

# Chapter 7. An Integrated AIS: Extended Dynamic Clonal Selection Algorithm

## 7.1 Introduction

In a real environment, self behaviours change after a certain period, and only a small subset of self antigens are shown at any one time. In order for the AIS to be able to deal with such an environment, the dynamic clonal selection algorithm (DynamICS) was introduced in chapter six. The results described there show that DynamICS could learn incrementally the globally converged distributions even though only one subset distribution is given at each generation. This feature was achieved by employing three important parameters, which are the tolerisation period, activation threshold and life span. However, DynamICS could not learn new self-antigens when learned self and non-self behaviours suddenly altered due to legal self change. This resulted in high FP rates when new antigens are monitored by DynamICS, although it produced high TP rates. The proposed explanation of this outcome was that the generated memory detectors had never been exposed to a certain antigen cluster within their tolerisation periods. Thus they could not have tolerance against a complete self set.

This chapter investigates a further extension of DynamICS, so that it can reduce FP rates increased by memory detectors. As one way to decrease the FP rates caused by memory detectors, the extended DynamICS handles generated memory detectors based on their detection results. DynamICS preserved memory detectors for an infinite lifespan. In contrast, the extended DynamICS kills memory detectors if they show poor self-tolerance to new antigens (section 7.4), [Kim and Bentley, 2002b]. This extended system is tested to determine whether surviving memory detectors no longer cause seriously high FP error rates or not. From this test, it is analysed whether any other problems occur as a consequence of killing memory detectors. The analysis shows that the extended DynamICS requires a larger amount of costimulation if it yields high TP rates. This analysis leads the extended DynamICS to employ hypermutation to simulate the evolution of a gene library (section 7.6), [Kim and Bentley, 2002c]. This additional extension is designed to fine-tune generated memory detectors so that the system obtains higher TP rates without increasing the amount of costimulation. The new extension is tested to determine whether it gains high TP rates without increasing the amount of costimulation as the result of gene library evolution. The test results are then analysed to see how a hypermutation leads to such a gene library evolution effect, and thus whether it improves the overall system performance. Finally, the novel features of DynamICS studied in this thesis are discussed in accordance with the comparison to the most similar AIS developed by [Hofmeyr, 1999; Hofmeyr and Forest, 2000].

## 7.2 DynamiCS Revisited

The experiments in the previous chapter simulated a situation in which converged behaviours learned in an incremental way are suddenly altered due to legal self change. The results of these experiments show that a value of  $T$  which was sufficiently large to show perfect FP rates no longer demonstrated satisfactory FP rates. More precisely, four experiments were performed when four different values, {5, 10, 20, 30}, were given to  $N$ : the number of generations that antigens are selected from a same cluster. In these four experiments, it was observed that the overall TP's and FP's increase as  $N$  grows. Particularly, when  $N$  is as large as a given  $T$ , which is the tolerisation period, the obtained FP rates reached high values greater than 0.3 (see figure 6.13 in chapter 6). The inference from this result was that some memory detectors have never been exposed to a certain antigen cluster, and thus those memory detectors caused high FP rates. For this reason, the extension of DynamiCS introduced in this chapter will handle generated memory detectors based on their detection results.

Before introducing the extended DynamiCS, another set of experiments was performed by giving different values for  $A$ , the activation threshold of a mature detector, but using smaller values than the ones used in section 6.6. These values are shown in table 7.1. For consistency of experiments throughout the current chapter, another four experiments were performed with the parameter values presented in table 7.1.

Parameters	Values
Tolerisation Period ( $T$ )	30
Life Span of Mature Detectors ( $L$ )	10
Activation Threshold of Mature Detectors ( $A$ )	{5, 10, 20, 40}
Number of Generations that Antigens are Selected from a Same Cluster ( $N$ )	30

**Table 7.1 Parameter values used for DynamiCS experiments**

Figure 7.1 illustrates the results of these four experiments. The experiments were run five times and average results of five runs are shown in figure 7.1. The X-axes of these graphs represent the number of generations and the Y-axes indicate detection rates. Each graph has two lines, one displaying a True Positive (TP) rate and another showing a False Positive (FP) rate. The grid lines on the X axis were placed at every  $N$  generations for  $N = 30$ . Each experiment was run for a maximum of 2000 generations.

It has already been seen in sections 6.5.1 Effect of the Tolerisation Period and 6.5.2 Effect of the Activation Threshold that large  $A$  and  $T$  values can prevent self antigen detection to some extent. The results shown in figure 7.1 also demonstrate the same consequence verified in those sections: large  $A$  and  $T$  values reduce very high FP. A relatively large  $T$  value, 30, is given in all the four experiments and increasing  $A$  value reduces high FP rates. Nevertheless, it still shows unacceptably high FP rates, around 0.5, even though FP rates drop by nearly half when the large

**Figure 7.1. TP and FP rates when  $A$  varies and  $T = 30, L = 10, N = 30$**

**Figure 7.2 TP and FP rates when  $A$  varies and  $T = 30, L = 10, N = 30$  without memory detectors**

A value, 40, is used. This implies that the memory detectors which detected non-self antigens were not sufficiently self-tolerant.

However, there are two types of detector that are qualified to detect antigens: memory detectors and mature detectors that have just activated. Although it was inferred from previous experimental results that memory detectors which have never been exposed to a certain antigen cluster could cause high FP rates, it has not been shown yet whether memory detectors or mature detectors that have just activated are the actual cause of this problem. In order to clarify this issue, another set of experiments was performed. In the new experiments, DynamiCS did not generate memory detectors. When mature detectors activated, they produced detection signals but they were not converted into memory detectors. Instead, they simply died off. As the result, antigen detection was performed only by activated mature detectors in the absence of memory detectors. Figure 7.2 shows these new experimental results. The four important parameters,  $T$ ,  $A$ ,  $L$ , and  $N$ , have identical values to those used in the experiments above and they are summarised in table 7.1.

The four experimental results in figure 7.2 display similar outcomes regardless of  $A$  values: low TP and FP rates. This verifies the important role of memory detectors. They indeed contribute to increase TP rates by detecting re-encountering antigens. Without memory detectors, TP rates of DynamiCS fluctuate irregularly within an unsatisfying range (between 0.1 and 0.8). The low FP rates revealed in figure 7.2 also imply that the high FP rates shown in figure 7.1 are originated from detection by memory detectors.

The results exhibited in figure 7.2 makes it clearer that DynamiCS needs an appropriate way to handle memory detectors. In order to propose a new way of handling generated memory detectors, the next section briefly introduces how the human immune system maintains lifetime lasting memory while it continues to sustain self-tolerance. It also presents the method used to ensure that these human mechanisms have been implemented in available AIS's, in order to acquire artificial immune memory.

## 7.3 Related Work: Handling Memory Detectors

### 7.3.1 Human Immune Memory

Immunologists define *immune memory* as the capability of the immune system that can fully protect the body from the re-attack of pathogens, which have previously been detected. Immune memory is long-lived, often lasting for many years, even for the lifetime of an individual [Tizard, 1995]. The life-long immune memory means that a quick immune response to reappearing pathogens lasts for the lifetime despite constant and unpredictable generation of new memory

cells, triggered by new antigen detection. Experimental observation showed that the memory cell population is maintained at a roughly constant size within an individual's body for his or her lifetime from puberty onwards [Yates and Callard, 2001], although there is gradual addition of new memory cells in old age. These two observations have raised a question of how the immune system maintains a roughly constant size of memory detectors while it continues to maintain immune memory of various types of pathogens that occur during a lifetime.

Several pieces of research by different immunologists attempted to explain the immune memory mechanism from various angles. For instance, one theory by Mackay [1993] explained this by showing the life-long lifespan of memory cells and another theory [Matzinger, 1994] described immune memory being provoked by constant re-stimulation of memory cells by reappearing antigens. In contrast to these theories, there is an observation of maintaining a roughly constant amount of memory cells in the absence of repeated exposure to antigens or cross-reactive stimulation [Yates and Callard, 2001]. Yates and Callard showed that a small minority of memory cells are susceptible to programmed death triggered by contact with other memory cells. Especially when memory T-cells proliferate, matching the receptors of other T memory cells triggers signal cascade leading to programmed death of T memory cells. This theory showed that the regulation of a stable population of memory cells is achieved in absence of antigen stimulation.

These interpretations can be combined together into one abstract explanation, which is the idiotypic based immune network theory proposed by Jerne [1974]. As described earlier in the section 2.4.4 Immune Network Theory of chapter 2, the immune network theory emphasises that the continuous chain of stimulation by antigens and suppression by other antibodies can form a large-scale network, and the final equilibrium status between suppression and stimulation determines the overall internal memory of the immune system. Therefore, the stabilised immune network constructed by proliferation by antigens and suppression by other antibodies constitutes a converged memory cell population.

Likewise, many studies in immunology approached the understanding of how to maintain a converged memory cell population as one step towards finding an explanation of lifetime lasting immune memory. Although there is no clear answer yet, the common explanation from these studies is that a memory cell population stabilises through constant death of existing memory cells, recruitment and proliferation of new memory cells. That is to say that a roughly constant size of memory cells is maintained not by keeping memory cells in a static way, but by continuous loss and new birth of memory cells in a dynamic way.

Although these studies illustrated how a stabilised memory cell population is maintained, they did not clearly explain how a stabilised memory cell population also maintains robust memory against various types of antigens and how it shows the associative property. The study by Smith *et al.* [1996] has attempted to explain the associative property of immune memory using Sparse Distributed Memory (SDM). SDM was originally introduced by Kanerva [1988] in order to store a very large number of data items into a memory space, which is mapped to a smaller number of physical data addresses. It approximately addresses given data items to a memory space when data is written. This means that the data item is recalled by an address sufficiently similar, but not necessarily equal, to the original address. This approximate addressing maps sparse and distributed physical addresses to logical addresses that is much more dense than existing physical addresses. Smith *et al.* [1996] took the view that distributed scattered physical addresses in SDM has an equivalence with memory cells in a stabilised memory cell pool, and that the approximated recalling mechanism of SDM is also equivalent to the primary and secondary responses of memory cells. Consequently, Smith *et al.* [1996] claimed that immune memory is robust and associative, as is SDM, since both employ a similar approximate addressing mechanism.

### **7.3.2 Artificial Immune Memory**

The immune memory of the human immune system has been implemented in various ways in different AIS's. The common feature of these implementations is that the immune memory was achieved in an implicit way without having a separate memory detector/antibody population [Dasgupta, 1998b; De Castro and Von Zuben, 1999; Timmis, 2001]. Rather, only one type of antibody population was used and the antibody population was usually maintained at a constant size. That is to say that the antibody population was maintained through constant death of existing antibodies and recruitment and proliferation of new antibodies. During this process, naïve antibodies (i.e. newly generated) and a surviving antibody antibodies (i.e. memory antibodies) remain in the same antibody population and compete with each other for survival. Unlike the human immune system, where there are two different types of immune cell population (a memory cell and non-memory cell pool) and the competition between immune cells is only within each type of population [Tanchot *et al.*, 2000], these AIS's did not label memory cells separately and thus they compete with other maturing and naïve immune cells for survival.

Among the AIS's which use only one antibody population, immune network theory has been a popular approach to make immune memory emerge by itself within an AIS [Timmis, 2001; Farmer *et al.*, 1986; Varela *et al.*, 1988]. The AIS's employing the network theory formed the immune network as the result of immune pattern recognition. The specific shape of the immune network described the immune memory of the given immune system. The memory that emerges, which is a stabilised network structure, was also used to handle a dynamic environment. When a new antibody is generated and inserted into already formed immune network, this new antibody

competes with other ones that are already in the network. The new network formed by the surviving antibodies is expected to provide a new solution to a new environment without losing the solutions to the previous environment. This was possible because the AIS decides on surviving antibodies in the network not only by their antigen stimulation level but also by their antibody suppression level. Although some antibodies did not receive a sufficient degree of stimulation from new antigens, they would not be deleted as long as they were not the subject of large suppression from other antibodies. These antibodies can remain and act as memory cells in the AIS.

Another type of immune memory employed for AIS's is Sparse Distributed Memory (SDM) [Smith *et al.*, 1996]. Hart and Ross [2001] adopted SDM in their co-evolutionary GA to cluster moving data. Immune memory was not explicitly implemented as a separate antibody population in either work. Instead, the SDM was used for antibody and antigen matching and recall. It lets each antibody vote (i.e. match and recall) several antigens instead of one antigen. Thus, when a new antigen is presented, the democratic result from all antibodies decides the label of the antigen, whether self or non-self. This kind of antibody and antigen matching and recalling mechanism showed an implicit immune memory feature by allowing one antibody to match more than one antigen.

Another work by Gaspar and Collard [1999] investigated an artificial immune system for a time dependent optimisation (TDO) problem. Their simple artificial immune system (Sais) was implemented by adding several immune system features (such as clonal selection, immune network theory, hypermutation) to a conventional GA. In this work, Gaspar and Collard have shown what affected system robustness, obtained through immune memory. *Robustness* is the ability to maintain diverse optima without losing previously encountered optima. This feature was expected to allow the system to provide solutions quickly when previously presented optimal functions are later given as targets. The experimental results illustrated that Sais showed stronger robustness than other types of GA. The stronger robustness of Sais was achieved by memorising previously found optima using idiotypic immune network selection. However, the good improved robustness resulting from the memory of previously encountered optima, was not maintained as the number of different optimisation targets increased.

In contrast to above approaches, Hofmeyr's AIS [1999] adopted a separate memory detector population that was isolated from other detector/antibody populations. Memory detectors in Hofmyer's AIS also had two significant features: quicker response and infinite life span. This system is the only AIS to provide immune memory by directly mimicking the memory cells of the human immune system. Immune memory was no longer maintained implicitly in this system. Instead, it had explicit antibodies to retain memory of previously detected antigens, and these



antibodies were treated differently from other antibodies. Although the initial life spans of memory antibodies were set to be infinite, they could be deleted when the number of existing memory antibodies reached the pre-defined maximum number. If the number of memory detectors was more than this number, randomly selected memory detectors were killed until the number of current memory detectors, including the newly generated memory detectors, reached the maximum number of memory detectors.

This section has introduced various types of artificial immune memories: those based on network theory, SDM and an explicit memory population. Among them, an explicit memory population seems to have an advantage over the two types of implicitly emerging immune memory. As reported in Gaspar and Collard's work [1999], the AIS without an explicit memory population failed to maintain its memory when the number of required antibodies grew in order to cover all the existing niches in a solution space. This was because these antibodies competed with newly generated antibodies that were more stimulated by current antigens. It might always be more likely for new antibodies to dominate antibodies memorising past antigens, since their antigen stimulation level is always higher than the memory antibody's antigen stimulation level. When the number of required antibodies is not large, they can still remain in the antibody population alongside the new antibodies. However, when the number of required antibodies grows, they cannot always remain in the antibody population and thus the AIS prefer currently stimulated antibodies to other memory antibodies. Thus, some memory antibodies will be lost. On the other hand, memory antibodies in an explicit memory population do not compete with new antibodies and thus memory antibodies would not be lost as the expense of space for new antibodies. To benefit from this advantage, DynamiCS uses an explicit memory population to maintain memory detectors.

Although this work has extensively studied how the human immune system maintains its immune memory and also how AIS's obtain their memory, it is not very clear how either of these systems stop self-detection of previously generated memory detectors. However, there is still one suggestion from this study that can be directly used: replacing memory detectors. As Yates and Callard [2001] have shown, memory detectors are constantly replaced while the population size is roughly constant. Following this understanding, the extended DynamiCS constantly replaces memory detectors. The next section describes one approach to replacement of memory detectors, via memory detector costimulation.

## **7.4 Extended DynamiCS: Killing Memory Detectors**

### **7.4.1 Algorithm Description**

All the generated memory detectors in DynamiCS have infinite life span and an activation threshold of one. However, this is quite different from what really happens to memory cells in the

human immune system. (See section 7.2 Related Work: Handling Memory Detectors). Although memory cells have a much lower activation threshold and a longer life span than other maturing cells, the memory cell population stabilises through constant death of existing memory cells, recruitment and proliferation of new memory cells. The infinite life span of memory detectors adopted by DynamiCS is not a biologically inspired idea. Thus, instead of giving an infinite life span to generated memory detectors, the extended DynamiCS kills memory detectors based on their current detection results. If antigens that are newly detected by memory detectors turn out to be self-antigens, these memory detectors are deleted from the memory detector population. This modification mimics the costimulation of memory detector detection. To be precise, whenever a memory detector in the memory detector population detects any antigen, it asks for confirmation about whether the detected antigen is self or non-self from a human officer. It sends a detection signal only if the human officer confirms that the detected antigen is non-self, otherwise it is deleted. Thus, the extended DynamiCS deletes harmful memory detectors by applying costimulation to memory detectors as it does to activating mature detectors.

#### 7.4.2 Experiment Results

Four different experiments were performed to test whether the extended DynamiCS can reduce the high FP rates observed in the previous experiment. The extended DynamiCS was set with the same parameter values used in the experiments reported in 7.3 DynamiCS Revisited and they are summarised in table 7.1.

Figure 7.3 illustrates the results of four different experiments. These results can be compared to those in figure 7.1. Regardless of  $A$ , all of these four results show reasonably low FP rates, which are mostly lower than 0.1. This outcome is clearly different from the ones seen in figure 7.1, which has much higher FP rates and was much more sensitive to various  $A$  values. In figure 7.1, as  $A$  increases, the FP rates drop rapidly. In contrast, the changes of the FP rates in figure 7.3 are not clearly noticeable depending on various  $A$  values. In addition, the TP rate changes found in figure 7.1 and 7.3 are quite different. The TP rates shown in figure 7.3 decrease to a much greater extent compared to the TP rate changes observed in figure 7.1. In summary, as  $A$  increases, the amount of FP rate drop is much larger in the experimental results of original DynamiCS, while the degree of TP rate fall is much larger in the experimental results of the extended DynamiCS.

These different effects explain how useful memory detectors in each system are for detecting new non-self antigens without mistakenly detecting self antigens. In the original DynamiCS, there are some memory detectors that detect self-antigens mistakenly and thus cause high FP rates. The generation of these memory detectors was prevented to some extent by restricting the conditions that allow mature detectors to be memory detectors. A large value for  $A$  in original DynamiCS did this job, and the large FP rate drop in figure 7.1 was obtained due to large  $A$ . Nevertheless, it

**Figure 7.3 TP and FP rates when  $A$  varies and  $T = 30, L = 10, N = 30$  after killing memory detectors**

has not yet gained satisfactory FP rates with a quite large value, 100, for  $A$  (see figure 6.13 in Chapter 6) and also large  $A$  caused TP rates to decline. In contrast, for extended DynamicCS, memory detectors that caused high FP rates could not survive and thus FP rates were consistently low regardless of  $A$ 's value. Extended DynamiCS only kept memory detectors that were useful for detecting non-self antigens without detecting self antigens. For the same reason, large  $A$  reduced the number of useful memory detectors and it resulted in lower TP rates.

Furthermore, the new strategy of the extended DynamiCS affects detection of non-self antigens. Compared to the original DynamiCS, it is much harder for memory detectors to survive in the extended DynamiCS. Table 7.2 shows the total number of surviving, generated and deleted memory detectors for a total of two thousand generations. These numbers are averaged across five runs. Thus, the average numbers of surviving memory detectors are smaller than the ones in DynamiCS when the same values are given to other parameters (see table 7.2). Consequently, the extended DynamiCS gains higher TP rates when it has a more relaxed condition for the activation of mature detectors, as in the cases having small values for  $A$  (see figure 7.3). Thus, the extended DynamiCS was able to obtain high TP rates and low FP rates when it had a small value for  $A$ .

	DynamiCS			Extended DynamiCS			
	Surviving Memory Detectors	Generated Memory Detectors	Deleted Memory Detectors	Surviving Memory Detectors	Generated Memory Detectors	Deleted Memory Detectors	Memory Detector CoStimulation per generation
$A = 5$	75.75 (8.2)	75.75 (8.29)	0	46.5 (3)	205.5 (8.03)	159 (8.33)	40.89 (4.48)
$A = 10$	49.5 (5.7)	49.5 (5.7)	0	32.75 (18.25)	124.25 (50.69)	91.5 (33.77)	29.36 (6.28)
$A = 20$	33.5 (1)	33.5 (1)	0	24.25 (14.25)	78.75 (5.62)	54.5 (3.83)	20.39 (8.35)
$A = 40$	20.5 (1.67)	20.5 (1.67)	0	14.5 (1.67)	55.25 (4.09)	40.75 (5.02)	16.43 (11.76)

**Table 7.2 Average numbers of surviving, generated and deleted memory detectors during 2000 generations, and average number of memory detector costimulations per generation for the DynamiCS and the Extended DynamiCS. The values in parentheses are variances.**

However, there is another issue to be concerned with in the application of the extended DynamiCS for intrusion detection. Since the extended DynamiCS cured a problem of DynamiCS by applying costimulation to memory detectors, and costimulation was implemented in extended DynamiCS by asking for confirmation from a human security officer, the large number of memory detector costimulations can hinder the adoption of the extended DynamiCS. Too much requirement for human intervention could render the extended DynamiCS useless. Thus, an effective IDS always requires the lowest frequency of costimulation per generation, leading to the least requirement for human intervention.

The amount of memory detector costimulation can be defined as the number of existing memory detectors per generation plus the number of mature detectors that have just activated (i.e. that just became memory detectors) per generation. The average numbers of memory detector costimulations per generation are shown in table 7.2. Although it is preferred for the extended DynamiCS to have smaller value of  $A$  because it leads to higher TP rates while sustaining low FP rates, this case tends to have larger number of memory detector costimulations. Thus, this approach, obtaining high TP rates and low FP rates by having small  $A$  values, does not seem to be ideal. Instead, these results suggest that large  $A$  can be more favourable than the case with small  $A$  if it can maintain a satisfactorily high TP rate. The experimental results require the extended DynamiCS to have a procedure to increase TP rates while it sustains a smaller number of memory detector costimulations. As one approach to this, the next section investigates applying hypermutation for simulation of gene library evolution, as observed in the human immune system.

## 7.5 Related Work: Gene Library Evolution

A problem found in previous experimental results is that the extended DynamiCS required a large number of memory detector costimulations in order to obtain satisfactory TP rates. This problem could originate from the simplification of the developed AIS, which did not adopt all the evolution processes engaged in the human immune system. Three major evolution processes involved in the human immune system were identified in chapter 3. They are gene library evolution, negative selection and clonal selection. From these three processes, negative selection and clonal selection have already been employed in DynamiCS and their effects were analysed. However, gene library evolution has not yet been adopted in DynamiCS.

The analyses of previous experimental results explained that the extended DynamiCS with large  $A$  provided a smaller number of memory detectors and thus it required less involvement from human security officers. However, it missed a larger number of non-self antigens. In addition, they have shown that the generation of more memory detectors by decreasing  $A$  can increase TP rates. This was mainly because all the new detectors were generated randomly and thus generated detectors were randomly scattered in the non-self antigen space. In other words, although existing memory detectors detected a sufficient number of non-self antigens to activate, they can be further finely tuned to match more non-self antigens. For instance, if new detectors are generated by taking some feedback from previous detection results into account, then a new detector can be improved to match a larger number of non-self antigens. In addition, this idea of taking some feedback from previous detection results into account can also be implemented by gene library evolution using hypermutation. Bearing in mind this effect of gene library evolution, this section

briefly addresses how the human immune system evolves over generations, and how existing AIS's adopt these mechanisms in order to have evolved antibodies.

### 7.5.1 Gene Library Evolution by Human Immune Systems

The human immune system learns dynamically changing antigens via clonal selection. To be more precise, activating antibodies clone themselves and proliferate across different parts of the body. Cloning antibodies trigger a *somatic hypermutation* process. Somatic hypermutation mutates the portion of genes that are randomly selected from antibody clones. Mutated offspring of activating antibodies are expected to have wider variations in their antigen-matching genes. Mutants are quickly disseminated across the body and start detecting other types of antibodies. During this process, mutants and existing antibodies compete to detect more antigens and their antigen detection results determine their affinities. The antibodies with higher affinities survive longer and clone themselves more. It is known that clonal selection with hypermutation is essential for the human immune system to permanently learn newly appearing antigens [Paul, 1993; Sompayrac, 1999].

The somatic hypermutation mechanism is distinguished from mutation taking place in a germ line level<sup>1</sup>. While a germ line level of mutation occurs typically at a low rate, mutation applied on activating antibody clones operates at a much higher rate. Another different feature of somatic hypermutation is that it is applied only on a somatic level. It is known that the mutated genes of antibody clones cannot be directly written back to the DNA (or a gene library) of an egg or sperm cell. As a result, the genes of surviving antibody mutants are not passed onto the next generation of the immune system [Paul, 1993; Sompayrac, 1999].

However, it is also known that the learning results via clonal selection with hypermutation during a lifetime indirectly lead the evolution of a gene library in the human immune system over generations. Although the genes of useful antibody mutants are not directly inherited, individuals that had more useful mutants are more likely to survive against various types of pathogens. Thus, the gene libraries of these individuals are passed over generations and offspring having these inherited gene libraries are more likely to have an immune system with a good *capability* of producing useful mutants. This effect was proposed for the first time by Mark Baldwin in 1896 and named as the Baldwin effect [Baldwin, 1896].

Since the Baldwin effect was proposed, many researchers have attempted to understand the relation between learning and evolution in nature. For instance, Hinton and Nowlan [1987] have

---

<sup>1</sup> *Germ line manipulation* means the altering of the fertilized egg, the first cell in the embryo to be, so that the genetic changes will be copied into every cell of the future adult, including his or her reproductive cells.

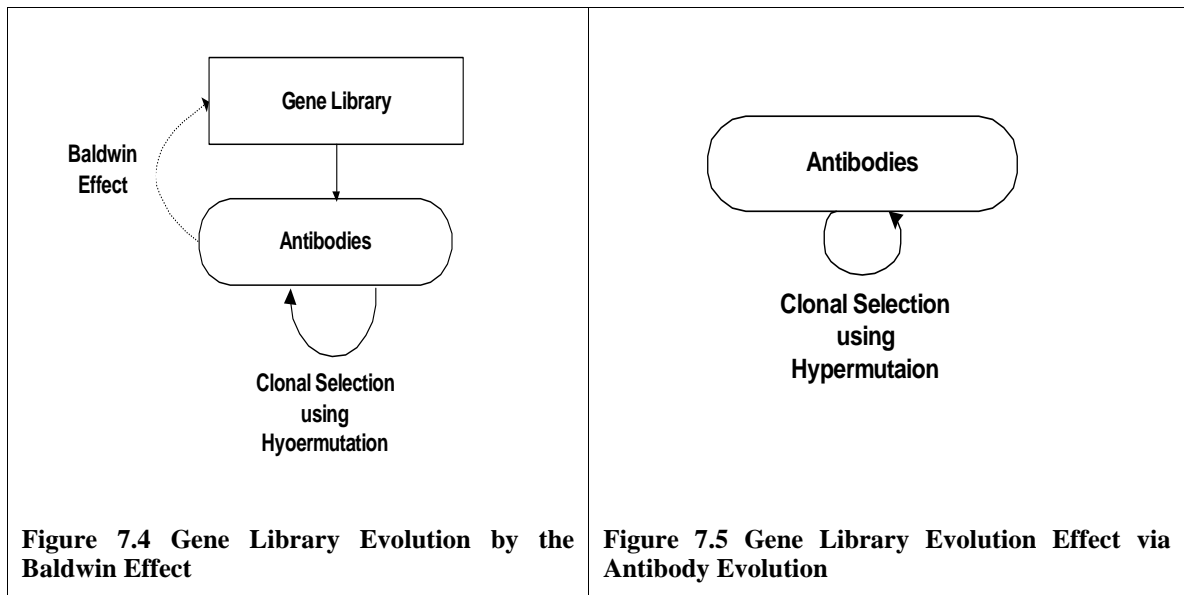
shown an example of the Baldwin effect from simple experiments that evolved appropriate neural network weights with and without learning. These experiments provided a specific example of the principle that “learning can guide evolution”. Hightower *et al.* [1996] have also reported that “learning accelerates evolution” by simulating the gene library evolution of the human immune system using GA. They used GA to let the gene library of a binary immune system evolve and the mutation of randomly selected bits was treated as learning. They measured the different average match scores of antibodies by varied learning rate,  $G$ , the number of randomly selected bits for mutation, and these scores were measured before mutation took place. Experimental results verified the Baldwin effect by showing that the scores increased as  $G$  increased, until a certain point. They also revealed that these scores decreased after this certain point if  $G$  continued to increase. Hightower *et al.* [1996] drew the interpretation that further mutation applied on antibodies after antibodies mastered given tasks could lead to the halt of gene library evolution.

While it has been reported that the learning of the human immune system during a lifetime indirectly determines the direction of gene library evolution [Hightower *et al.*, 1996; Perelson *et al.*, 1996], another work by Hightower *et al.* [1995] investigated what determines the direction of gene library evolution (i.e. where the selection pressure of gene library evolution is aimed). This question is about what the evolution strategy of the human immune system is when the goal is that a dynamically changing vast number of antigens should be covered by a much smaller number of antibodies. This work showed that the binary antibodies of AIS evolve toward a balancing point between maximum coverage of the antigen space and the least overlapping coverage of antibody space.

Oprea and Forrest [1998; 1999] advanced further the work by Hightower *et al.* [1995] and studied the diversity required of a gene library in the human immune system, and the role of gene library evolution. This work verified that antibody evolution gets slower and evolves to cover more random antigen niches when the pathogen size (exposed to antibodies) gets smaller. In this case, the immune system does not let the gene library evolve toward existing antigen specific niches. Instead, it evolves toward covering a coarse-grained antigen space. This understanding was drawn from the observation that the survival probability of the organism (the average fitness of immune systems) increased logarithmically with the size of its germ line-encoded antibody repertoire (the number of antibody genes in the library). This result clearly illustrated that the gene library diversity is not maintained for specific recognition of individual pathogens, but rather it evolved to cover a coarse-grained encoding of the regions of the pathogen universe that the species has encountered. A later study by the same author [Oprea, 1999] investigated the role of hypermutation by investigating its mutating targets. Her experiments showed that hypermutation usually targets to mutate the antigen-binding regions of a gene library and the mutation results often led fine-tuning of antigen-binding parts.

Likewise, the gene library evolves by getting indirect feedback from what the human immune system has learned during its lifetime. In addition, the germ line diversity that is obtained through gene library evolution is somewhat directed toward covering a coarse-grain antigen space, and learning through hypermutation leads the immune system to fine-tune its detection of the existing antigens.

## 7.5.2 Gene Library Evolution by Artificial Immune Systems



There are two methods employed by the currently available AIS's in order to evolve their gene libraries. The first approach directs gene library evolution through the Baldwin effect and the second approach allows provision of direct feedback from learning results to a gene library. The first approach initially builds a gene library that is a collection of previously known antibody genes (see figure 7.4). This initial gene library provides a certain degree of antigen diversity but it obtains a satisfactory level of antigen diversity through gene expression and learning using hypermutation. Although this approach does not directly alter the genes in the gene library, it still allows the gene library to evolve via the Baldwin effect. The second approach often does not distinguish a gene library from an antibody population (see figure 7.5). It treats a currently existing antibody population as a gene library and thus concentrates on antibody population evolution using learning through hypermutation. As the result, this approach ignores the difference between lifetime learning and evolution over generations, but it emphasises more the study of whether hypermutation accelerates the degree of antibody population evolution, and controls the evolution direction. These two different approaches have been implemented in various ways depending on the adopted AIS model.



## Gene Library Evolution with Gene Library

One popular group of AIS is the extension of a conventional genetic algorithm. Researchers added several immune features to GA in order to complement some weaknesses found from a conventional GA [Dasgupta *et al.*, 1999a; Hart *et al.*, 1998; Hart and Ross, 1999; Gaspar and Collard, 1999; Hajela and Yoo, 1999; Potter and De Jong, 1998; Nikolaev *et al.*, 1999; Michaud *et al.*, 2001]. The static clonal selection algorithm introduced in chapter 5 of this thesis belongs to this group. Among these systems, [Hart and Ross, 1999] and [Michaud *et al.*, 2001] used a gene library that is separate from the antibody population. The gene libraries used in these work are collections of some partial solutions and thus new antibody solutions were generated by concatenating these partial solutions. While Hart and Ross [1999] generated new antibodies using this method exclusively [Michaud *et al.*, 2001] generated only the initial antibody population using a gene library and the antibody population was evolved using a conventional GA. However, neither investigated whether these approaches have additional benefits compared with others that did not differentiate the antibody population from the gene library.

## Gene Library Evolution Effect via Antibody Evolution

Among those that add immune features to GA, a group of work that did not employ a explicit gene library typically generated new antibodies using crossover and mutation operators of GA and antibodies in the population were continuously replaced with evolved new ones [Dasgupta *et al.*, 1999a; Hart *et al.*, 1998; Gaspar and Collard, 1999; Hajela and Yoo, 1999; Potter and De Jong, 1998; Nikolaev *et al.*, 1999]. From these latter approaches, apart from [Gaspar and Collard, 1999], none of these systems employed hypermutation that might provide fine-tuned diversity of the antibody population that can cover currently existing antigens. The AIS developed in [Gaspar and Collard, 1999] cloned the best  $n$  % of antibodies and mutated them with a high rate. From these mutated antibodies, only ones having improved fitness values were entered to the antibody population for selection. Nevertheless, they also did not study the effect of hypermutation in terms of antibody evolution.

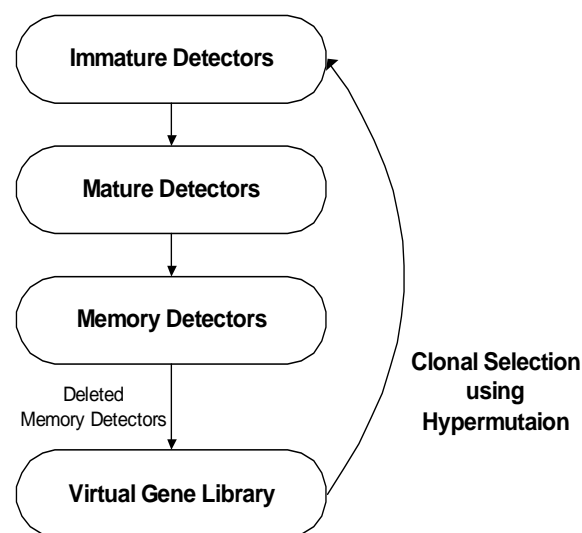
Another popular type of AIS, which use a network theory, usually apply a mutation operator to  $n$  % of best antibodies in an antibody network, and mutated antibodies are tested whether it is added to an existing immune network [Timmis, 2001; Fukuda *et al.*, 1998; Watanabe *et al.*, 1998a, b; Ishiguro *et al.*, 1997; Ishida, 1996a, 1997; Lee *et al.*, 1999a; Watanabe and Ishida, 2002]. From these AIS's, Timmis [2001] and Fukuda *et al.* [1998a] did not use a gene library to create initial antibody nodes while others [Fukuda *et al.*, 1998b; Watanabe *et al.*, 1998a, b; Ishiguro *et al.*, 1997; Lee *et al.*, 1999a; Ishida, 1996a; 1997; Watanabe and Ishida, 2002] initialised antibody nodes with already known local solutions, which can be regarded as a gene library. The systems

using a gene library typically developed an artificial immune network in order to find a global solution under a dynamically changing environment by finding an optimal combination of existing local solutions as a global solution. Among these systems, Timmis [2001], Fukuda *et al.* [1998], and Lee *et al.* [1999] applied a high rate of mutation when cloning new antibodies, and only Timmis [2001] investigated the different effects according to different rates of mutation. In this work, he has shown that the network connectivity declined as the mutation rate got higher and thus contributes to increasing the diversity of the antibody network.

Other work by [De Castro and Von Zuben, 2000a; De Castro and Von Zuben, 2001a] developed an AIS by mimicking exactly the clonal selection process without differentiating the gene library and the antibody population. When this system cloned new antibodies, it applied various mutation rates to each antibody depending on its affinity. It assigned smaller mutation rates when affinity is higher with the intention of increasing the diversity by correcting poorly performing antibodies. However, this work neither investigated the effect of mutation on the antibody population evolution, nor the need to have a separate gene library to accelerate antibody evolution.

## 7.6 Extended DynamiCS: Simulating Gene Library Evolution using Hypermutation

### 7.6.1 Algorithm Description



**Figure 7.6 Gene Library Evolution in the extended DynamiCS**

The problem found in previous experiments was that the extended DynamiCS achieved high TP rates only when it produced a large amount of memory detector costimulation. In contrast, for the case having a smaller amount of memory detector co-stimulation, extended DynamiCS struggled to show high TP rates. However, the related work introduced in the previous section indicates that applying hypermutation on immune cells for cloning is necessary to fine-tune existing immune

cells to target non-self antigen binding regions. As a way of resolving the problems associated with a large amount of costimulation, the extended DynamiCS applies hypermutation.

It can be inferred that the low TP rates obtained by the extended DynamiCS originate from coarse-grained non-self antigen niche coverage of activating detectors. Thus, if these detectors were fine-tuned to cover existing non-self antigens, the extended DynamiCS could have higher TP rates without necessarily having a large number of activating detectors. In order to investigate only the effect of hypermutation, the extended DynamiCS does not create a separate gene library (i.e. a collection of useful detector genes). This is shown in figure 7.6. Instead, it continues to maintain three detector populations, immature, mature and memory detectors, and treats a portion of memory detector population as a gene library. In order to let memory detectors to evolve towards existing non-self antigens without binding self antigens, the extended DynamiCS clones memory detectors by applying a hypermutation operator to deleted memory detectors.

These mutants of deleted memory detectors are added to the immature detector population for the negative selection test. Immature detectors in the DynamiCS have always been randomly generated for negative selection. Now the extended DynamiCS produces immature detectors from mutants of deleted memory detectors, if there are deleted memory detectors available or otherwise randomly. Hence, this further extension of the DynamiCS employs a “virtual gene library” dynamically made from mutations of deleted memory detectors. Through the various selection mechanisms and hypermutation operator, the seed immature detectors produced by the virtual gene library evolve over time, just as the immature, mature and memory detectors evolve in their separate populations. This modification is summarised in the pseudo code shown in figure 7.7.

```

If (immature detector population size + mature detector population size
    < non-memory detector pop size)
{
    Do
    {
        if ( number of deleted memory detectors > 0 && mutation rate != 0 )
        {
            Select a deleted memory detector randomly and create its mutant
            Add this mutant to an immature detector population.
        } else
            Generate a random detector and add it to an immature detector population
    }Until (immature detector population size + mature detector population size =
            non-memory detector pop size)
}

```

**Figure 7.7 Modified Pseudo Code for the Extended DynamiCS**

The above pseudo code is a modified version of lines 33 to 40 of figure 6.2 in chapter 6. At each generation, when the extended DynamiCS has to replenish the non-memory detector population by generating new immature detectors, it checks whether there are deleted memory detectors

available that the predefined mutation rate is not zero. If this is the case, new immature detectors are generated by mutating randomly selected memory detectors that were deleted from previous generations. However, if there are no memory detectors available or the predefined mutation rate is zero, the extended DynamiCS generates immature detectors randomly. The mutation operator used to create a mutant adopts a significantly higher mutation rate than that usually used in GA. While the mutation rate used in GA is very low at around 0.01%-0.05%, the extended DynamiCS employs much higher mutation rates of 0.1%-0.2%. This corresponds to the mutation strategy of the human immune system. The human immune system uses a higher mutation rate, which helps it to maintain its diversity [Paul, 1993]. Similarly, adopting a higher rate of mutation is expected to lead detectors to explore new non-self antigen niches and thus escape from existing self antigen niches. The following sections will study how this unusually high mutation rate affects the performance of the extended DynamiCS.

It should be noted that the hypermutation is applied to deleted memory detectors, not to existing memory detectors. This feature is a slight variation on the human immune system. The human immune system clones successful memory detectors and distributes them to other lymph nodes around the body. These new cloned detectors are expected to detect associative non-self antigens that share some non-self antigen patterns detected by previous detectors, but do not necessarily have the same non-self antigen patterns as the previous detectors. In other words, cloned detectors are expected to detect new antigens belonging to a new antigen cluster as soon as possible. During this process, the self-tolerance of new mutants is maintained by the helper T-cells (see Appendix A. Detailed Description of human immune systems). However, the extended DynamiCS does not have a separate helper T-detector population to confirm the self-tolerance of newly cloned detectors. Therefore, the extended DynamiCS uses hypermutation in a way to generate new detectors more tuned to target non-self antigen detection, and at the same time effectively avoiding self antigen detection. Memory detectors are deleted when they match self antigens of the current antigen cluster, but the fact that they managed to become memory detectors at all implies that they hold valid information about non-self antigens in previous clusters. By mutating these and reusing them in the form of a virtual gene library to seed new immature detectors, this evolved information is being retained and fine-tuned by the system.

### **7.6.2 Experimental Results**

Two series of experiments were performed in order to investigate the effects of hypermutation on TP and FP rates achieved by the extended DynamiCS. These experiments used the same parameters that were used in the experiments in the previous section, summarised in table 7.1.

The first series of experiments was performed by varying  $A$  values with mutation rate 0.1 and the second series was performed with mutation rate 0.2. Figures 7.8 and 7.9 show the average TP and FP rates of each series of experiments after five executions. The X-axes of these graphs represent

**Figure 7.8 TP and FP rates when  $A$  varies and  $T = 30, L = 10, N = 30$  with mutation rate = 0.1**

**Figure 7.9 TP and FP rates when  $A$  varies and  $T = 30, L = 10, N = 30$  with mutation rate = 0.2**

the number of generations and the Y-axes indicate detection rates. Each graph has two lines, one displaying the True Positive (TP) rate and another showing the False Positive (FP) rate. The grid lines on the X axis were placed at every  $N$  generations for  $N = 30$ . Each experiment was run for 2000 generations.

The effects of hypermutation are clearly revealed when these results are compared to the results obtained in the previous section (see figure 7.3). In figures 7.8 and 7.9, FP rates are consistently low except for one case where  $A = 5$  and mutation rate = 0.2. The differences in TP rates depending on different mutation rates are clearly noticeable when  $A$  has larger values. For instance, when  $A$  is 40 without mutants of memory detectors, TP rates range between 0.5 and 0.9 (see figure 7.3). On the other hand, when  $A$  is 40 with mutation rate 0.2, TP rates increase so that they range between 0.85 and 0.95 (see figure 7.9). More importantly, the improvement in TP rates was obtained without increase in FP rates. The scale of TP rate increase is much more noticeable when the mutation rate is 0.2 although the TP rate increase can also be seen when  $A$  is 40 with mutation rate 0.1 (see figure 7.8). Thus, it is verified that hypermutation affects the result of the extended DynamiCS in a positive way: TP rates increase while maintaining low FP rates.

The Extended DynamiCS with Mutation Rate = 0.1				
	Surviving Memory Detectors	Generated Memory Detectors	Deleted Memory Detectors	Memory Detector CoStimulation per generation
$A = 5$	45.5 (21.67)	535.5 (8869.67)	490 (8448.67)	40.48 (14.35)
$A = 10$	37 (4)	376 (1444.67)	339 (1456.67)	31.39 (1.43)
$A = 20$	32.5 (7)	259.5 (176.33)	227 (172)	28.08 (2.99)
$A = 40$	27.5 (24.5)	203.5 (14964.5)	176 (13778)	22.56 (6.66)

**Table 7.3 Average numbers of surviving, generated and deleted memory detectors during 2000 generations, and average number of memory detector costimulations per generation for the extended DynamiCS with mutation rate = 0.1. The mean values are followed by the variances in parentheses.**

The Extended DynamiCS with Mutation Rate = 0.2				
	Surviving Memory Detectors	Generated Memory Detectors	Deleted Memory Detectors	Memory Detector CoStimulation per generation
$A = 5$	44.75 (8.25)	264.5 (94.33)	219.75 (88.25)	39.15 (10.52)
$A = 10$	32.75 (24.92)	193.5 (539)	160.75 (393.58)	27.52 (14.62)
$A = 20$	29 (8.67)	126.5 (53.67)	97.5 (67)	24.48 (6.94)
$A = 40$	19.5 (0.33)	98 (1078)	78.5 (1013)	16.75 (1.12)

**Table 7.4 Average numbers of surviving, generated and deleted memory detectors during 2000 generations, and the average number of memory detector costimulations per generation for the extended DynamiCS with mutation rate = 0.2. The values in parentheses are variances.**

In addition, with  $A = 40$  and mutation rate 0.2, resulting in high TP and low FP rates, the extended DynamiCS still maintained the an average amount of memory detector costimulation per

generation as small as in table 7.2, when mutants of memory detectors were absent. This result can be seen in tables 7.3 and 7.4. These tables show the total number of surviving, generated and deleted memory detectors after two thousand generations when the mutation rate was 0.1 and 0.2 respectively. These numbers are the averages over five runs. In both cases, the extended DynamiCS has the smallest amount of memory detector costimulation when  $A = 40$ . Furthermore, when the extended DynamiCS has a larger mutation rate, 0.2, it performed a smaller amount of memory detector costimulation than when it had a mutation rate of 0.1.

To summarise, the two series of experimental results show that TP rates increased when immature detectors were generated by applying a hypermutation operator to deleted memory detectors. Furthermore, low FP rates and the small amounts of memory detector costimulation were maintained. These positive effects were more clearly demonstrated when a larger mutation rate was applied.

### 7.6.3 Experimental Analysis

The new experimental results presented in the previous section showed that this model of gene library evolution using hypermutation resulted in positive effects for intrusion detection purposes. Then, how were these positive effects obtained? To answer this question, it is necessary to conduct a more careful analysis of how TP rates and FP rates were altered, taking account of the different antigen clusters that were selected to present antigens.

The common feature of the graphs in figures 7.3, 7.8, and 7.9 is that TP rates rapidly increased until the first antigen cluster, from which antigens were selected, was replaced by the second antigen cluster. For instance, during the first 30 generations when antigens were selected from the first antigen cluster, TP rates increased to between 0.5 and 0.6. However, at generation 31, when the second antigen cluster was used to present antigens, TP rates dropped sharply. This sudden drop in TP rates was because detectors were introduced from the first antigen cluster, and these were initial mature detectors that had been tolerated only against self antigens from the first antigen cluster. Many memory detectors at generation 31 could mistakenly detect self antigens chosen from the second antigen cluster, and thus many memory detectors were deleted at generation 31. After the deletion of these memory detectors, the surviving memory detectors at that generation were not sufficient enough to cover the various non-self antigen niches. This caused a large drop in the TP rate.

After generation 31, TP rates increased very slowly for another 30 generations, which are the generations that the second antigen cluster was chosen to present antigens. Mature detectors and memory detectors generated from generation 32 onwards begin to have self tolerance against self antigens selected from the second cluster. For instance, mature detectors and memory detectors



that were converted from immature detectors at generation 32 were tolerated mostly against self antigens selected from the first cluster for 29 generations. However, they also had a chance to be tolerated against self antigens selected from the second cluster for 1 generation. Therefore, as the number of generations increased, new mature detectors and memory detectors had more tolerance against the second antigen cluster. This slowly decreased the number of detectors being deleted and resulted in gradual increase of TP rates from generation 30 to generation 60. When the third antigen cluster was chosen at generation 61, TP rates drop a little but they increase sharply for the next 30 generations. This can be explained by the same reasoning<sup>2</sup>. Thus, this cycle, in which TP rates dropped when a new antigen cluster was introduced and then started to increase until another new antigen cluster was presented, repeated over a total of 2000 generations.

Apart from the results common to figures 7.3, 7.8 and 7.9 as discussed above, it should also be noted that there are some differences between these three figures. Though the repeated cycles of TP rates are common, the *degrees* of TP rate drop, when the first antigen cluster was replaced by the second antigen cluster, are quite different. The degrees of TP rate drop shown in figure 7.3 are much larger than those seen in figure 7.8 and 7.9, when hypermutations were applied to generate immature detectors. In these cases, the new immature detectors were mutants of deleted memory detectors in the virtual gene library, and deleted memory detectors failed to avoid detecting self antigens selected from a new antigen cluster. When these immature detectors become mature detectors, by passing negative selection against the new antigen cluster, the mutations that have occurred on these immature detectors must allow them to survive negative selection against the new self antigen cluster. In addition, the parents of these immature detectors, which were deleted memory detectors, were previously successful at detecting a sufficient number of non-self antigens. Therefore, these immature detectors are more likely to preserve non-self antigen patterns that were learnt by the deleted memory detectors in the virtual gene library.

Let us see why these immature detectors could retain the memory of non-self antigen patterns learnt by the deleted memory detectors. Since the mutation operator affected randomly selected genes, it could be expected that mutants of deleted memory detectors could partially preserve information that was learnt by the deleted memory detectors, regardless of whether this was information about self or non-self antigens. Similarly they also partially lost other information as a result of mutation, regardless of whether the information was about self and non-self antigens. However, this random loss of information was influenced by negative selection. Immature

---

<sup>2</sup> The difference between the scales of TP rate increase from generation 31 to generation 60 and from generation 61 and generation 90 could originate from the different nature of the two antigen clusters. The second antigen cluster that was used from generation 31 to generation 60 could consist of antigen niches that were highly scattered. In this case it would have been harder to detect all the niches with a small number of detectors. In contrast, the third antigen cluster that was used from generation 61 to generation 90 could have consisted of antigen niches that were more clustered. Similarly, it would have been easier to detect them with a small number of detectors. However, both cases showed TP rate increases during these periods, and we pay attention only to this symptom.

detectors that have not bound to self antigens will pass negative selection and they are inclined to preserve partial information about non-self antigen information. In contrast, immature detectors that lost non-self antigen information but preserved self-information would be deleted by negative selection. Therefore, it is expected that mature detectors, after passing negative selection, were likely to have some non-self antigen information. Thus, the TP rate dropped slightly when antigens were presented from the same antigen cluster.

In addition, it was observed that these types of positive effects were more likely to occur when a larger mutation rate was given. In tables 7.3 and 7.4, the third column record how many memory detectors were deleted over the full two thousand generations. These numbers are much larger when the smaller mutation rate, 0.1, was applied than when the mutation rate 0.2 was applied. When the larger mutation rate was applied, the extended DynamiCS not only maintained a smaller amount of memory detector costimulation, but also operated in a more economical way by killing a smaller number of memory detectors. These results also can be compared to those in table 7.2, when all the immature detectors were randomly generated. In this case, the number of memory detectors deleted is smaller than when the mutation rate 0.2 was applied. This implies that newly generated memory detectors were quite different from previously generated memory detectors, and thus the new memory detectors more easily avoided matching self antigens that were matched by previously deleted detectors. However, at the same time, these new detectors were more likely to lose information about non-self antigens matched by previously deleted detectors. This caused the smaller number of deleted memory detectors and the lower TP rates when immature detectors were generated randomly without the virtual gene library.

## **7.7 Discussion of DynamiCS**

As has been demonstrated in this chapter and the previous one, DynamiCS has been introduced as an AIS that fulfils two properties required by IDS: learning of stabilised self behaviours with only a small subset of self antigens at any one time, and learning of sudden changes in converged self behaviours. In order to provide these features to the AIS, DynamiCS employed several novel components such as immature, mature and memory detector populations, a tolerisation period, an activation threshold, a mature detector life-span, mature and memory detector costimulation, and applying hypermutation to generate immature detectors. All of these novel components were inspired by mechanisms existing in the human immune system and led the AIS to yield the two desired properties.

As introduced in chapter six, many of these novel components are based on a different AIS, called LYSIS, proposed by [Hofmeyr, 1999; Hofmeyr and Forrest, 2000]. LYSIS is also equipped with three detector populations (immature, mature and memory), a tolerisation period, an activation threshold, costimulation and a mature detector life-span. [Hofmeyr, 1999; Hofmeyr and

Forrest, 2000] tested the LYSIS system against network traffic headers collected for 50 days, consisting of 3900 unique self strings. In order to deal with this quantity of self strings efficiently, Hofmeyr and Forrest [Hofmeyr, 1999; Hofmeyr and Forrest, 2000] developed LYSIS in a distributed environment, and thus fifty different hosts generated a total of 5000 immature detectors per day. Similarly to DynamiCS, LYSIS also dynamically generated immature detectors and started to monitor new antigens after the first tolerisation period. Although this system was tested against real network headers, the real environment scenario given to these tests was limited to the first real scenario studied in this thesis: learning of stabilised self behaviours with only a small subset of self antigens at any one time. DynamiCS is the only AIS that employs the novel components introduced in this thesis and has been tested on another important real IDS scenario: learning of sudden changes in converged self behaviours. In this scenario, DynamiCS was capable of detecting non-self antigens to a satisfactory level without losing its self-tolerance, and this was achieved by simulating gene library evolution using hypermutation which is not adopted by LYSIS.

Furthermore, [Hofmeyr, 1999; Hofmeyr and Forrest, 2000] investigated a way to tune LYSIS behaviours to obtain the desired TP and FP rates. This study was focused on choosing an appropriate tolerisation period, activation threshold and decay rate. It should be noted that the decay rate used in LYSIS was not adopted by DynamiCS. It was considered that the number of parameters used in DynamiCS was already large enough to make it difficult to control system behaviour. Although a decay rate was introduced in LYSIS in order to replace detectors more dynamically, DynamiCS managed to produce a similar effect without this parameter by using a gene library evolution model with hypermutation.

## 7.8 Summary

The experimental results in the previous chapter verified that DynamiCS could not learn new self-antigens when learned self and non-self behaviours suddenly alter due to legal self change. This resulted in high FP rates when new antigens were monitored by DynamiCS, although it produced high TP rates. The proposed explanation of this outcome was that the generated memory detectors had never been exposed to a certain antigen cluster within their tolerisation periods. Thus they could not have a sufficient degree of tolerance against a complete self set. To tackle this problem, the current chapter investigated an extension of DynamiCS, so that it can reduce the FP rates increased by memory detectors.

As one way to decrease the FP rates caused by memory detectors, DynamiCS was extended by eliminating memory detectors when they showed a poor degree of self-tolerance to new antigens. This extended system was tested to determine whether surviving memory detectors no longer cause seriously high FP error rates. The test results showed that deletion of memory detectors

based on their self-antigen detection dramatically decreased the FP rates that were observed in the previous chapter. However, this method required a larger amount of costimulation in order to gain such benefits. The large amount of costimulation can render the system weak for intrusion detection. This disadvantage demanded a further extension of DynamiCS.

In order to resolve this problem, DynamiCS employed the use of hypermutation in DynamiCS to produce the effect of gene library evolution. This additional extension was designed to fine-tune generated memory detectors so that the system obtained higher TP rates without increasing the amount of co-stimulation. The gene library evolution was modelled by producing immature detectors via hypermutation on deleted memory detectors. Thus a “virtual gene library”, made from mutations of deleted memory detectors was maintained. The new extension was tested to determine whether it gains high TP rates without increasing the amount of costimulation as the result of gene library evolution. The test results proved that hypermutation led the progress of gene library evolution and thus produced immature detectors that are more tuned to cover existing non-self antigens. These were understood by analysing different *degrees* of TP rate drop when gene library evolution was and was not applied. With gene library evolution, DynamiCS showed a smaller TP rate drop when antigens were presented from the same antigen cluster. This is because the mutants of previously deleted memory detectors, having survived the negative selection stage, are likely to have some non-self antigen information without patterns matching self antigens. Finally, the novel features of DynamiCS studied in this thesis are discussed in the context of comparison with the most similar AIS, LYSIS, developed by [Hofmeyr, 1999; Hofmeyr and Forest, 2000].