nonbinary paratopic affinities. These values will vary depending on antigen concentration, the number of epitopes per antigen, and the number of clones present for instances of the same class in the AIS.

serum cholesterol, fasting blood sugar (T = hyperglycemic), exercise electrocardiographic results, exercise thallium results, fluoroscopy results (number of vessels noted), the presence of exercise-induced angina, S-T slope shape during peak exercise, S-T segment depression (in millimeters), maximal heart rate, and postangiography diagnosis. Patients were divided into five groups on the basis of angiography results—no ischemic heart disease and ischemic disease of S1 through S4. For the testing of Immunos-81, patients were placed in only two groups—CAD+ and CAD-. The case set was then divided into ten groups and used to conduct a ten-way cross-validation study. Each input record consisted of 13 input variables, since postangiography diagnosis was excluded.

### Input (Learning)

Each antigen processed by Immunos-81 is received in the form of an ordered list (database record or array) (Figure 4). The order of the fields in the record con-

stitutes the primary sequence or structure of the antigen. The names and data type of each field are read from the database and used to construct the receptor of the T cell for this antigen class. All records that represent a particular instance of class are then presented to the T cell, which then guides the creation of a clone population of B cells. In the present case, clone populations representing CAD+ and CAD- states are created. Once clones are created, clonal avidities are assigned on the basis of the antigen concentration that caused their formation and the number of competing clones (Figure 5). When learning is completed, the active components of the Immunos-81 artificial immune system are the amino acid library, T cells, and clones (Figure 6).

### Output (Recognition)

During recognition, one or more unknown antigens are presented to Immunos-81. Each record undergoes a two-level match prior to processing. The first-level match is performed by the T cells in the system (for

```

Present Antigen Record to T-Cell
  IF Perfect Sequence Match
    Present to Clone Population
      Calculate Avidity for each Clone controlled by T-Cell
    Sort Avidity Values
    Compare top two values
      Declare Clone winner
        (SHA) IF it has highest total avidity
        OR
        (RHA) IF it has highest total avidity
          AND
          its avidity is at least 5% greater than
          that of the nearest competitor
  END

```

**Figure 7** Recognition algorithm for Immunos-81, version 1.0.

the current system, for coronary artery disease, only one T cell exists). If a perfect match occurs (that is, all paratopic sites of the T cell match those of the antigen by name and type), then all clones controlled by that T cell compete for the antigen. At the level of the clone, each epitopic site of the antigen is compared with its clonal equivalent (paratopic site), and the affinities for all sites that match are summed. The clone that generates the highest avidity (the sum of affinities for all sites) is declared the winner (Figure 7). Avidities are represented as real numbers whose values may range from 0.0 to 1.0. Table 4 is an example of the output obtained from the system after presentation of test data.

## Data Analysis

For each validation run, nine groups were selected as the training group. The training group was then divided into CAD+ and CAD- groups and presented to Immunos-81. (Angiography data were removed during cross-validation and training runs.) Next, the remaining group was provided as an unknown set, and each case was assigned to either a CAD+ or a CAD- diagnostic group on the basis of total avidity. For each validation run, decisions were provided at both 0 percent (SHA) and 5 percent (RHA) levels of difference between the avidities of competing clones. When the RHA metric was used, unknowns whose total avidity for CAD+ and CAD-clones resulted in a difference of less than 5 percent between the two calculated avidities were labeled as "too close to determine" by Immunos-81. The RHA metric provides a mechanism for allowing Immunos-81 to offer some idea of its level of certainty when classifying unknowns. The results of the validation runs are found in Table 5.

A second test of Immunos-81 recognition capabilities was conducted using the Long Beach Veterans Administration (VA) database, which contains 200 cases. This database was selected because it offered a reasonable challenge, due to the number of missing values for each case. On the average, each case has four missing values. The entire VA database was treated as an unknown, and each case was presented to Immunos-81, which had been trained with the Cleveland data set.

## Results

The cross-validation yielded very interesting results. Correct recognition using SHA ranged from a high of 96 percent (set 4) to a low of 63.2 percent (set 8). Overall, the greatest number of incorrect classifications occurred with attempts to separate CAD- level subjects from those in the S1 category (the least ill of the CAD+ group). This is as expected, since these two classes have the least amount of difference (clinically) between them. The average correct classification for all runs was 83.2 percent using SHA. Using the RHA metric (5 percent difference), the system labeled 11.2 percent (34 cases) "too close to determine" and no further attempt was made to classify them. Of the remaining 269 cases, 85.5 percent (230 cases) were correctly classified.

When presented with the Long Beach VA data set (Table 6), Immunos-81 was able to identify the correct diagnosis in 73.5 percent of the cases using SHA. When tested using RHA, the system deemed 16 percent of cases "too close to determine." The remaining 168 unknowns were classified with an 80.3 percent

Table 4 ■

Sample Output from the Immunos-81 Recognition Engine

Antigen Number	Avidity	Clone	Winner
1	0.606277	CAD-	W
	0.552119	CAD+	-
2	0.385981	CAD-	-
	0.597804	CAD+	W
3	0.717996	CAD-	W
	0.468982	CAD+	-

NOTE: The antigen offered for recognition was test DataSet9. The sample output is a subset taken from the cross-validation run using test set 9 and is presented here as a table listing the number of the antigen record in the parent antigen set; the total avidity of the antigen for each clone to which it was exposed; and the winner (the clone with the highest avidity), designated by the letter W.

Table 5 ■

## Ten-way Cross-validation Results (Cleveland Data Set)

Set Number	0% Difference (SHA)		5% Difference (RHA)		
	Correct	Incorrect	Correct	Incorrect	Not Attempted
1	27/33	6/33	25/28	3/28	5/33
2	23/30	7/30	23/28	5/28	2/30
3	26/30	4/30	25/29	4/29	1/30
4	29/30	1/30	27/28	1/28	2/30
5	22/30	8/30	20/26	6/26	4/30
6	26/30	4/30	23/27	4/27	3/30
7	27/30	3/30	25/27	2/27	3/30
8	19/30	11/30	18/28	10/28	2/30
9	27/30	3/30	24/26	2/26	4/30
10	26/30	4/30	20/22	2/22	8/30
TOTALS	252/303 (83.2%)	51/303 (16.8%)	230/269 (85.5%)	39/269 (14.5%)	34/303 (11.22%)

Table 6 ■

## Long Beach Veterans Administration Data Results

0% Difference (SHA)		5% Difference (RHA)		
Correct	Incorrect	Correct	Incorrect	Not Attempted
147/200	53/200	135/168	33/168	32/200

success rate. Roughly two thirds of the cases placed in the “too close to determine” group had been incorrectly classified when SHA was used as the recognition threshold.

## Discussion

Immunos-81 performed very well when classifying the Cleveland Data set, compared with other systems. The closest competitor, as described by Wettschereck et al.,<sup>25</sup> is a *k*-nearest-neighbor classifier utilizing a feature-weighting scheme (82.4 percent). A cluster algorithm created by Gennari et al.<sup>26</sup> managed 78.9 percent correct. Aha<sup>27</sup> attained 75.4 percent accuracy with the machine learning algorithm C4.0 and 77.9 percent accuracy with an instance-based learning algorithm (IB3). Gallagher et al.,<sup>28</sup> using a constrained discriminant analysis model, attempted a ten-way validation and test-set recognition in which all groups (healthy subjects and patients with S1 through S4 disease labels) served as the test standard (not simply CAD+ or CAD−). They achieved 70.3 percent accuracy during the validation exercise and only 42.5 percent during actual testing.

Simplicity, a major design goal of Immunos-81, is a feature that sets it apart from techniques like cluster analysis and from machine learning algorithms like C4.0. The internal representations of Immunos-81 are

easily understood, and its outputs are each presented as a numeric avidity value (0.0 to 1.0) and a nominal class assignment. Another significant feature is speed. Immunos-81 is a single-pass learning and recognition system. The average time required to learn the CAD+ and CAD− sets and to create clones was less than one minute. Immunos-81 is easy to use. It does not require any special knowledge or expertise.

The availability of a user-selectable threshold may prove to be a valuable feature. In both the Cleveland and Long Beach VA data sets, the 5-percent threshold resulted in a higher level of correct classifications compared with the total attempted. In the Cleveland data set, however, most of the cases placed in the “too close to determine” group had been correctly classified initially. In the VA data set, most cases had previously been incorrectly classified. The performance of Immunos-81 in classifying the VA data set seems to compare well with the discriminant function (DF) designed by Detrano et al.,<sup>24</sup> which also used the Cleveland database as a training set. Detrano’s group performed its analysis using two groups identified by probability thresholds assigned by the researchers. One group consisted of those subjects whose probability for having CAD was either less than 0.2 or greater than 0.8; the other contained all subjects, regardless of prior probability. The DF performed best at the 0.5 prior probability threshold when all patients were included (79 percent). However, performance decreased to 66 percent when “less than 0.2” or “greater than 0.8” group was excluded from the analysis. Overall, Immunos-81 compares very well with the performance of the DF when one considers that no attempt to determine prior probability of disease was required and human input into the lengthy process of finding a good discriminant function (as described by Detrano et al.) was not necessary.

The uneven performance of SHA compared with RHA in the VA and Cleveland data sets is a puzzle. It was expected that use of RHA would tend to reduce the number of incorrect classifications produced using SHA, by classifying borderline cases as "too close to determine." This proved to be true with the VA data set. However, when it was applied to the Cleveland data set, the number of correct classifications was reduced, an unexpected result. Perhaps the quality of data in each set played a role in the outcomes. The Cleveland data set is, for the most part, complete (only four of 4,242 values were missing). The VA data set, on the other hand, had an average of four missing values per case. When complete data sets are available SHA may be best, because it relies on the data to reflect the true (and absolute) differences between antigens during recognition. If this is true, RHA might best be considered an "educated guess" that is acceptable when dealing with noisy data. At this stage in the development of Immunos-81, it is not possible to determine the ultimate utility of this feature. Even with this variance in behavior, however, it appears that the 5-percent threshold result does offer real value that does not often appear in many classification systems. First, it points out borderline cases, which improves the reliability of its output. Second, the predictive value of its classifications are higher because of the removal of borderline cases. Finally, it may eventually provide the basis of recognizing instances of a previously unknown class in an unsupervised learning scenario. However, unsupervised learning is not a focus in this version of Immunos-81.

When compared with back-propagation artificial neural networks, Immunos-81 appears to offer a few definite advantages, the most significant of which may be the transparency of the internal state of Immunos-81.<sup>29</sup> In addition, Immunos-81 has a finite training time and permits the user to set the "level of certainty" (using SHA and RHA) required to make a classification.

The immune system model offered by Immunos-81 may prove to be useful for data that have important temporal characteristics. Since T cells in Immunos-81 are capable of monitoring specific sequence information, it may be possible to use the system to discern patterns that occur in temporal dimensions along with more static data. An example of such an application might be the analysis of patient outcomes when the temporal sequence of a predetermined set of interventions is altered.

One design goal that was not realized for Immunos-81 was the creation of an associative memory. No such property is evident in the current version. However,

this may be because of the data sets chosen for testing and the limited range of interactions currently supported among system elements. Many potential features of the system remain unexplored (e.g., temporal capabilities of T cells, use of avidity thresholds as a means of allowing unsupervised learning, recognition properties of systems with multiple T cells, and an extensive amino acid library).

## Conclusion

In summary, Immunos-81 is an example of immune computation in the form of a supervised learning system. It has performed well in comparison with other machine learning algorithms when tested using standard data sets. It is potentially embeddable, and versions capable of learning on line in real time are possible with the current architecture. Immunos-81 represents a new application of immune system concepts for machine learning. Further testing is required to prove its ultimate utility.

## References ■

1. Benjamini E, Sunshine G, Leskowitz S. Immunology: A Short Course. 3rd ed. New York: Wiley-Liss, 1996.
2. Forrest S, Hofmeyr SA, Somayaji A. Computer immunology. *Commun Assoc Comput Machinery*. 1997;10:88–96.
3. Dasgupta D, Attouh-Okine N. Immunity-based systems: a survey. *Proceedings of the 1997 IEEE Conference on Systems, Man, and Cybernetics*. 1997:369.
4. Forrest S, Perelson AS, Allen L, Cherukuri R. Using genetic algorithms to explore pattern recognition in the immune system. *Evol Comput*. 1993;3:191–211.
5. Hunt JE, Cooke DE. A learning using an artificial immune system. *J Network Comput Appl*. 1996;122:33–67.
6. Farmer JD, Packard NH, Perelson AS. The immune system, adaptation, and machine learning. *Physica D*. 1986;22:187–204.
7. Gilbert CJ, Routen TW. Associative memory in an immune-based system. *Proceedings of the 12th National Conference on Artificial Intelligence*. Menlo Park, Calif: AAAI Press, 1994:852.
8. Holland JH. *Adaptation in natural and artificial systems*. Ann Arbor, Mich: The University of Michigan Press, 1975.
9. Jerne NK. Idiotypic networks and other preconceived ideas. *Immun Rev*. 1984;79:5–24,9.
10. Anderson RW, Neumann AU, Perelson AS. A cayley tree immune network model with antibody dynamics. *Bull Math Biol*. 1993;6:1091–131.
11. Detours V, Sulzer B, Perelson AS. Size and connectivity of the idiotypic network are independent of the discreteness and size of affinity distribution. *J Theor Biol*. 1996;183:409–16.
12. Carneiro J, Coutinho A, Faro J, Stewart J. A model of the immune network with B–T cell co-operation, part I: prototypical structures and dynamics. *J Theor Biol*. 1996;182:513–29.
13. DeBoer RJ, Segel LE, Perelson AS. Pattern formation in one and two dimensional shape-space models of the immune system. *J Theor Biol*. 1992;155:295–333.
14. Fishman MA, Perelson AS. Modeling T cell–antigen pre-

- senting cell interactions. *J Theor Biol.* 1993;160:311–42.
15. Faro J, Santiago V. Studies on a recent class of network models of the immune system. *J Theor Biol.* 1993;164:271–90.
  16. Vertosick FT, Kelly RH. The immune system as a neural network: a multi-epitope approach. *J Theor Biol.* 1991;150:225–37.
  17. DeMonvel JHB, Martin OC. Memory capacity in idiotypic networks. *Bull Math Biol.* 1995;57:109–36.
  18. DeBoer RJ, Perelson AS. Size and connectivity as emergent properties of a developing immune network. *J Theor Biol.* 1991;149:381–424.
  19. Celada F, Seiden PE. A computer model of cellular interactions in the immune system. *Immunol Today.* 1992;13:56–62.
  20. DeBoer RJ, Hogeweg P. Memory but no suppression in low-dimensional symmetric idiotypic networks. *Bull Math Biol.* 1989;51:223–46.
  21. Ishida Y, Mizessyn F. Learning algorithms on an immune network model: application to sensor diagnosis. *Proceedings of the International Joint Conference on Neural Networks.* New York: IEEE Press, 1992:33–8.
  22. Rumelhart DE, McClelland JL. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol 1. Cambridge, Mass: MIT Press, 1986.
  23. Aha DW. UCI repository of machine learning databases. 1994. Available at Department of Computer Science, University of California-Irvine Web site: [www.ics.uci.edu/AI/ML/Machine-Learning.html](http://www.ics.uci.edu/AI/ML/Machine-Learning.html).
  24. Detrano R, Janosi A, Steinbrunn W, et al. International application of a new probability algorithm for the diagnosis of coronary disease. *Am J Cardiol.* 1989;64:304–10.
  25. Wettschereck D, Aha DW, Mohri T. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artif Intell Rev.* 1997;11:273–314.
  26. Gennari JH, Langley P, Fisher D. Models of information concept formation. *Artif Intell.* 1989;40:11–61.
  27. Aha DW. Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *Int J Man-Machine Stud.* 1992;36:267–87.
  28. Gallagher RJ, Lee EK, Patterson DA. An optimization model for constrained discriminant analysis and numerical experiments with iris, thyroid, and heart disease data sets. *Proc AMIA Annu Fall Symp.* 1996:209–13.
  29. Russell S, Norvig P. *Learning in neural and belief networks.* In: Russell S, Norvig P (eds). *Artificial Intelligence: A Modern Approach.* Upper Saddle River, NJ: Prentice-Hall, 1995.