

Immunity-Based Systems: A Survey

Dipankar Dasgupta

Dept. of Mathematical Sciences
The University of Memphis
Memphis, TN 38119

Nii Attoh-Okine

Dept. of Civil Engineering
Florida International University
Miami, FL 33199.

ABSTRACT

Biological systems such as human beings can be regarded as sophisticated information processing systems, and can be expected to provide inspiration for various ideas to science and engineering. Biologically-motivated information processing systems can be classified into: brain-nervous systems (neural networks), genetic systems (evolutionary algorithms), and immune systems (artificial immune systems). Among these, nervous systems and genetic systems have been widely applied to various fields. There have been a relative few applications of the immune system. This paper presents a survey of artificial immune systems and provides numerous insights of immunity-based systems applications in science and engineering.

1 INTRODUCTION

The natural immune system is a very complex system with several mechanisms for defense against pathogenic organisms. The main purpose of the immune system is to recognize all cells (or molecules) within the body and categorize those cells as self or nonself. The nonself cells are further categorized in order to induce an appropriate type of defensive mechanism. The immune system learns through evolution to distinguish between dangerous foreign antigens (e.g., bacteria, viruses, etc.) and the body's own cells or molecules.

According to the Immunologists [21], [25], our body maintains a large number of immune cells-called lymphocytes, which circulate throughout the body. There are mainly two types of lymphocytes, namely T cells and B cells. These two types of lymphocytes play different roles in the immune response, though they may act together and control or affect one another's function. For example, T cells can either enhance or suppress the B cells' response to a stimulus. When an antigen invades the body, only a few of these immune cells can recognize the invader's peptides.¹ This recognition

stimulates proliferation and differentiation of the cells that produce matching clones (or antibody).² This process, called clonal expansion, generates a large population of antibody-producing cells that are specific to the antigen. The clonal expansion of immune cells result in destroying or neutralizing the antigen. It also retains some of these cells in immunological memory, so that any subsequent exposure to a similar antigen leads to rapid immune response (secondary response).

From an information-processing perspective, the immune system is a remarkable parallel and distributed adaptive system [11]. It uses learning, memory, and associative retrieval to solve recognition and classification tasks. In particular, it learns to recognize relevant patterns, remember patterns that have been seen previously, and use combinatorics to construct pattern detectors efficiently. Also, the overall behavior of the system is an emergent property of many local interactions. These remarkable information-processing abilities of the immune system provide several important aspects in the field of computation.

The purpose of this paper is to provide an overview of the rapidly emerging field called Artificial Immune Systems (also called *Immunological Computation*). In the next section, we will discuss different models based on various mechanisms of the immune system. Section 3 provides the applications of these models in the fields of science and engineering. The last section will conclude with some remarks.

2 IMMUNE SYSTEM BASED MODELS

There exist several theories [22], [24], [30] and mathematical models [26], [27] to explain immunological phenomena. There is also a growing number of computer models [2], [28], [29] to simulate various components of the immune system and the overall behavior from a biological point of view. However, the natural immune system is also a source of inspiration for developing

antigen (called peptides) on its surface, to bring the attention of B and T cells for recognition.

²Affinity maturation occurs when the mutation rate of a cell clone increases in response to a match between the clone's antibody and an antigen. Those mutant cells that bind more tightly are stimulated to divide more rapidly.

A version of this paper was presented at the ICMAS workshop on *Immunity-Based Systems*, Japan (December 1996).

¹Antigen presenting cells or macrophages present pieces of the

intelligent methodologies toward problem solving, but there is not much research in this direction. In particular, more research is needed to extract information processing mechanisms which are of direct practical use.

The few computational models developed that have been based on immune system principles, seem to have chosen a different set of principles to emulate. Among these, the following models are often used by researchers.

2.1 Immune Network Model

Jerne [21], [22] initiated the theoretical development of idiotypic networks which presents a mathematical framework of an immune system. His theory is modeled with a differential equation which simulates the dynamics of lymphocytes - the increase or decrease of the concentration of a set of lymphocyte clones and the corresponding immunoglobins. The idiotypic network hypothesis is based on the concept that lymphocytes are not isolated, but communicate with each other among different species of lymphocytes through interaction among antibodies. Accordingly, the identification of antigens is not done by a single recognizing set but rather a system level recognition of the sets connected by antigen-antibody reaction as a network.

The key postulate of his theoretical framework is that one cell makes only one antibody. The law, one cell makes one antibody, leads to several predictions which include (i) allelic exclusion, (ii) all antibody-like receptors displayed by a lymphocyte should be identical or at least have identical light chains and identical heavy variable regions, and (iii) all antibodies produced by a single cell and its progeny should have the identical idio-type.

In formulating the framework, Jerne discusses formal and functional networks. The formal network discusses repertoires, dualism, and suppression. In the discussion of functional networks, a quantitative picture of the theory is presented. According to the functional network, even the absence of antigens that do not belong to the system must display an eigen-behavior. This mainly results from paratope-idiotope interaction within the system. Jerne concluded that the immune systems bear a striking resemblance to the nervous system when viewed as functional networks.

Based on Jerne's work, Perelson [26], presented a probabilistic approach to idiotypic networks. Perelson's approach is very mathematical, discussing more about phase transition in idiotype networks. Perelson divided phase transition in idiotypic networks to pre-critical region, transition region and post-critical region.

Jerne's proposed immune network theory [22] received a lot of attention among the researchers [26]–[28] over the last two decades and many computational as-

pects of this model are derived for practical use [16], [18], [20].

2.2 Negative Selection Algorithm

Forrest et. al. [10] developed a negative-selection algorithm for change detection based on the principles of self-nonsel self discrimination [25] in the immune system. This discrimination is achieved in part by T-cells, which have receptors on their surface that can detect foreign proteins (antigens). During the generation of T cells, receptors are made by a pseudo-random genetic rearrangement process. Then they undergo a censoring process, called negative selection, in the *thymus* where T cells that react against self-proteins are destroyed, so only those that do not bind to self-proteins are allowed to leave the thymus. These matured T cells then circulate throughout the body to perform immunological functions to protect against foreign antigens. The negative-selection algorithm works on similar principles, generating detectors randomly, and eliminating the ones that detect self, so that the remaining T-cells can detect any nonself. This algorithmic approach can be summarized as follows:

- Define *self* as a collection S of strings of length l over a finite alphabet, a collection that needs to be protected or monitor. For example, S may be a program, data file (any software), or normal pattern of activity, which is segmented into equal-sized substrings.³
- Generate a set R of *detectors*, each of which fails to match any string in S . Instead of exact or perfect matching,⁴ the method uses a partial matching rule, in which two strings match if and only if they are identical at at least r contiguous positions, where r is a suitably chosen parameter (as described in [10]).
- Monitor S for changes by continually matching the detectors in R against S . If any detector ever matches, then a change is known to have occurred, because the detectors are designed to not match any of the original strings in S .

In the original description of the algorithm [10], candidate detectors are generated *randomly* and then tested (censored) to see if they match any self string. If a match is found, the candidate is rejected. This

³This is analogous to the way, proteins are broken up by the immune system into smaller subunits, called peptides, to recognize by T-cell receptors.

⁴For strings of any significant length a perfect match is highly improbable, so a partial matching rule is used which rewards more specific matches (i.e., matches on more bits) over less specific ones. This partial matching rule reflects the fact that the immune system's recognition capabilities need to be fairly specific in order to avoid confusing self molecules with foreign molecules.

process is repeated until a desired number of detectors are generated. A probabilistic analysis is used to estimate the number of detectors that are required to provide a certain level of reliability. The major limitation of the random generation approach appears to be computational difficulty of generating valid detectors, which grows exponentially with the size of self. Subsequently, a more efficient detector generation algorithm is proposed by Helman and Forrest [13] which runs in linear time with the size of self. Other methods for generating nonself detectors have also been suggested [6] which have varying degrees of computational complexities.

This algorithm relies on three important principles: (1) each copy of the detection algorithm is unique, (2) detection is probabilistic, and (3) a robust system should detect (probabilistically) any foreign activity rather than looking for specific known patterns of changes. Further studies show [6] many insights of the algorithm. The algorithm seems to have many potential applications in change-detection, some of these are discussed in the next section.

2.3 Other Models

There exist other computation models [7], [9] which emulate different immunological aspects, for example, its ability to detect common patterns in a noisy environment [9], its ability to discover and maintain coverage of diverse pattern classes, and its ability to learn effectively, even when not all antibodies are expressed and not all antigens are presented [14]. Hoffman [15] has compared the immune system and the nervous system. He has shown many similarities in the two systems, at the level of system behavior (though differ at the respective building-block level). He postulated a symmetrical neural network model that can produce desired stimulus-response behavior similar to immune response. Farmer et. al. [7], and Bersini and Varela [1] have compared the immune system with learning classifier systems. Gilbert and Routen [12] experimented with immune network model to create a content-addressable auto-associative memory, specifically for image recognition. In this application, inputs to the system are black and white pictures of 64 by 64 pixels that are analogous to antigens. This approach seems very interesting, but their implementation failed to obtain a stable solution for the problem.

3 SOME APPLICATIONS

The models based on immune system principles are finding increasing applications in the fields of science and engineering.

3.1 Computer Security

Stephanie Forrest and her group at the University of New Mexico are working on a research project with a long-term goal to build an artificial immune system for computers. This immunity-based system has much more sophisticated notions of identity and protection than those afforded by current operating systems, and it would provide a general-purpose protection system to augment current computer security systems. The security of computer systems depends on such activities as detecting unauthorized use of computer facilities, maintaining the integrity of data files, and preventing the spread of computer viruses.

The problem of protecting computer systems from harmful viruses is viewed as an instance of the more general problem of distinguishing *self* (legitimate users, uncorrupted data, etc.) from dangerous *other* (unauthorized users, viruses, and other malicious agents). This method is intended to be complementary to the more traditional cryptographic and deterministic approaches to computer security. As an initial step, the negative-selection algorithm (discussed in the previous section) has been used as a file-authentication method on the problem of computer virus detection.

3.1.1 Virus Detection: In this application, Forrest et. al. [10] used the negative-selection algorithm to detect changes in the protected data and program files. A number of experiments are performed in a DOS environment with different viruses, including file-infecter and boot sector virus samples. Reported results showed that the method could easily detect the modification that occurred in the data files due to virus infection. Compared to other virus detection methods, this algorithm has several advantages over the existing change detection methods: it is probabilistic and tunable (the probability of detection can be traded off against CPU time), it can be distributed (providing high system-wide reliability at low individual cost), and it can detect novel viruses that have not previously been identified.

However, since the stored information in a computer system is volatile in nature, the definition of self in computer systems should be more dynamic than in the case of natural immune systems. For example, computer users routinely load in updated software systems, edit files, or run new programs. So this implementation seems to have limited use - only to protect static data files or software.

3.1.2 UNIX Process Monitoring: As an on-going research on computer security, Forrest et al. [8] studied the proposed negative selection algorithm to monitor UNIX processes. The purpose is to detect harmful intrusions in a computer system. This implementation aimed at identifying a *sense of self* for UNIX processes, they redefined self to accommodate the legitimate ac-

tivities in dynamic computer environment so that the definition is sensitive to malicious attacks.

This work is based on the assumption that the system calls of root processes are inherently more dangerous to cause damage than user processes. Also root processes have a limited range of behavior, and their behavior is relatively stable over time. The *normal or self* is defined by short-range correlations in a process' system calls. This definition of self seems to be stable during normal behavior for several standard UNIX programs. Further, it is able to detect several common intrusions involving *sendmail*. Their reason of monitoring *sendmail* is that its behavior is sufficiently varied and complex that it provides a good preliminary test, and there are several documented attacks against *sendmail* that can be used for testing. The experiments generated traces of three types of behavior that differ from that of normal *sendmail*: traces of successful *sendmail* attacks, traces of *sendmail* intrusion attempts that failed, and traces of error conditions. They have been able to execute and trace two attacks.

These preliminary experiments [8] suggest that short sequences of system calls provide a stable signature that can detect some common sources of anomalous behavior in *sendmail*. Because the current measure is easy to compute and is relatively modest in storage requirements, it would be plausible to implement it as an on-line system, in which the kernel checks each system call made by processes running as root. Under this scheme, each site would generate its own normal database, based on the local software/hardware configuration and usage patterns. One advantage of using local usage patterns is that every site would then have its own unique identity, slightly different from everyone else. This would mean that a successful intrusion at one site would not necessarily be successful at all sites running the same software, and it would increase the chance of at least one site noticing an attack. This work appears to be very promising and opens new venue in computer security research.

3.1.3 An alternative approach to Virus Detection: Kephart suggested another immunologically inspired approach for virus detection [23]. In this approach, known viruses are detected by their computer-code sequences (signatures) and unknown viruses by their unusual behavior within the computer system.

In this immunity-based method, a diverse suit of *decoy programs* are kept at different strategic areas in memory (e.g. home directory) to capture samples of viruses. Decoys are designed to be as attractive as possible to trap those types of viruses that spread most successfully. Each of the decoy programs is examined from time to time, to see if it has been modified. If one or more have been modified, it is almost certain that an unknown virus is loose in the system, and each

of the modified decoys contains a sample of that virus. In particular, the infected decoys are processed by - *the signature extractor* - so as to develop a recognizer for the virus. It also extract information from the infected decoys about how the virus attaches to it's host program (attachment pattern of the virus), so that infected hosts can be repaired. The signature extractor must select a virus signature (from among the byte sequence produced by the attachment derivation step) such that it can avoid both false negatives and false positives while in use. In other words, the signature must be found in each instance of the virus, and it must be very unlikely to be found in uninfected programs. Once the best possible signature is selected from candidate signatures of the virus, it run against a half-gigabytes corpus of legitimate programs to make sure that they do not cause false positive. The repair information is checked out by testing on samples of the virus, and further by the human expert.

Finally, the signature and the repair program is stored in archive of the AntiVirus database, and the updated (new) version is distributed to the customers. According to Kephart, this approach will also be used to stop the spreading of viruses in networked computers, where infected machines send out "kill signals" to warn other computers of the rampant virus. The signals tell how to kill the new virus as well as similar one.

However, it is not clear how the repair program works in different circumstances. Moreover, decoy programs should have some special characteristics to trap the viruses (no example of such decoy program is given in the paper [23]). Also keeping them in different strategic locations in a computer system is crucial for its success.

3.2 Anomaly Detection in time series data

Dasgupta and Forrest [3], [5] experimented with several time series data sets (both real and simulated) to investigate the performance of the negative selection algorithm [10] for detecting anomaly in the data series. The objective of this work is to develop an efficient detection algorithm that can be used for noticing any changes in steady-state characteristics of a system or a process. In these experiments, the notion of self is considered as the normal behavior patterns of the monitored system.⁵ So, any deviation that exceeds an allowable variation in the observed data, is considered as an anomaly in the behavior pattern. This approach relies on sufficient enough sample of normal data (that can capture the semantics of the data patterns) to gen-

⁵It is assumed that the normal behavior of a system or a process can often be characterized by a series of observations over time. Also the normal system behavior generally exhibit stable patterns when observed over a time period.

erate a diverse set of detectors that probabilistically detect changes without requiring prior knowledge of anomaly (or faulty) patterns.

They applied the algorithm for "The Tool Breakage Detection" in a milling operation [4]. The tool breakage detection problem is formulated as the problem of detecting temporal changes in the cutting force pattern that results from a broken cutter. That is, the new data patterns are monitored to check for whether or not the current pattern is different from the established normal pattern, where a difference (i.e. a match in the complement space) implies a shift in the cutting force dynamics.

This detection algorithm was successful in detecting the existence of broken teeth from simulated cutting force signals in a milling process. The results suggest that the approach can be used as a tool for automated monitoring of safety-critical operations.

3.3 Fault Diagnosis

Ishida [19] studied the mutual recognition feature of the immune network model [22] for fault diagnosis. In his implementation, fault tolerancy was attained by mutual recognition of interconnected units in the studied plant. That is, system level recognition was achieved by unit level recognition.

The model has the following properties: (a) it has the ability to do parallel processing, (b) can handle incomplete information and data, (c) it is self-organizing, and (d) no feedback loop is necessary in the failure propagation. Ishida and Mizessyn [18] presented an application of this mutual recognition model to the process instrumentation system of a chemical plant. Using the relationship among sensors, sensor network are constructed by bi-directional arcs, in order to apply the model for fault diagnosis. The results are very promising and worth further investigation.

Ishiguro [20] applied the immune network model [22] to on-line fault diagnosis of plant systems. To apply the immune network to plant fault diagnosis, the following assumptions were made: (a) Sensors are not equipped with all components of the plant systems, and they inform the state of the equipped component as binary states, i.e fault-free (normal) or faulty (abnormal), (b) the number of failure origins is assumed to be one (namely, simultaneous and complex failure are not taken into account), (c) failure states propagate through branches without exceptions, and (d) no feedback loop exists in the failure propagation. This work attempts to develop an integrated fault diagnosis method which can be used in industrial plants.

3.4 AIS for Pattern Recognition

Hunt and Cooke [17] investigated an Artificial Immune System (AIS) based on the theory of immune network [22] within the context of machine learning. The AIS offers noise tolerant, unsupervised learning within a system which is self-organizing, does not require negative examples and explicitly represents what it has learnt. Such a system combines the advantages of learning classifier systems with some of the advantages of neural networks, machine induction and case-based retrieval.

The operation of the AIS comprises a root object, a network of cells, a teaching data set and a test data set. Each cell in the network possess a pattern matching element which is generated by mimicking the genetic mechanisms by which antibodies are formed in the natural immune system. This enables complex vocabularies and promotes diversity of the pattern matching elements. The system exhibits two types of response: primary and secondary. The primary response is the learning phase when the AIS learns about patterns in the input teaching data. The secondary response represents a pattern recognition process during which the AIS attempts to classify new data relative to the data it has seen before.

To apply the AIS to a particular problem, it is first taught with a sample teaching set in a one shot or an incremental manner (depending on the problem). The information learnt can then be exploited in a number of ways. They have shown the potential of AIS on a pattern recognition problem. The AIS was applied to the recognition of promoters in DNA sequences, i.e. to determine whether new sequences were promoter containing or promoter negative.

Hunts and Cooke's work [17] demonstrated that how the AIS can represent a powerful example of learning within an adaptive non-linear network that contains an explicit content addressable memory. Their research objective is to develop a immunity-based toolkit for machine learning applications.

4 SUMMARY

The natural immune system is a subject of great research interest because of its powerful information processing capabilities. In particular, it performs many complex computations in a completely parallel and distributed fashion. Like the nervous system, the immune system can learn new information, recall previously learned information and performs pattern recognition tasks in a decentralized fashion. Also its learning takes place by evolutionary processes similar to biological evolution. The paper reviews the models that have been developed based on various computational aspects of the immune system. Because of the page re-

strictions, we are unable to provide a detailed account of each of the models and their applications. However, the existing immunity-based methods emulate one or the other mechanisms of the natural immune system. Further study should integrate all the potentially useful properties in a single framework in order to develop a robust immunity-based system.

There are many potential application areas in which immunity-based models appear to be very useful. They include fault detection and diagnosis, machine monitoring, signature verification, noise detection, computer and data security, image and pattern recognition, and so forth. It is to be noted that the mechanisms of the immune system are remarkably complex and poorly understood, even by immunologists. If we can understand the functionalities and the inherent mechanisms of various components of the immune system from the computational viewpoint, we may gain better insights about how to engineer massively parallel adaptive computations.

REFERENCES

- [1] H. Bersini and F. J. Varela. Hints for adaptive problem solving gleaned from immune networks. In *Proceedings of the first workshop on Parallel Problem Solving from Nature*, pages 343–354, 1990.
- [2] Franco Celada and Philip E. Seiden. A computer model of cellular interactions in the immune system. *Immunology Today*, 13(2):56–62, 1992.
- [3] Dipankar Dasgupta. Using Immunological Principles in Anomaly Detection. In *Proceedings of the Artificial Neural Networks in Engineering (ANNIE'96)*, St. Louis, USA, November 10-13 1996.
- [4] Dipankar Dasgupta and Stephanie Forrest. Tool Breakage Detection in Milling Operations using a Negative-Selection Algorithm. Technical Report CS95-5, Department of Computer Science, University of New Mexico, 1995.
- [5] Dipankar Dasgupta and Stephanie Forrest. Novelty Detection in Time Series Data using Ideas from Immunology. In *ISCA 5th International Conference on Intelligent Systems*, Reno, Nevada, June 19- 21 1996.
- [6] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: algorithms, analysis, and implications. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, Oakland, CA, May 1996.
- [7] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Physica D*, 22:187–204, 1986.
- [8] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for unix processes. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, Oakland, CA, 1996.
- [9] S. Forrest, B. Javornik, R. Smith, and A. S. Perelson. Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, 1(3):191–211, 1993.
- [10] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-Nonself Discrimination in a Computer. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 202–212, Oakland, CA, 16-18 May 1994.
- [11] Steven A. Frank. *The Design of Natural and Artificial Adaptive Systems*. Academic Press, New York, M. R. Rose and G. V. Lauder edition, 1996.
- [12] C. J. Gibert and T. W. Routen. Associative memory in an immune-based system. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 852–857, Seattle, July 31-August 4 1994.
- [13] Paul Helman and Stephanie Forrest. An Efficient Algorithm for Generating Random Antibody Strings. Technical Report Technical Report No. CS94-7, Department of Computer Science, University of New Mexico, 1994.
- [14] R. Hightower, S. Forrest, and A.S. Perelson. The evolution of emergent organization in immune system gene libraries. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, Pittsburg, 1995. Morgan Kaufmann, San Francisco, CA.
- [15] Geoffrey W. Hoffmann. A neural network model based on the analogy with the immune system. *Journal of Theoretical Biology*, 122:33–67, 1986.
- [16] John E. Hunt and Denise E. Cooke. An adaptive, distributed learning system, based on the immune system. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 2494–2499, 1995.
- [17] John E. Hunt and Denise E. Cooke. Learning using an artificial immune system. *Journal of Network and Computer Applications*, 19:189–212, 1996.
- [18] Y. Ishida and F. Mizessyn. Learning Algorithms on an Immune Network Model: Application to Sensor Diagnosis. In *Proceedings of International Joint Conference on Neural Networks*, volume I, pages 33–38, China, November 3-6 1992.
- [19] Yoshiteru Ishida. An Immune Network Model and its Applications to Process Diagnosis. *Systems and Computers in Japan*, 24(6):38–45, 1993.
- [20] A. Ishiguru, Y. Watanabe, and Y. Uchikawa. Fault Diagnosis of Plant Systems Using Immune Networks. In *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI' 94)*, pages 34–42, Las Vegas, October 2-5 1994.
- [21] N. K. Jerne. The immune system. *Scientific American*, 229(1):52–60, 1973.
- [22] N. K. Jerne. Towards a network theory of the immune system. *Ann. Immunol. (Inst. Pasteur)*, 125C:373–389, 1974.
- [23] Jeffrey O. Kephart. A biologically inspired immune system for computer. In *Proceedings of Artificial Life*, Cambridge, M.A., July 6-8 1994.
- [24] Ronald R. Mohler, Carlo Bruni, and Alberto Gandolfi. A System Approach to Immunology. *Proceedings of the IEEE*, 68(8):964–990, 1980.
- [25] J. K Percus, O. Percus, and A. S. Person. Predicting the size of the antibody combining region from consideration of efficient self/non-self discrimination. *Proceedings of the National Academy of Science*, 60:1691–1695, 1993.
- [26] Alan S. Perelson. Immune network theory. *Immunological Reviews*, (10):5–36, 1989.
- [27] Francisco J. Varela and John Stewart. Dynamics of a class of immune networks I. Global Stability of idiotypic interactions. *Journal of Theoretical Biology*, 144(1):93–101, 1990.
- [28] Frank T. Vertosick and Robert H. Kelly. Immune network theory: a role for parallel distributed processing? *Immunology*, 66:1–7, 1989.
- [29] Richard G. Weinand. Somatic mutation, affinity maturation and antibody repertoire: A computer model. *Journal of Theoretical Biology*, 143(3):343–382, 1990.
- [30] Gerard Weisbuch. A shape space approach to the dynamics of the immune system. *Journal of Theoretical Biology*, 143(4):507–522, 1990.