Recognising Promoter Sequences Using An Artificial Immune System

Denise E. Cooke and **John E. Hunt,**
Centre for Intelligent Systems, Department of Computer Science,
University of Wales, Aberystwyth, Dyfed, U.K. SY23 3DB
Email: {dzc,jjh}@uk.ac.aber

Abstract

We have developed an artificial immune system (AIS) which is based on the human immune system. The AIS possesses an adaptive learning mechanism which enables antibodies to emerge which can be used for classification tasks. In this paper, we describe how the AIS has been used to evolve antibodies which can classify promoter containing and promoter negative DNA sequences. The DNA sequences used for teaching were 57 nucleotides in length and contained procaryotic promoters. The system classified previously unseen DNA sequences with an accuracy of approximately 90%.

1. Introduction

Most of the research in intelligent systems and molecular biology has been motivated by the needs of those molecular biologists working on genome mapping projects. These projects are producing a huge amount of DNA sequence data which requires computational techniques for its easy and speedy analysis.

Identifying protein coding regions (genes), and their control elements, in DNA sequences is a complex task for a number of reasons:

- Non-coding regions. Only approximately 5% of the human genome encodes proteins.
- Control sequences. Recognising these sequences is difficult because they can be sequentially remote from the coding region. The largest documented distance is 60,000 nucleotides in the human genome.
- Introns. These are non-coding regions in eucaryotic DNA which are spliced out of the transcribed DNA before being translated into protein. Recognition of coding regions and non-coding regions of DNA is an important problem.
- Redundancy. There is considerable variability in the use of alternative redundant codes among different species. Redundancy refers to the fact that of the twenty amino acids, most are encoded by more than one nucleotide triplet. The most commonly found triplet for a particular amino acid tends to be species specific. This makes it more difficult to compare DNA sequences derived from different species.
- Complexity. A stretch of eucaryotic DNA can code for several genes, possibly using overlapping reading frames, going in opposite directions, and interrupted by different introns (which can cause shifts of reading frames within a single protein).

However, procaryotic genomes are not as complex or as large as eucaryotic genomes and hence are easier to analyse, e.g. procaryotic genes do not contain introns, and have promoter sequences which are located upstream, in close proximity.

A number of artificial intelligence techniques have been applied to the analysis of DNA sequence data and are reviewed by (Hunter 1991; Rawlings & Fox 1994). They include expert systems, model-driven vision, grammar induction, formal inference, minimal length encoding, case-based reasoning, and knowledge-based artificial neural networks.

Our approach was to develop an artificial immune system (AIS), based on a model of the human immune system. The immune system is a remarkable defence mechanism protecting our bodies from invasion by foreign substances. It is essential to our very survival, yet has not attracted the same kind of interest from the computing field as the neural operation of the brain or the behavioural actions of lower organisms. However, the immune system is a rich source of theories and as such can act as an inspiration for computer based solutions. For example, it is an adaptive learning system with a content addressable memory. Indeed, this memory is self organising, is dynamically maintained and allows items of information to be forgotten. This is all the more noteworthy as the immune system is a distributed system with no central controller.

In this paper, we describe the human immune system and how the AIS can be used in a pattern recognition task, specifically to learn to recognise procaryotic promoters in DNA sequences. To do this the AIS creates antibodies to promoter containing DNA sequences. This is similar in nature to immunisation. The antibodies which are created can then be used to determine whether new sequences are promoter containing

or promoter negative. We then discusses the results we have obtained and consider related work and future work.

2. The Immune System

The *immune system* protects our bodies from attack from foreign substances (called antigens) which enter the bloodstream. The immune system does this using antibodies which are proteins produced by the white blood cells called *B cells* (or B-lymphocytes). The B cells (which originate in the bone marrow) collectively form what is known as the *immune network*. This network acts to ensure that once useful B cells are generated, they remain in the immune system until they are no longer required.

When a B cell encounters an antigen an *immune response* is elicited, which causes the antibody to bind the antigen (if they match) so that the antigen can be neutralised. If the antibody matches the antigen sufficiently well, its B cell becomes *stimulated* and can produce mutated clones which are incorporated into the network. The stimulation level of the B cell also depends on its affinity with its neighbours in the network of B cells.

Five percent of the least stimulated B cells die daily and are replaced by an equal number of completely new B cells generated by the bone marrow, thus maintaining *diversity* in the immune system. The new B cells are only added to the immune network if they possess an affinity to the cells already in it, otherwise they die.

2.1 The primary and secondary responses of the immune system

The immune system possesses two types of response: primary and secondary. The primary response is the initial process by which the immune system destroys the antigen by generating antibodies. The secondary response occurs when the immune system encounters the same antigen again. This response is characterised by a quicker destruction of the infectious agent.

It is clear from the faster secondary response time, that the initial contact with the antigen allows the immune system to adapt to (to learn) that antigen which thus allows the immune system to deal with that antigen with much less effort in the future. The immune system thus possesses a form of memory, and it is a form of *content addressable* memory since the secondary response can be elicited from an antigen which is similar, although not identical, to the original one which established the memory (this is known *as cross-reactivity*).

2.2 The immune network

There are two views on how memory works in the immune system. The most widely held view uses the concept of "virgin" B cells being stimulated by antigen and producing memory cells and effector cells. A theory less accepted by experimental immunologists, but held by some theoretical immunologists, uses the concept of an immune network (reviewed by Perelson 1989). We are using this theory as the inspiration for our learning system, rather than trying to accurately model the biological behaviour of the immune system. The theory states that the network dynamically maintains the memory using feedback mechanisms within the network. Thus if something has been learned, it can be forgotten unless it is reinforced by other members of the network. The immune network is discussed in more detail later.

From the point of view of a self organising learning system, the concept of an immune network has a number of interesting (and potentially extremely useful) capabilities. For example, if something has been learnt by the immune network, it can be forgotten unless it is reinforced by other members of the network or by antigens. If this analogy is applied to a learning system, it might be possible to "forget" information which is no longer being used. Obviously care must be taken with such a process as the information may become useful in the future, depending upon the application.

2.3 Antibody/antigen binding

Antibodies identify the antigens they can bind by performing a complementary pattern match (in a fashion much like a lock and key). The strength of the bind depends on how closely the two match. The closer the match between antibody and antigen the stronger the molecular binding and the better the recognition.

2.4 B cells

The antibodies reside on the surface of the B cells so that they are in a position to bind to any antigens they encounter while the B cells travel around the blood stream. All the antibodies associated with a single B cell will be identical, thus giving the B cell an "antigen specificity".

The huge variety of possible antibodies is a result of the way in which their heavy and light chain variable regions are each divided up into several distinct protein segments. Each segment is encoded by a library of genes which lie on the same chromosome, but are widely separated. A functional antibody gene is not constructed until the genes are randomly chosen from the various libraries and are folded into place. This combination of random gene selection and folding results in millions of

possibilities and thus allows the immune system to possess a wide range of antibody types.

2.5 B cell stimulation

When an antibody on the surface of a B cell binds an antigen, that B cell becomes stimulated. The level of B cell stimulation depends not only on how well it matches the antigen, but also how it matches other B cells in the immune system. If the stimulation level rises above a given threshold, the B cell becomes enlarged and starts replicating itself many thousands of times, producing clones of itself. To allow the immune system to be adaptive, the clones that grow also turn on a mutation mechanism that generates, at very high frequencies, point mutations in the genes that code specifically for the antibody molecule. This mechanism is called *somatic hypermutation* (Kepler & Perelson 1993). Alternatively, if the stimulation level falls below a given threshold, the B cell does not replicate and in time it will die off.

```
Load antigen population
Randomly initialise the B cell population
Until termination condition is met do
   Randomly select an antigen from the antigen population
   Randomly select a point in the B cell network to insert the antigen
   Select a percentage of the B cells local to the insertion point
   For each B cell selected
      present the antigen to each B cell and determine whether this antigen can be bound
       by the antibody and if so, how stimulated the B cell will become.
      If the B cell is stimulated enough, then clone the B cell
   If no B cell could bind the antigen,
         generate a new B cell which can bind the antigen
   Order these B cells by stimulation level
   Remove worst 5% of the B cell population
   Generate n new B cells (where n equals 25% of the population)
   Select m B cells to join the immune network (where m equals 5% of the population)
```

Figure 1: The Immune System Object algorithm

2.6 Influence of the network

As stated above, the stimulation level of the B cell also depends on its affinity with other B cells in the immune network (Perelson 1989). This network is formed by B cells recognising (possessing an affinity to) other B cells in the system. The network is self organising, since it determines the survival of newly created B cells as well as its own size (see De Boer & Perelson 1991). The more neighbours a B cell has an affinity with, the more stimulation it will receive from the network, and visa versa.

Survival of the new B cells (produced by the bone marrow, or by hypermutation) depends on their affinity to the antigen and to the other B cells in the network. The new B cells may have an improved match for the antigen and will thus proliferate and survive longer than existing B cells. The immune network reinforces the B cells which are useful and have proliferated. By repeating this process of mutation and selection a number of times, the immune system "learns" to produce better matches for the antigen.

3. The Artificial Immune System

The Artificial Immune System (AIS), is comprised of a central immune system node (which performs some of the functions of the bone marrow in the natural immune system), a network of B cells and an antigen population. Most of the processing of the AIS is encapsulated within the B cells and their antibodies (these will be discussed below).

The immune system node possesses a main algorithm which initiates the immune response by presenting antigen to the B cells. At the end of every iteration of the main loop of the algorithm, the immune system node also generates completely new (random) B cells which can be considered for inclusion into the immune network.

3.1 The immune system object

The first step in the algorithm, illustrated in Figure 1, is to load the antigen population, and then initialise the B cell population (described later). When the antibody and antigen populations are initialised, the main loop of the immune system is executed. This loop first selects an antigen randomly from the antigen population. It then selects a random point in the immune network. A percentage of the B cells within this neighbourhood are then selected to process the antigen. At present, 75% of the cells immediately surrounding the insertion point are considered, then 50% of the cells around those cells are

selected, then 25% of the neighbours of these cells are selected. At this point the "spreading" effect of the antigen terminates. This means that the antigen has an influence which spreads through the network, gradually decreasing in concentration as it goes.

When the antigen is presented to a B cell, an immune response is triggered (this is discussed in more detail in section 4). It is important to note that presenting an antigen to a B cell which can bind it, can result not only in a single B cell analysing the antigen, but also in the creation of many new B cells, all of which may in turn analyse the antigen and generate further B cells (note that this is one of the two ways in which new B cells are created).

If at this point no B cells have been able to bind the antigen, the system will explicitly generate a new B cell which will bind it. This is done by creating an antibody, using the antigen as a template, which is then inserted into the immune network next to B cells to which it has the most affinity. Although this particular B cell is specific to the antigen just processed, daughter B cells will include mutation which may be able to bind not only the antigen which was used to generate the B cell, but similar antigens.

Once the antigen has been processed by all the appropriate B cells, some of the B cells in the immune population are highlighted for deletion. This is done by identifying those B cells whose stimulation level is within the lowest 5% of the B cell population. These B cells are then deleted from the immune population (and the immune network). The immune system node, independently of the content of the immune network, generates a complete new set of B cells (this is the second way in which B cells are generated). From these "virgin" B cells a number corresponding to the number deleted are added to the immune network. The B cells for this are selected on the basis of their *affinity* to the cells already in the network. The main loop is then repeated until a pre-specified number of iterations have been completed.

3.2 The B cell model

The B cell possesses a library of genes, a description of the adult DNA sequence, a number of intermediate representations (e.g., the nuclear RNA and the mRNA), as well as the antibody. It also records the stimulation level of the B cell and maintains links to a parent B cell (if present) as well as any sister and daughter B cells.

A new antibody can be created in one of two ways. The first method attempts to mirror the gene selection, folding, transcription and translation steps which occurs in the B cells in the natural immune system. The library of genes are made available to the B cell and the gene selection routines are instantiated. The paratope of the antibody is created from the mature mRNA. To do this, the mRNA string is copied in a complementary manner, e.g. a T $\rightarrow$ A, an A $\rightarrow$ T, a G $\rightarrow$ C and a C $\rightarrow$ G (we realise that this does not accurately reflect actual mRNA translation which involves using the nucleotide triplet code to identify the correct amino acids to incorporate into protein).

The second method uses the string defined by an antigen as the blueprint for the mRNA string. This is then copied in a complementary manner as before. The first method is used when the immune system object generates new B cells at the end of each iteration. The second method is used to initialise the B cell population and in any situation where no B cell could bind the current antigen.

3.3 The immune network

The immune system memory actually forms a network which maintains the current set of B cells which have been produced and links between these B cells (which represent affinities between the B cells). An example of a B cell network memory is illustrated in Figure 2.

In the immune-based memory, a newly created B cell is inserted near to those B cells which it has an affinity. The way in which the B cell is absorbed into the network is accomplished by first finding the two B cells with which the new B cell has the highest affinity. It is then linked to these two B cells and to any other B cells associated with them. Over time this enables the emergence of regions within the network which contain B cells which can deal with similar problems. Between these regions, bridges exist which indicate common characteristics between different problem areas.

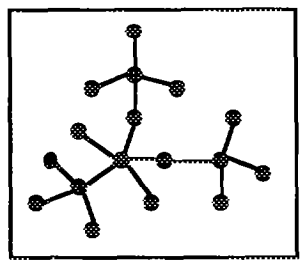

Figure 2: Schematic representation of the structure of the immune network

3.4 The antibody model

In the AIS, the antibody possesses a paratope which represents the pattern it will use to match the antigen. We have chosen a representation which uses A, T, G, C, and the wild card X. This wild card matches all of the nucleotides and was found to be necessary. For example,

we might want to identify a common antibody for the following antigens:

TATAATGCCGTATA
TATAATCGGCTATA
TATAATGATCTATA

With the wild card, we could generate a B cell with

TATAATXXXXTATA

as its antibody. Without it, we could never generate a B cell which could bind all three. This is important as we are relying on identifying common features amongst the antigens to enable us to determine whether new DNA sequences are promoter positive or negative.

3.5 The antigen model

To teach the AIS, we use the DNA sequences used by (Towell, Shavlik & Noordewier 1990). These sequences are 57 nucleotides in length. Some contain promoters of known *Escherichia coli* (procaryotic) genes, while others are promoter-negative. The promoter containing sequences are aligned so that the transcription initiation site is 7 nucleotides from the right of the sequence, i.e. the sequence extends from 50 nucleotides upstream of the protein coding region to 7 nucleotides into the protein coding region (referred to as -50 to +7). Some examples are shown below in Figure 3.

4. The Behaviour of the Artificial Immune System

4.1 Mimicking antibody/antigen binding

The immune response involves calculating how closely the antibody matches the antigen which results in a *match score*. If this match score is above a certain threshold the antibody may *bind* the antigen. When the binding strength has been calculated, the B cell can determine its stimulation level. Each of these steps is considered in more detail below.

The match algorithm counts each element which matches (in a complementary fashion) between the antigen and the antibody. In addition, the match algorithm is also weighted in favour of continuous match regions. If a continuous region of 4 elements matches, then such a region will have a value of 2 to the power of 4 (i.e. 2^4).

Figure 4 illustrates the result of an antibody being "matched" with an antigen. As can be seen from the figure, the number of elements which match is 12. However, this number must be added to the value of each of the match regions (e.g., the 6 elements which match at the front of the pattern). This means that the final match score for this example is 98.

```
+,S10,      tactagcaatacgcttgcgttcggtggttaagtatgtataatgcgcgggcttgtcgt
+,AMPC,     tgctatcctgacagttgtcacgctgattggtgtcgttacaatctaacgcatcgccaa
+,AROH,     gtactagagaactagtgcattagcttatttttttgttatcatgctaaccacccggcg
+,DEOP2,    aattgtgatgtgtatcgaagtgtgttgcggagtagatgttagaatactaacaaactc

-,1218,     ttcgtctccgcgactacgatgagatgcctgagtgcttccgttactggattgtcacca
-, 668,     catgtcagcctcgacaacttgcataaatgctttcttgtagacgtgccctacgcgctt
-, 413,     aggaggaactacgcaaggttggaacatcggagagatgccagccagcgcacctgcacg
-, 991,     tctcaacaagattaaccgacagattcaatctcgtggatggacgttcaacattgagga
```

Figure 3: Examples of promoter containing (+) and promoter negative (-) sequences.

The antigen population is made up of 52 of the 53 positive examples (for "leave one out" testing). Thus each antigen is composed of A, G, C and T (note no wild cards are present in the antigen).

The antigen model used in our Artificial Immune System is very simple; each potential antigen is represented by an antigen object. This object possesses a single epitope (a string representing the antigen) which antibodies will attempt to match. The antigens are defined in an external ASCII file and are loaded into the AIS by the antigen population object. This object reads a series of lists from a data file and instantiates those lists as antigen objects.

The binding value, derived from the match score, represents how well two molecules bind. For an antibody to bind an antigen, the binding must be stable, that is the match score must exceed a certain threshold before the binding takes place. This threshold is determined by the AIS itself. It does this by adjusting the threshold relative to the current average binding value, either increasing or decreasing it by half the difference between the current threshold and the average threshold. This approach is a variation on the matching algorithm used by (Hightower, Forrest & Perelson 1993) which we have extended to include a wild card 'X' for reasons which are discussed in section 3.4. However, it was important that we did not just generate a set of B cells whose antibodies were all wild cards (these antibodies could theoretically bind to

anything). We therefore gave a "wild card match" a value of 1 compared to a value of 2 for a "full match" which appeared to work well.

Antigen:	c	g	c	t	t	g	c	g	t	t	c	g	g	t	g	
Antibody:	g	c	x	x	a	c	a	c	a	c	g	c	t	a	c	
Evaluation:	2	2	1	1	2	2	0	2	2	0	2	2	0	2	2	=> 22
Length:						6				2			2			2
Match value:	$22 + 2^6 + 2^2 + 2^2 + 2^2 =>$ 98															

Figure 4: Calculating a match value

4.2 Simulating B cell stimulation

In the immune network theory the level to which a B cell is stimulated relates to how well its antibody binds to the antigen and to other B cells. Thus in the AIS, a B cell is stimulated by an antigen which is complementary to it and by B cells which are similar to it, but it is repressed by B cells which are complementary to it. The algorithm for calculating the stimulation level is that of (Farmer, Packard, & Perelson 1986.). If the resulting stimulation level is above a certain threshold then the B cell generates clones of itself. These clones are added to the immune network and turn on a hypermutation operator (see next section).

4.3 Modelling somatic hypermutation

When a new B cell is created somatic hypermutation is applied to it (we follow the approach used by (Farmer, Packard, & Perelson 1986) to hypermutation). The AIS uses three types of mutation: multi-point mutation, substring regeneration and simple substitution. The actual form of mutation applied is chosen randomly.

In multi-point mutation each element in the antibody is processed in turn. If a randomly generated number is above the mutation threshold, then the element is mutated. This means that a value of A, T, G, C or X is randomly generated to replace the original element. In substring regeneration, two points are selected at random in the antibody's paratope. Then all the elements between these two points are replaced by one of the values A, T, G, C or X which is chosen randomly. The simple substitution operator uses the roulette wheel algorithm (Goldberg 1989) to select another B cell to use as a source of new elements for the current B cell. The operator then either selects an element from the original antibody's paratope or an element from the retrieved B cell's antibody.

Whatever mutation operator is applied, the new "mutated" B cells are added to the immune network if they can bind the antigen present or if an affinity can be found for them somewhere within the network.

4.4 Maintaining diversity

The lowest 5% of the B cell population are killed off at the end of each iteration of the main algorithm. They are replaced by an equal number of new B cells which are absorbed into the immune network. This results in a network within which the number and nature of the B cell population is continuously changing. This helps to promote the diversity of the B cells in the system.

5. Promoter recognition

Control regions (e.g., promoters) in the DNA are responsible for the timing and level of gene expression. Identifying these control sequences in the DNA aids the identification of gene sequences. Promoters are regions of DNA to which RNA polymerase (the enzyme which initiates transcription) binds. Procaryotic promoters are easier to recognise than eucaryotic promoters. They contain a sequence known as the Pribnow sequence or TATA box which most commonly contains the nucleotides TATAAT.

5.1 Running the system

We used only the positive examples from the Towell DNA data set to teach the AIS (similar to immunisation, i.e., the primary immune response). For all tests we used a "leave one out" method. That is, we presented the AIS with 52 of the 53 sequences so that we could test the AIS not only on the 53 negative examples that it had never seen but also on a positive example it had never seen. Note that each of the positive examples was left out in turn so that we could obtain reliable results.

The main loop of the AIS was run for 52 iterations. This meant each positive example had the chance of being presented to the AIS twice. However, as we selected antigens at random, there was no guarantee that a particular sequence would be presented to the AIS during this loop.

When the total number of iterations is complete the generated antibodies are saved into an ASCII file. This is used by a shorter version of the algorithm, known as a "run time" environment, from which all the learning elements have been removed, leaving only the pattern matching elements (similar to the secondary immune response). The generated antibodies were compared to each of the sequences (positive and negative) in the

Towell data set. In this case, all the antibodies available in the system were given the chance to "bind" the input sequences. If one or more of the antibodies could "bind" the sequence then that sequence was deemed to be a promoter containing sequence. If no antibodies could bind the sequence than it was considered to be a promoter negative sequence.

5.2 Testing the generated antibodies

As the AIS is stochastic in nature, the different runs performed with the system resulted in different results. We thus obtained a range of error margins between 8 and 12%. On average the AIS generated a set of antibodies which could correctly classify 90% of the sequences presented to it. By analysing the errors in classification it was found that about 3% of the errors resulted in positive examples being classified as negative. This is not surprising because the AIS generalises from the positive examples.

6. Discussion of Results

The results we obtained remained relatively constant even when the number of antibodies or the number of iterations (and hence the number of antigens presented to the AIS) was varied. It may not be possible to obtain greater than 90% correct classification of DNA sequences using the information (patterns) available within this set of DNA sequences alone. This appears to be backed up by the work of (Towell, Shavlik & Noordewier 1990) who have reported the results in Table 1. The systems presented in this table have all been tested on the same data set as the AIS. As can be seen in this table, the average 10% error margin of the AIS fits in with the results obtained from the back propagation and kNN methods. Both of these methods rely on the content of the sequences on which they are taught to enable them to correctly classify unseen sequences. In contrast, the KBANN system uses additional knowledge to help in this classification.

A contrast between the systems presented in Table 1 and the AIS is that the AIS is only taught using the positive examples. That is, it does not need to see a series of negative examples in order to be able to correctly classify approximately 90% of the sequences. We believe that this is a strength of the system as it is not necessary to choose and optimise a negative training set, saving time and not affecting the accuracy of performance by, for example, accidentally introducing true examples in the negative data set (Horton & Kanehisa 1992).

Table 1: Published data on Towell data set

System	*Error rate*	*Method*
KBANN	5.3%	Hybrid KB and NN
Back propagation	9.2%	Standard back propagation with hidden layer
ID3	19%	Quinlan's Decision Tree Builder
kNN	12.3%	Nearest neighbour algorithm (k=3)

To test how robust the performance of the system was, we tested the system by teaching it using only 25 of the 53 positive examples. We then presented all 106 examples to the B cells which were generated. We found that although the performance of the system did degrade we still managed to correctly classify about 86% of the sequences.

A notable feature of the antibodies generated by the AIS is that it is possible to examine the patterns it has learnt. This means that it is easy to determine what the AIS has determined is significant in the DNA sequences it has seen. This is a significant advantage over approaches such as neural networks and nearest neighbour algorithms.

It must be noted, however, that any machine learning approach can only be as good as the data with which it has been taught. Thus if there is any bias in the example set, then that bias will also be present in its performance.

7. Related work

7.1 Approaches to promoter recognition

(Towell, Shavlik & Noordewier 1990) derived a rule set to identify the -10 and -35 sequence patterns. This promoter rule set was used to initialise a neural network's topology and weights. The authors found that the networks initialised by the KBANN algorithm (knowledge-based artificial neural networks) generalised better than conventional neural networks, decision trees and nearest-neighbour classifiers, and that the promoter rules assisted learning.

Researchers using different neural network approaches to promoter recognition have achieved different levels of prediction accuracy, ranging from 75% (Nakata, Kanehisa & Maizel 1988) to 98% (Demeler & Zhou 1991) (reviewed by Hirst & Sternberg 1992). However, (Horton & Kanehisa 1992) found that the

differences in reported prediction accuracy could be explained in terms of the information content of the data sets used. It is often the case with neural networks that the test set contains sequences that are quite similar to the training sequences.

All these neural network approaches use a false (negative) data set for training. The accuracy of the neural network is affected by this data set. For example, (Demeler & Zhou 1991) found that the number of random sequences in the negative data set had a significant effect on accuracy. Our AIS has the advantage that it only needs to be presented with true (positive) data. This therefore omits the need to choose and optimise a negative training set, saving time and not affecting the accuracy of performance. For example, (Horton & Kanehisa 1992) used coding regions at random as their negative data and recognised that there was a potential problem with this because promoters are sometimes located in the coding regions. An additional problem relates to the possibility that their neural network might have a poor ability to recognise, as promoter-negative sequences, non-coding regions which did not contain a promoter.

7.2 Immune-based learning

The concept of an immune network to create a content-addressable auto-associative memory has been used by (Gilbert & Routen 1994). They have applied this system to the recording and recognition of 64 * 64 black and white pictures. Their system views the immune system as essentially a connectionist device in which localised nodes (B cells) interact to learn new concepts or to recognise past situations. This differs greatly from our approach as they highlight themselves when they state that they "are not interested in representing cells and antibodies .... but only representing those aspects relevant from the point of view of their interactions .... only their combining regions". In contrast, in this paper, we are not only interested in representing cells and antibodies but also the genetic mechanisms by which the antibodies are formed.

(Forrest et al. 1993) use a genetic algorithm (GA) to model the evolution and operation of the immune system. This is the opposite of the approach we have taken in this paper, that is, they are using a GA to model the immune system whereas we are explicitly modelling the immune system within a computer program. Another way in which our AIS differs from the work of (Forrest et al. 1993) is that, although a bit string representation is used for both the genes that code for the antibody as well as the antibody itself, we take into account the gene selection and folding, transcription and translation steps in antibody generation. This means that we promote the diversity of the population by acknowledging the genetic aspects of antibody generation without introducing the addition of a separate representation scheme. In contrast, (Forrest et al. 1993) by-pass the genetics, making antibodies directly by generating random strings and apply "standard" mutation and crossover operators.

Finally, (Bersini 1991) has explored further the use of the immune network as a evolutionary learning mechanism. His work concentrates almost solely on the immune network and ignores the genetic operation of the immune system. However, this work may offer ways of enhancing the operation of the network within the AIS.

7.3 Discussion

It is useful to consider where the AIS actually fits within the spectrum of machine learning systems. Within this field the AIS can be categorised as an example of a system which learns in an unsupervised manner (i.e. it does not require an external critic or fitness function to assess its progress unlike some neural network (Lippmann 1987) and learning classifier systems (Goldberg 1989), it is noise tolerant (due to the emergence of generalist antibodies) in contrast to some induction approaches such as ID3 (Quinlan 1993), it is a symbolic learning system (as opposed to the subsymbolic approach of neural networks) and it learns in an incremental manner (i.e. examples are presented to it individually rather than in bulk).

It is interesting to note that (Hoffmann 1986) has compared the immune system, immune network and immune response to neural networks, while (Farmer, Packard, & Perelson 1986) and (Bersini & Varela 1990) have compared them with learning classifier systems. In contrast, we have attempted to mirror the information processing abilities of the immune system explicitly.

8. Future work

At present the AIS fails to achieve the same level of accuracy as the KBANN system. We believe that a hybrid approach (such as that used in KBANN) could increase the accuracy. There are a number of possibilities which we are considering; we could pre-process the information in the sequence before it is analysed by the AIS to identify any "probable known sequences" or we could post-process the results of the AIS to see if additional knowledge can be used to confirm or question the AIS classification. Another approach is to take advantage of any additional knowledge available, such as known sequences (e.g. TATAAT), when initialising a B cell's antibody. We are also considering merging the AIS with a case-based reasoning (CBR) approach in which the CBR system attempts to confirm the classification of the AIS.

9. Conclusions

This paper has shown that a learning mechanism based on analogies with the immune system can be used to generate classifiers for DNA sequences. These classifiers can indicate, in approximately 90% of cases, for the data we have used whether a DNA sequence contains a promoter or not. As these classifiers are generated using only positive examples (and it is therefore not necessary to employ negative examples) the AIS is a useful addition to the range of techniques used in promoter recognition learning systems. In addition, as the AIS evolves antibodies, it is easy to determine what patterns have emerged. That is, it is possible to identify what the AIS has determined is significant in the DNA sequences it has seen. This is a significant advantage over approaches such as neural networks. However, to improve the accuracy of the system significantly above the 90% level, we recognise the need to exploit additional domain knowledge.

References

Bersini, H. 1991. Immune network and adaptive control, *Proceedings of the First European Conference on Artificial Life*, ed. by F. J. Varela and P. Bourgine, Pub. MIT Press.

Bersini, H.; and Varela, F. 1990. Hints for adaptive problem solving gleaned from immune networks, *Proceedings of the First Conference on Parallel Problem Solving from Nature*, 343-354.

De Boer, R. J.; and Perelson, A. S. 1991. How Diverse Should the Immune System Be? *Proceedings of the Royal Society of London B* 252:171-175.

Demeler, B.; and Zhou, G. 1991. Neural Network Optimization for *E. coli* Promoter Prediction. *Nucleic Acids Research* 19: 1593-1599.

Farmer, J. D.; Packard, N. H.; and Perelson, A. S. 1986. The Immune System, Adaptation and Machine Learning. *Physica* 22D:187-204.

Forrest, S.; Javornik, B.; Smith, R. E. and Perelson, A. S. 1993. Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. *Evolutionary Computation* 1(3):191-211.

Gilbert, C. J.; and Routen, T. W. 1994. Associative Memory in an Immune-Based System. In Proceedings of AAAI'94, 2:852-857.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley.

Hightower, R.; Forrest, S.; and Perelson, A. S. 1993. The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation. Department of Computer Science, University of New Mexico, Albuquerque, USA.

Hirst, J. D.; & Sternberg, M. J. E. 1992. Prediction of Structural and Functional Features of Protein and Nucleic Acid Sequences by Artificial Neural Networks. *Biochemistry* 31: 7211-7218.

Hoffmann, G. W. 1986. A neural network model based on the analogy with the immune system. *Journal of Theoretical Biology* 122:33-67.

Horton, P. B.; and Kanehisa, M. 1992. An assessment of Neural Network and Statistical Approaches for Prediction of *E. coli* Promoter Sites. *Nucleic Acids Research* 20:4331-4338.

Hunter, L. 1991. Artificial Intelligence and Molecular Biology. *AI Magazine* 11:27-36.

Kepler, T. B.; and Perelson, A. S. 1993. Somatic Hypermutation in B cells: An Optimal Control Treatment. *Journal of Theoretical Biology* 164:37-64.

Lippmann, R. P. 1987. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine* April, 4-22.

Nakata, K.; Kanehisa, M.; and Maizel, J.V. 1988. Discriminant Analysis of Promoter Regions in *Escherichia coli* Sequences. *Computer Applications in the Biosciences* 4:367-371.

Perelson, A. S. 1989. Immune Network Theory. *Immunological Review* 110:5-36.

Rawlings, C. J.; and Fox, J. P. 1994. Artificial Intelligence in Molecular Biology: a Review and Assessment. *Philosophical Transactions of the Royal Society of London. Series B. Biological Sciences* 344:353-363.

Quinlan, J. R. 1993. *C4.5 Programs for Machine Learning*, Pub. Morgan Kaufmann CA.

Towell, G.; Shavlik, J.; and Noordewier, M. 1990. Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks. In Proceedings of the Eighth National Conference on Artificial Intelligence, 861-866. AAAI Press, Menlo Park, California.