An Artificial Immune System for Evolving Amino Acid Clusters Tailored to Protein Function Prediction

A. Secker[1], M.N. Davies[2], A.A. Freitas[1], J. Timmis[3], E. Clark[3], D.R. Flower[2]

[1] Computing Laboratory and Centre for BioMedical Informatics, University of Kent,
Canterbury, CT2 7NF, UK
[2] The Jenner Institute, University of Oxford, Compton, Newbury, Berkshire, RG20 7NN, UK
[3] Departments of Computer Science and Electronics, University of York,
York, YO10 5DD, UK
a.d.secker@kent.ac.uk, m.davies@mail.cryst.bbk.ac.uk, a.a.freitas@kent.ac.uk,
jtimmis@cs.york.ac.uk, edclark@cs.york.ac.uk, darren.flower@jenner.ac.uk

Abstract. This paper addresses the classification task of data mining (a form of supervised learning) in the context of an important bioinformatics problem, namely the prediction of protein functions. This problem is cast as a hierarchical classification problem, where the protein functions to be predicted correspond to classes that are arranged in a hierarchical structure, in the form of a class tree. The main contribution of this paper is to propose a new Artificial Immune System that creates a new representation for proteins, in order to maximize the predictive accuracy of a hierarchical classification algorithm applied to the corresponding protein function prediction problem.

Keywords: artificial immune systems, data mining, bioinformatics, classification, clustering.

1 Introduction

This paper addresses classification within data mining in the context of bioinformatics, more precisely the prediction of protein function. In essence, a protein consists of a linear sequence of amino acids, and predicting the function of a protein, based on information derived from its sequence of amino acids, remains an important problem in bioinformatics.

The main contribution of this paper is to propose a new Artificial Immune System (AIS) – a variant of opt-aiNet (a well-known AIS) – that creates a new representation for proteins, in order to maximize the predictive accuracy of a classification algorithm applied to the corresponding protein function prediction problem.

In order to understand the task to be solved by the proposed AIS, it should first be noted that the type of attribute representation addressed in this paper involves local descriptors of amino acid sequences (Zhang et al., 2005; Cui et al., 2007). In developing the local descriptors technique, Cui et al. (2007) divided the amino acids into three functional groups (clusters); namely hydrophobic, neutral and polar, based upon the amino acid clustering suggested by Chothia and Finkelstein (1990). There are, however, many different ways of clustering amino acids, according to many different physical-chemical properties. Furthermore, it is unlikely that a given amino acid clustering will be the most effective one for all possible protein function prediction problems. The optimal amino acid clustering tends to be strongly dependent on the type of protein being classified.

In this context, this paper proposes an AIS that evolves clusters of amino acids optimized for a given type of protein. The evolved clusters are then used to define the protein representation that will be used by the classification algorithm. In the words of machine learning and data mining, the AIS algorithm solves a clustering (unsupervised learning) problem, consisting of finding the optimal clustering of amino acids for the type of protein whose data is being mined, and the result of the AIS is then used to solve a classification (supervised learning problem).

The proposed AIS is evaluated on a challenging real-world protein function prediction problem: the classification of GPCRs (G-protein-coupled receptors) into their functional classes. GPCRs constitute a large and diverse group of proteins that perform many important physiological functions (Christopoulos & Kenakin, 2002; Gether et al., 2002; Bissantz, 2003). The addressed GPCR classification problem is challenging because it involves a large number of classes organized in a hierarchy – being an instance of the so-called hierarchical classification problem – as will be explained later.

The remainder of this paper is organized as follows. Section 2 describes how the problem of predicting GPCR functions is cast into a classification problem. This section also provides some background on bioinformatics, in order to make the paper more understandable to readers without a biology background. Section 3 described the proposed AIS for clustering amino acids. Section 4 reports computational results, and Section 5 concludes the paper.

2 Casting Protein Function Prediction as a Classification Problem in Machine Learning/Data Mining

2.1 Representing Proteins by Local Descriptors of Amino Acid Sequences

Proteins are large molecules that perform a wide range of vital functions in living organisms. A protein consists of a linear sequence of amino acids – each of which can be represented by a single letter. For instance, the sub-sequence "AVC…" corresponds to (A)lanine, (V)aline, (C)ysteine, … Given a protein's sequence of amino acids, one can try to determine its function via either biological experiments or computational prediction methods. The former produce in general more precise results, but are much more time consuming and expensive. Hence, the latter is often used in practice, and it can provide valuable information for the more cost-effective use of biological experiments. This work addresses the computational prediction of protein function, by casting this problem as a classification (supervised learning) problem in machine learning/data mining, where protein functions are classes and attributes derived from the protein's sequence of amino acids are the predictor attributes.

The number of amino acids in the sequence varies widely across different proteins. Since the vast majority of classification algorithms can cope only with datasets where all examples (records, data items) have the same length, it is necessary to convert all proteins (examples) to the same fixed number of attributes, using an attribute representation at higher level of abstraction than the full sequence of amino acids. The high-level representation used here involves the attribute creation technique defined in (Zhang et al., 2005), which is based on summarizing the protein's entire sequence of amino acids by a fixed number of local descriptors (attributes), as follows.

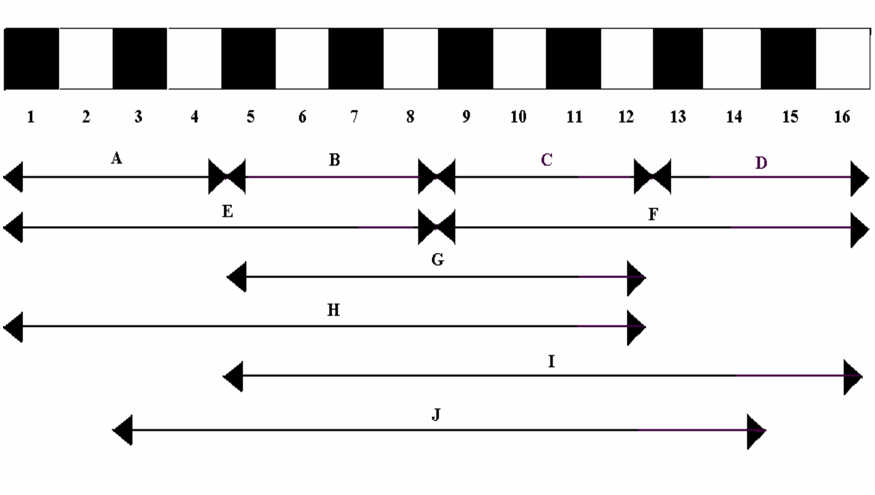

Fig. 1. The 10 descriptor regions (A-J) for a hypothetical protein sequence of 16 amino acids. Adapted from Zhang *et al.* (Zhang et al., 2005) (unpublished)

Cui et al. (2007) divided the amino acids into three functional clusters: hydrophobic (amino acids C,V,L,I,M,F,W), neutral (amino acids G,A,S,T,P,H,Y), and polar (amino acids R,K,E,D,Q,N), as suggested by Chothia and Finkelstein (1990). It is then possible to substitute the amino acids in the sequence for the cluster in which that amino acid belongs. Assuming H=hydrophobic, N=neutral and P=polar, the protein sequence CVGRK would be converted to HHNRR. The position or variation of these clusters within a sequence is the basis of three local descriptors: composition (C), transition (T), and distribution (D).

C is the proportion of amino acids with a particular property (drawn from a particular cluster such as the hydrophobic one). As an example, given the cluster H, we can determine C(H) over the example sequence of HHNRR as 0.4 as 2 of 5 positions in the sequence are of value H. T is the frequency with which amino acids with one property are followed by amino acids with a different property. Thus to compute T(N) over the example sequence, we can see there is a transition between H and N from positions 2 to 3, then a transition from N to R between positions 3 and 4. In this case T(N) = 2/4 = 0.5 as there are 4 places where a transition may occur. Any transitions between H and R are ignored here as neither of these clusters are the subject. Descriptor D measures the chain length within which the first, 25%, 50%, 75% and 100% occurrences of the particular property are located.

Given that the amino acids are divided into three clusters in this instance, the calculation of the C, T and D descriptors generates 21 attributes in total (3 for C, 3 for T and 15 for D). While this technique is valid if applied over the whole amino acid sequence, Zhang et al. (2005) split the amino acid sequences into 10 overlapping regions – see Fig 1. For sequences A-D and E-F there may be cases where the sequence cannot be divided exactly, in which case each subsequence may be extended by one residue. Each descriptor - C, T, and D - is calculated over the 10 subsequences. The number of attributes created with this technique therefore generalises to 70*n*, where *n* is the number of amino acid clusters. In the case of 3 clusters of amino acids, proteins are now represented by 210 numerical descriptors, which can be offered to any of the plethora of well understood, well documented classification algorithms.

In the case of (Zhang et al., 2005), the three clusters as defined in (Cui et al., 2007) were used, however no explicit explanation was included to justify the use of this particular clustering scheme. The keys to the success or failure of the technique described thus far are: (a) the number of clusters used, and (b) the specific amino acids that are included in each cluster. While there exists a truly enormous number of ways to partition the 20 amino acids, it seems clear that some will be more useful than others. However, in general it is not possible to determine, *a priori*, which amino acid clustering will result in the optimal performance for a given protein dataset. In addition, the classifier used may have certain biases that can be exploited during the clustering procedure. Hence, in principle we can use a data-driven approach to evolve an amino acid clustering that approaches optimality with respect to both the data being mined and the classification algorithm applied to that data.

This is the approach followed in this paper, whose main contribution is to present a new variant of the opt-aiNet algorithm for producing an amino acid clustering tailored to the problem of protein function prediction – cast as a classification problem.

2.2 Hierarchical Classification of G-Protein-Coupled Receptors (GPCRs)

Some data can be naturally organised as a hierarchy of classes. The classification of data in such a hierarchy poses some unique challenges to data miners, such as the large number of classes to be predicted. One particular case of this is the classification of G-Protein Coupled Receptor (GPCR) proteins by their function. GPCRs are important proteins as they can transmit messages from a cell's exterior to its interior, changing that cell's behaviour, and approximately 50% of all marketed drugs are targeted towards GPCRs (Klabunde & Hessler, 2002).

The method of optimising clusters for a local descriptor-based attribute construction technique, as proposed in this paper, is generic to any protein dataset where it is sensible to represent the data using the local descriptors representation, but it should be pointed out that the GPCR dataset used in this study is hierarchical in nature. Because of this, the algorithm used to assess the quality of the attriute-construction technique and compare it with a baseline is also hierarchical in nature. Most extant classifiers deal with flat data sets, i.e., data for which a single level of classes may be assigned to an example. In a hierarchical dataset an example may be assigned to one class at a number of levels of specialisation. The most general level being near the root of the tree and becoming more specialised as the tree's branches are traversed. In this paper we deal only with structures where each class has exactly 1 parent – i.e. the data is structured like a tree. The class structure of a typical flat dataset will contain, for example, classes A, B and C which are all equally different from each other. However, in a hierarchy some classes may be more alike than others. Classes A and B are equally dissimilar, but these classes may subdivide such that classes A1 and A2 are more alike than A1 and B1 as A1 and A2 share a common parent class. For more details about the hierarchical classification of GPCRs, see (Secker et al., 2007a).

3 The Proposed Artificial Immune System for Amino Acid Clustering

Pseudocode 1 shows the most general view of the process of attribute creation based on amino acid clustering (performed by an AIS) and subsequent use of a classification algorithm. Note that this process of attribute *creation* (or *construction*) based on clustering should not be confused with attribute *selection*. The goal of attribute selection is essentially to choose a subset of relevant attributes, out of all available attributes. This work rather involves attribute construction, where the goal is to create new attributes (new descriptors of amino acid sequences corresponding to higher-level information about proteins) based on the original sequence of amino acids (corresponding to lower-level information about proteins). The actual process of attribute creation is performed by using a clustering algorithm that groups together similar amino acids, and the result of this clustering is then used to produce a new set of predictor attributes for the classification algorithm.

```
1. Split full dataset into training and testing sets
2. Split training set into sub-training and validation sets
3. Generate initial random candidate clustering solutions
4. Evolve clustering
   4a. Create attributes for sub-training and validation
       data from clusters
   4b. Train classifier on sub-training data
   4c. Evaluate classifier on validation data
   4d. Assign quality to this clustering
   4e. Update population depending on individual's quality
   4f. Repeat from 4 until stopping criterion is met
5. Return the best clustering from the population
6. Create attributes for training and testing datasets using
   this best clustering
7. Train classifier using newly transformed training set
8. Evaluate classifier using newly transformed test set.
```

Pseudocode 1. High level description of amino acid clustering-based attribute creation and subsequent use of classification algorithm.

In Pseudocode 1, points 1 and 2 are standard pre-processing tasks. Point 3 initialises the population for the AIS that performs amino acid clustering; while point 4 and sub-points thereof describe, at a high level of abstraction, the evolutionary process of amino acid clustering. Point 6 uses the output of the AIS (point 5) to create the data which will form the input to the classification algorithm, while points 7 and 8 are the standard training/testing steps used in a classification scenario.

The proposed AIS for amino acid clustering is a new variant of opt-aiNet, which we call opt-aiNet-AA-Clust (opt-aiNet for Amino Acid Clustering). The original opt-aiNet is an optimiser based on abstract ideas of clonal selection and somatic hypermutation as found in natural immune systems (de Castro & Timmis, 2002b). Opt-aiNet was first proposed in (de Castro & Von Zuben, 2001; de Castro & Timmis, 2002a), and updated slightly in (Timmis & Edmonds, 2004). In this latter paper, opt-aiNet was proposed as a function optimisation tool. In this case, each immune cell would encode a single floating point value – the input to the function to be optimised.

Several modifications were required to allow the opt-aiNet algorithm to work in our scenario of amino acid clustering. These included the changing of the individual representation from a real value to a string of symbols to represent clusters, the changing of the fitness evaluation from a straightforward mathematical function to a much more complex system for creating and evaluating the attributes produced by the clustering results and some minor procedural changes such as the termination function. In the case of the original opt-aiNet, the algorithm will terminate when there has been no improvement above a threshold in the population between successive iterations. In this case, it is possible that many iterations could pass before an improvement is found and thus the system terminates after a given number of iterations. These changes are explained in more detail below.

Individual (Immune Cell) Representation – Each individual (immune cell) encodes a candidate solution to the problem of clustering the 20 amino acids. More precisely, each individual consists of a vector with 20 elements, <c_1, …, c_{20}>, where the *i*th element, c_i = 1,..,20, indicates the id of the cluster to which the ith amino acid is assigned – since there are 20 amino acids. To consider a simple hypothetical example, if the first five elements of a vector were 3, 1, 2, 1, 3, this would mean that the second and fourth amino acids would be assigned to the same cluster (arbitrarily denoted as cluster 1); the first and fifth amino acids would be assigned to another cluster (denoted as cluster 3); and the third amino acid would be assigned to yet another cluster (denoted as cluster 2); and so on, for all the 20 amino acids. Different individuals can produce different numbers of clusters.

The Algorithm's Pseudocode and Search Operators – The opt-aiNet-AA-Clust algorithm proceeds as shown in Pseudocode 2, which is a more detailed description of points 4a-4f from Pseudocode 1. Thus, the algorithm is initialised by generating a population of immune cells such that the representation of each immune cell is in a random configuration. That is, amino acids are randomly assigned to clusters. Next, the quality of each immune cell (that is, the accuracy of the attributes defined by the clustering represented by that individual) is assessed. This is a somewhat complex process, explained in the Fitness Function paragraph. Each immune cell is then cloned (copies of that cell are produced) mimicking the clonal expansion stage of an immune reaction. These clones are mutated with a rate inversely proportional to their parent's (and therefore their) quality. The mutation scheme used in this algorithm is somewhat different to the original opt-aiNet. In the latter, the single value encoded by each immune cell will be incremented or decremented with a magnitude based on its fitness. However, a mutation in this context is simply a change in one or more positions in the immune cell's representation. This has the effect of switching an amino acid from one cluster to another. As well as switching an amino acid between clusters, this would include taking the amino acid out of a cluster with others and placing it in a cluster on its own or vice versa. The better the solution encoded by an immune cell the fewer positions are mutated. This has the effect of drastically changing poorly performing clustering schemes in the hope that a better solution may be found, while at the same time not destroying solutions that are already good. These newly mutated clones are then assessed for quality once again and the best solution is kept to form part of the next generation. When all immune cells in the population have been cloned and mutated a small number of badly performing cells are discarded. These are replaced in the population with an equal number of randomly configured immune cells. This injection of randomness into the population discourages the population converging prematurely on a single local optimum.

```
1.Initalise population with each cell having randomly generated
   features
2. While (stopping criteria not met)
    2a. Determine fitness of each cell
    2b. Generate clones for each cell, keeping the parent cell in
         the population
    2c. Mutate each clone based on the fitness of its parent
    2d. Determine the fitness of all new clones
    2e. For each parent cell, select its fittest clone for
         survival into next generation
    2f. Determine average fitness of the population. If it has
         improved significantly, then loop from 2.
    2g. Remove the least fit cells from the population
    2h. Replace the cells removed in 2.g. with randomly generated
         new cells
```

Pseudocode 2. opt-aiNet (adapted from (de Castro & Timmis, 2002a))

Fitness Function – The original opt-aiNet used a single mathematical function as a measure of quality whereas the assessment of quality for each immune cell in this scenario is not as straightforward. Several stages must be gone through to assess the quality of the representation as encoded by the immune cell. For each immune cell, the clustering must firstly be translated from the immune cell representation, as explained earlier. The clusters defined can then be used to create a set of predictor attributes. In detail, each protein sequence in the training data set is split into 10 regions as defined in Fig. 1. Then the C, T and D (Composition, Transition and Distribution) values are determined for each protein subsequence (A-J) based on the clusters defined by the immune cell. This produces a dataset consisting of 70n predictor attributes (where n is the number of clusters as defined by the immune cell). This dataset (the training data) must then be split into two further sets – sub-training and validation. For this algorithm the split between these datasets is 80%/20%. The chosen classification algorithm is now trained on the sub training data and evaluated using the validation data. The quality of the cell's representation is defined as the percentage predictive accuracy output from the classifier on the validation set. Note that this predictive accuracy is measured on the validation set, separated from the sub training set (used to build a classification model), because the goal is to estimate the generalization ability of classification models, as is usual in classification.

Parallel processing – As each immune cell encodes a different set of clusters, it is important to note here that the above-described entire process of creating the new training set from the encoded clustering and then training/evaluating the classifier must be repeated every time a fitness evaluation is requested and each iteration of opt-aiNet-AA-Clust may require many hundreds of such evaluations to occur. The fitness evaluation in this AIS is therefore extremely processor-intensive and as such the assessment of immune cell fitness was distributed over a cluster of 30 computers. Given each node in the cluster has its own copy of the training partition of the data set, each fitness evaluation is atomic in nature. Therefore multiple fitness evaluations can occur simultaneously while the algorithm pauses until all evaluations are complete. The main algorithm can then resume and continue as if the fitness evaluations had taken place in the normal, serial manner. It was found that executing these fitness evaluations in parallel was the only way to ensure the algorithm completed a reasonable number of iterations in a reasonable amount of time.

4 Computational Results

The new variant of opt-aiNet proposed in Section 3 – opt-aiNet-AA-Clust – was implemented by modifying the original opt-aiNet's code kindly obtained from Andrews (2007), which formed part of (Andrews & Timmis, 2005). The WEKA data mining toolkit (Witten & Frank, 2005) was used to provide the classification algorithm used in the fitness function, many of the algorithms used in the selective top-down classifier and a number of auxiliary functions regarding data manipulation. Some algorithms from (Brownlee, 2006) were also used in the selective top down classifier. The dataset used for training and testing was our own comprehensive dataset of GPCR sequences. This dataset, called the GDS dataset, originally contained 8354 protein sequences (examples), but classes with fewer than 10 examples were discarded – since in general such rare classes cannot be reliably predicted. This left 8222 protein sequences in the dataset. The dataset contains 5 classes (A-E) at the family level (the first level), 40 classes at the sub-family level and 108 classes at the sub-sub-family level (the third level). This dataset is described in more detail in (Davies et al., 2007).

For each run of opt-aiNet-AA-Clust, the algorithm was run on the training data and then the classification algorithm was trained on the same training data. Hence, following standard machine learning principles, no data used during the amino acid clustering stage was present in the ultimate testing of the classifier. For each run of the algorithm the number of training items was reduced to half the size by random sampling, in order to reduce processing time – due to the rather processor-intensive fitness function.

Ideally, the opt-aiNet-AA-Clust's fitness function would use a classification algorithm to predict classes in all 3 hierarchical levels of GPCR function. However, this is prohibitively slow with each individual evaluation likely to take many hours. Clearly a faster solution must be found. It was decided that just one classifier should be used in the fitness function. As 1-Nearest Neighbour (1-NN) has appeared to be the more accurate than other classifiers on this type of data in preliminary tests, it was chosen here. As only one classifier is to be used, it was decided that for the purpose of fitness computation the classifier will distinguish between classes only at the top level of the hierarchy (GPCR families A-E).

For each opt-aiNet-AA-Clust run, the algorithm performs 40 generations, using a population size of 20 individuals. While the algorithm was allowed to form clusters using any combination of amino acids, a limit of 5 clusters per individual was enforced. Because of the way the clustering is used to produce the predictor attributes, large number of clusters per individual results in a very large number of predictor attributes, and so the classifier becomes too slow to train and test in a reasonable amount of time. Thus, it was decided that 5 clusters struck a reasonable balance between the algorithm's flexibility and constraining the time taken during evaluation of the representation. Table 1 shows the parameters used for each run of opt-aiNet-AA-Clust.

Table 1. opt-aiNet-AA-Clust parameters

Number of initial cells in the network	20
Number of clones for each immune cell during clonal selection	20
Number of algorithm iterations	40
Suppression threshold for network cell affinities	0.5
Maximum number of clusters that can be produced by each immune cell	5
Fitness evaluation method	1-NN classifier

To assess the effectiveness of the proposed algorithm, an experiment was undertaken to compare the accuracy of a classifier when attributes are evolved by the algorithm against a baseline. As stated above, the dataset used was our GDS dataset. In the case of the baseline, attributes were generated from raw protein sequences by the approach of Zhang et al. (2005), as described earlier. For each set of constructed attributes the same classification algorithm was used. In this case it was the selective top down classification algorithm as defined in (Davies et al., 2007) and (Secker et al., 2007b). In other words, the experiments compare the performance of a given hierarchical classification method in two different scenarios, using two different types of predictor attributes: the attributes created by using our proposed opt-aiNet-AA-Clust and the baseline attributes proposed by Zhang et al. (2005). Hence, what is ultimately being compared is the effectiveness of two different protein representations: one of them automatically evolved by opt-aiNet-AA-Clust and the other manually proposed by Zhang et al. using their domain knowledge about proteins and amino acid properties.

Because of the sheer amount of time taken to evolve the protein representations, only one run of a 10-fold cross-validation procedure – a standard procedure for evaluating predictive accuracy in data mining (Witten & Frank 2005) – was performed with opt-aiNet-AA-Clust. However, as the experiments with the baseline representation have been run before during other investigations, the results of 10 runs of a 10-fold cross-validation procedure (100 runs of the classifier in total) are available. The results are shown in Table 2 where the mean predictive accuracy over the 10 folds of the cross-validation procedure is shown. The mean accuracies for the baseline are shown and finally the statistical significance of the difference between the accuracies of the evolved representation and the baseline is displayed. This has been computed using Student's t-test with 2-tails. This test was used as the number of runs is small while it can be used to compare distributions where there are different numbers of observations for each. In this case, 10 observations for the evolved attributes and 100 for the baseline.

Table 2. Predictive accuracy (%) per class level

	1st level	2nd level	3rd level
Classifier using attributes evolved by opt-aiNet-AA-Clust	96.91	83.14	72.75
Classifier using baseline attributes	96.97	82.72	70.46
P value result of Student's t-test	0.775	0.280	0.003

It can be seen from the table that the difference in the predictive accuracy of the two approaches on the first (most general) and second class levels are statistically negligible – the t-tests produced high p values. On the other hand, at the third class level the attributes evolved by opt-aiNet-AA-Clust led to a very significant improvement in predictive accuracy over the baseline attributes, statistically significant at the 1% level.

It should be noted that the third class level represents the most challenging classification scenario, since it involves many classes and typically a smaller number of examples per class (making generalization more difficult), as compared with the first two levels. In addition, classes at the third level are often more informative to biologist users, since they specify a protein's function more precisely.

It should be stressed that, although the automatically evolved clusters of amino acids have led to an improvement for the particular dataset of GPCR proteins used in our experiments, there is no guarantee that the same evolved amino acid clusters will be optimal for predicting other types of protein functions. However, the proposed algorithm is generic enough to be easily applicable to other types of proteins, offering us an automated approach for trying to find a near-optimal cluster of amino acids tailored to the type of protein whose functions have to be predicted.

5 Conclusions

Previous experience has shown that the protein representation generated by the local descriptors method results in highly competitive predictive accuracies when attempting to classify GPCR proteins. The local descriptors technique, as currently published in the literature, divides amino acids into 3 clusters, leading to a specific set of predictor attributes. When evaluating this published representation, , we found no clear reason why these three clusters were used. It was therefore hypothesised that predictive accuracy could be improved over this "one size fits all" set of clusters by assigning amino acids to clusters in a data driven manner. In this spirit, this paper proposed a new variant of opt-aiNet, called opt-aiNet-AA-Clust, that optimizes the clustering of amino acids for the type of protein being mined and for the type of classification algorithm being used.

When compared against the original local descriptors-based representation, which was not optimized for the data nor for the classification algorithm, it was found that a significant increase in predictive accuracy was observed at the 3rd level of the class hierarchy, which is the most informative (most specialized) type of protein function for the user.

One future direction would be to let the AIS algorithm have free reign to decide the number of clusters. It is thought that allowing an unlimited number of clusters could result in better predictive accuracy. However, in the experiments reported here this was impractical as, firstly, the AIS would have a hugely increased solution space to search, which would require an increase in time taken to solve the clustering problem. Secondly, an increase in the number of clusters defined by the solution returned by the AIS would result in a huge number of attributes being created for the data, which can be impractical when using a hierarchical classification algorithm.

References

Andrews, P. (2007). opt-aiNet source code in Java, last modified October 2005. (Personal communication, 10 July 2007)

Andrews, P. S., & Timmis, J. (2005). *On Diversity and Artificial Immune Systems: Incorporating a Diversity Operator into aiNet.* International Workshop on Natural and Artificial Immune Systems (NAIS), Vietri sul Mare, Salerno, Italy. Lecture Notes in Computer Science 391. pp. 293-306

Bissantz, C. (2003). Conformational changes of G protein-coupled receptors during their activation by agonist binding. *J Recept Signal Transduct Res, 23*, 123-153.

Brownlee, J. (2006). *WEKA Classification Algorithms. Version 1.6.* Retrieved December 2006 from http://sourceforge.net/projects/wekaclassalgos

Chothia, C., & Finkelstein, A. V. (1990). The Classification and Origins of Protein Folding Patterns. *Annual Review of Biochemistry, 59*, 1007-1035.

Christopoulos, A., & Kenakin, T. (2002). G protein-coupled receptor allosterism and complexing. *Pharmacology Review, 54*, 323-374.

Cui, J., Han, L. Y., Li, H., Ung, C. Y., Tang, Z. Q., Zheng, C. J., Cao, Z. W., & Chen, Y. Z. (2007). Computer prediction of allergen proteins from sequence-derived protein structural and physicochemical properties. *Mollecular Immunology, 44*, 514-520.

Davies, M. N., Secker, A., Freitas, A. A., Mendao, M., Timmis, J., & Flower, D. R. (2007). On the hierarchical classification of G Protein-Coupled Receptors. *Bioinformatics, 23*(23), 3113-3118.

de Castro, L., & Von Zuben, F. (2001). Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, 6*(3), 239-251.

de Castro, L. N., & Timmis, J. (2002a). *An artificial immune network for multimodal optimisation.* 2002 Congress on Evolutionary Computation (CEC 2002). Part of the 2002 IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, USA. IEEE, pp. 699-704

de Castro, L. N., & Timmis, J. (2002b). *Artificial Immune Systems: A New Computational Intelligence Approach*: Springer-Verlag.

Gether, U., Asmar, F., Meinild, A. K., & Rasmussen, S. G. (2002). Structural basis for activation of G-protein-coupled receptors. *Pharmacological Toxicology, 91*, 304-312.

Klabunde, T., & Hessler, G. (2002). Drug Design Strategies for Targeting G-Protein Coupled Receptors. *ChemBioChem, 3*, 928–944.

Secker, A., Davies, M. N., Freitas, A. A., Timmis, J., Mendao, M., & Flower, D. R. (2007a). An Experimental Comparison of Classification Algorithms for the Hierarchical Prediction of Protein Function. *Expert Update (Magazine of the British Computer Society's Specialist Group on AI), Special Issue on the 3rd UK KDD (Knowledge Discovery and Data Mining) Symposium, 9*(3), 17-22.

Secker, A., Davies, M. N., Freitas, A. A., Timmis, J., Mendao, M., & Flower, D. R. (2007b). *An Experimental Comparison of Classification Algorithms for the Hierarchical Prediction of Protein Function.* 3rd UK Data mining and Knowledge Discovery Symposium (UKKDD 2007), Canterbury. pp. 13-18

Timmis, J., & Edmonds, C. (2004). *A Comment on opt-AINet: An Immune Network Algorithm for Optimisation.* Genetic and Evolutionary Computation Conference (GECCO 2004). Lecture Notes in Computer Science 3102. Springer, pp. 308-317

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd Edition ed.). San Francisco: Morgan Kaufmann.

Zhang, Z. H., Tammi, M. T., Zhang, G. L., & Tong, J. C. (2005). Prediction of protein allergenicity using local description of amino acid sequence. *(unpublished).*