

# Incremental Learning for Classification of Protein Sequences

Shakir Mohamed, David Rubin, Tshilidzi Marwala

**Abstract**—The problem of protein structural family classification remains a core problem in computational biology, with application of this technology applicable to problems in drug discovery programs and hypothetical protein annotation. Many machine learning tools have been applied to this problem using static machine learning structures such as neural networks or support vector machines that are unable to accommodate new information into their existing models. We utilize the fuzzy ARTMAP as an alternate machine learning system that has the ability of incrementally learning new data as it becomes available. The fuzzy ARTMAP is found to be comparable to many of the widespread machine learning systems. The use of an evolutionary strategy in the selection and combination of individual classifiers into an ensemble system, coupled with the incremental learning ability of the fuzzy ARTMAP is proven to be suitable as a pattern classifier. The algorithm presented is tested using data from the G-Coupled Protein Receptors Database and shows good accuracy of 83%.

## I. INTRODUCTION

Protein sequence analysis has become important area of research due to its application in drug discovery programs [1] with computational analysis becoming popular. Consider the problem of new drug development, which often takes up to 15 years and costing up to \$700 million per drug under investigation [1]. Computational tools have had the most impact in the discovery phase of drug design. In pharmaceutical drug discovery programs it is often useful to classify the sequences of proteins into a number of known families. In a mathematical notation, if it is known that a sequence  $S$  is obtained for some disease  $\mathcal{X}$ , and that  $S$  belongs to family  $\mathcal{F}$ , treatment for the disease is initially determined using a combination of drugs that are known to apply to  $\mathcal{F}$  [2].

The G-Protein Coupled Receptors (GPCRs) are the most important superfamily of proteins found in the human body. Many classification systems have been developed over the years based on machine learning to classify sequences as belonging to one of the GPCR families, and have shown great success in this task. These classification systems produce static classifiers which cannot accommodate any new sequences that may be discovered, and do not aid in solving any of these grand problems.

This paper introduces the use of a classification system based upon an evolutionary strategy, incremental learning and the Fuzzy ARTMAP to realise a protein classification system for the GPCR protein superfamily that allows all-vs-all comparison of these proteins. Being an incremental system, the classifier is dynamic and has the ability to incorporate new information into the classification model.

Authors are with the School of Electrical and Information Engineering, University of the Witwatersrand, South Africa. Email: {d.rubin, t.marwala}@ee.wits.ac.za

## II. IMPORTANCE OF THE GPCRS

The G-Protein Coupled Receptors (GPCRs) are a superfamily of proteins and forms the largest superfamily of proteins found in the human body. The GPCRDB is a database dedicated to the storage and annotation of G-Coupled proteins and at present consists of 16764 entries [3]. GPCRs play important roles in cellular signalling networks in processes such as neurotransmission, cellular metabolism, secretion, cellular differentiation and growth and inflammatory and immune responses. Because of these properties, the GPCRs are the targets of approximately 60%–70% of drugs in development today [4] and results in more than US\$23.5 billion in pharmaceutical sales revenue from drugs which target this superfamily.

The GPCR superfamily consists of five major families and several putative families, of which each family is further divided into level I and then into level II subfamilies. The extreme divergence among GPCR sequences is the primary reason for the difficulty of classifying these sequences [1]. In this research eight GPCR families are considered from the number of families available at the GPCRDB, with the sequences being stored in the EMBL format.

## III. REVIEW OF IMPORTANT TOOLS

### A. Overview of Fuzzy ARTMAP

TFuzzy ARTMAP is a neural network architecture based on Adaptive Resonance Theory (ART) that is capable of supervised learning of arbitrary mappings of clusters in the input space and their associated class labels; that was introduced by Carpenter et al [5]. The key features of this type of network architecture is that it is capable of fast, online, supervised, incremental learning, classification and prediction [5]. Figure 1 shows the structure of the fuzzy

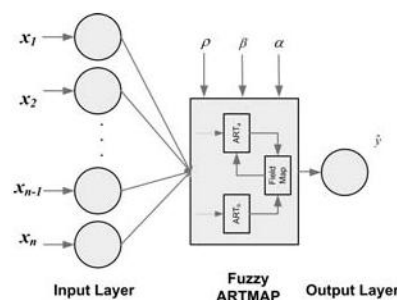


Fig. 1. Representation of the Fuzzy ARTMAP Architecture

ARTMAP. This system takes  $n$ -dimensional input patterns and maps them into the  $n$ -dimensional feature space. The system divides this input space into a number of hyperboxes of varying size, and maps these hyperboxes to a category

in the output space, i.e to the class label. The network learns and adjusts its parameters on a per-pattern basis, not after entire cycles as in the standard neural network model. This is known as instance-based learning and thus each individual input pattern is mapped into the feature space, existing hyperboxes are increased to accommodate the new pattern or a new hyperbox is created. If a new hyperbox is created, this hyperbox is also related to the output class. This entire process is controlled through a set of internal weights and a process known as match tracking. It is this instance-based learning that gives the fuzzy ARTMAP its incremental learning ability. This instance-based learning also makes the order in which training patterns are received an important factor, one which is not often considered in the use of fuzzy ARTMAP networks [6].

The fuzzy ARTMAP is controlled by three parameters: the vigilance  $\rho$ , the learning rate  $\beta$  and the choice parameter  $\alpha$ . The choice parameter is a constant and is kept small, generally 0.001, as used in this application. The learning rate adjusts the factor by which the hyperboxes are increased each time a new training pattern is received, and can be any value between zero and one. For  $\beta < 1$ , the network is said to be in *fast-commit slow-recode* mode, resulting in the hyperboxes increasing in a size proportional to the value of  $\beta$ . If  $\beta = 1$ , the system is in *fast learning* mode and the hyperboxes will be enlarged just enough to include the point represented by the input vector. The vigilance controls how large any hyperbox can become, and will result in new hyperboxes being formed, if the measured *degree* to which an input pattern *belongs* to a hyperbox is less than the vigilance. From this it is observed that the larger the vigilance (higher expected degree of belonging) the smaller the hyperboxes created in the input space. This is a key factor to consider in the application of fuzzy ARTMAP systems, since large values of  $\rho$  will result in what is known as category proliferation, which will be observed as overtraining in the system [6].

### B. Overview of the Genetic Algorithm

Genetic algorithms (GA) find approximate solutions to problems by applying the principles of evolutionary biology, such as crossover, mutation, reproduction and natural selection [7]. The GA search process consists of the following steps: 1) Generating a pool of candidate solutions and encoding all values in a binary or floating point representation. 2) Evaluation of the fitness for each chromosome in the gene pool. The fitness is determined via a fitness function defined for the problem being solved, and chromosomes with the lowest fitness are discarded and make way for a new set of chromosomes. Replacement sets of chromosomes are created by the genetic operations of crossover and mutation on the most fit individuals. These genetic operations add an element of randomness to the search process allowing a wider range of the solution space to be explored. 3) Steps 1 and 2 are repeated until a specified fitness level is attained or the maximum number of generations is exceeded [7].

## IV. PRIOR WORK

The problem of incremental learning has not been considered before as it is presented here. Vijaya et al [8] consider the incremental clustering of protein sequences, but that is a different problem from that considered here. The fuzzy ARTMAP has been chosen as the incremental classifier and as mentioned, has been shown to be an effective incremental classifier [5]. The fuzzy ARTMAP has been chosen specifically because of this ability for incremental learning, quick classification times, the ability for multiclass classification and its ability to learn complex data. The Support Vector Machine (SVM) is widely used in protein classification and it would appear that the use of an incremental SVM would be more suitable. While some algorithms for incremental SVM [9] exist, the problem with many of these systems is that they cater to the binary-classification problem only and are not applicable to multi-class classification problems, which is the case for the classification of proteins into families. Other incremental classification systems also exist, such as incremental common-sense models and incremental fuzzy decision trees. Of these incremental classification systems, the fuzzy ARTMAP is the most established and well known, being used in systems such as the MIT Lincoln Laboratory (LL) sensor fusion system, medical diagnosis systems and function estimators [10] among others; and is thus used.

## V. SYSTEM OVERVIEW

A schematic representation of the system is shown in figure 2. Input sequences are extracted from a protein database

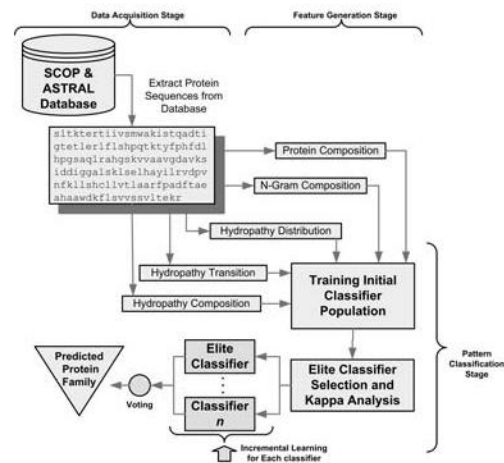


Fig. 2. Overview of System Architecture

and then converted into a numerical feature vector. We then create a population of classifiers to introduce classification diversity, with the selection of suitably diverse classifiers from this population using the Genetic Algorithm coupled with kappa analysis. An ensemble of classifiers is used as a means of introducing modularity in the learning system. This system is implemented using the fuzzy ARTMAP (FAM) and a series of experiments are conducted to evaluate the performance of this system. Pseudocode for the creation

and operation of the system is shown in algorithm listing V. The ability of the FAM as an alternative classifier to many of the other more popular classifiers is demonstrated by comparing the classification ability of these systems using the GPCR data set. The incremental learning system described by algorithm listing V is then tested using the GPCR data and shown to be able to learn new data as well as maintain existing data.

---

**Algorithm V.1: FUZZY ENSEMBLE( $D$ )**

---

**Training Phase**

**comment:** Create population  $j$  of FAM classifiers each trained with a different permutation of the input data  $\mathbf{X}_1$   
Each classifier is a hypothesis  $h_t : \mathbf{X}_1 \mapsto \mathbf{Y}_1$   
 $\epsilon = \frac{1}{N} \sum_{h_t(x_i) \neq y_i} n_i$   
**comment:** Sort classifiers based on incr. error on validation set.  
**comment:** SORT( $\epsilon$ )  
**comment:** Select lowest error classifier as elite classifier  $h_{elite}$   
**comment:** Calculate the agreement  $\kappa$ , of the 15 best classifiers (based on error) with respect to the elite classifier  
 $\kappa = \frac{N \sum_{i=1}^N x_{ii} - \sum_{i=1}^N x_{i+*} x_{*i}}{N^2 - \sum_{i=1}^N x_{i+*} x_{*i}}$   
**comment:** Genetic Algorithm selection of  $p$  classifiers based on a trade-off between error  $\epsilon$  and agreement  $\kappa$   
 $GA_{fitness}(\kappa, \epsilon) = \lambda \sum_{i=1}^p \kappa_i + \sum_{i=1}^p \epsilon_i$   
Create ensemble classifier using the elite classifier  $h_{elite}$  and the  $p$  selected classifiers  $h_t, t = 1, \dots, p$   
**comment:** Fusion of individual predictions using majority voting.

**Operation Phase**

If predicting sequence family, convert to feature representation and classify using the Fuzzy ARTMAP based system created during this previous training phase  
**comment:** If incrementing system knowledge, increment each classifiers in the Fuzzy ARTMAP base system independently, using the training data for new sequences  
 $h_t^{incr} = \mathcal{T}(h_t, \mathbf{X}_k \mapsto \mathbf{Y}_k)$ ,  
where the transformation  $\mathcal{T}$  is the incremental training process and  $k$  is the dataset to be added to the system

---

## VI. PROTEIN VECTORISATION

The data obtained from the GPCRDB is in the form of amino acid sequences. In order for these sequences to be used in classification systems, they must be converted into a numerical form. Before this conversion though, preprocessing in the form of outlier removal must be completed. Outlier removal consists of removing sequences which have characters which are not part of the standard 20-letter amino acid alphabet — the letters are B and Z and have ambiguous meanings. Once this process is complete, these protein sequences must be transformed into numerical features. Two types of features have been identified in the literature, these being global and local features. Huang *et al* [11] provide a good description of the difference between global and local features and this distinction is used in this work.

### A. Global Feature Generation

Global features represent the nature of the entire protein sequence. These features must capture the global similarity

between related sequences allowing for comparison. Consider the amino-acid composition of the sequence. The composition is simply the presence frequency of each of the 20-possible amino acids in the given sequence. Thus the composition is calculated by [12]:

$$\nu_i = \frac{s_i}{\sum_{j=1}^{20} s_j}, \text{ for } i = 1, 2, \dots, 20. \quad (1)$$

where  $\nu_i$  is the value for the  $i$ th feature and  $s_i$  is the number of times the  $i$ th amino acid appears in the sequence. This results in 20 features: a frequency of appearance for each of the possible amino acids. If a particular amino acid does not appear at all in the sequence, the corresponding feature value is zero.

A second set of features based on the hydrophathy of amino acids in a given protein sequence is also calculated. Amino acids are either hydrophobic, hydrophilic (polar) or neutral. We use the Chothia and Finkelstein [13] hydrophathy classification. We calculate three descriptors, the hydrophathy composition ( $\mathcal{C}$ ), the hydrophathy distribution ( $\mathcal{D}$ ) and the Hydrophathy transmission ( $\mathcal{T}$ ) for the sequences as described by Dubchak [13].

The composition  $\mathcal{C}$  is calculated similarly to the amino acid composition described previously. In this case we calculate the presence frequency of hydrophobic, hydrophilic and neutral amino acids in the sequence. This results in three features being generated. The transmission is defined by three values. The first is the number of times a polar molecule is followed by a neutral molecule or vice versa. Similarly the other two are the number of times a neutral molecule is followed by a hydrophobic molecule or vice versa and the number of times the polar molecule is followed by a hydrophobic molecule or vice versa.

The distribution looks at intervals of 25%, 50%, 75% and 100% along the sequence length. For each interval the presence frequency of hydrophobic, hydrophilic and neutral molecules for each percentage interval is calculated. This results in 12 features, 4 features for each of the three hydrophathy groups. A more detailed description of these features can be found in Dubchak [13]. In total 38 features (20+3+3+12) are generated based on global sequence descriptors.

### B. Local Feature Generation

The local features capture local interactions between amino acids and groups of amino acids in a protein sequence. The  $n$ -gram method is well established as a good descriptor of local similarities in a sequence and has been used by many authors such as Cheng *et al* [1], Tomovic *et al* [14] and Zhao *et al* [15]. Essentially the  $n$ -gram method considers the presence frequency of consecutive  $n$ -letter combinations in the protein sequence, for integer  $n$ . For example, consider the short sequence SLTKTERTIIVSM, the 2-grams of this sequence are: SL, LT, TK, KT, etc. Given a sequence, features are generated by calculating the presence frequency of all possible  $n$ -grams for the amino acid alphabet.

A total of 438 features have been generated and as a final post-processing step undergo min-max normalisation. The normalisation is a requirement for using the FAM, since the FAMs complement coding scheme assumes normalised data.

## VII. INCREMENTAL ALGORITHM AND DIVERSITY

The creation of the committee-based system is based on a novel approach, implementing an evolutionary strategy which was summarised in algorithm listing V. We first train an initial population of  $j$  classifiers, each classifier having been trained with a different permutation of the input training data. This permutation is needed in order to add diversity to the classifiers being created. As mentioned, the fact that the fuzzy ARTMAP learns in an instance-based fashion, makes the order in which the training patterns are received an important factor [6]. In the experiments performed, the initial population consists of 30 classifiers.

The classification error  $\epsilon$  of each of these classifiers is then evaluated against a validation data set. The classifiers are then ranked in terms of increasing error. The lowest error classifier from this population is the *elite classifier* and is the classifier that automatically becomes a member of the ensemble system. The inclusion of this elite classifier ensures that at least one high accuracy classifier is selected for the committee.

The next step is to select the remaining  $n$  classifiers. In this application we select a further 4 classifiers. The selection of the other members of the committee is important and requires a number of factors to be considered:

- We do not wish to select classifiers that perform exactly as the elite classifier, since this gives no diversity to the predictions that are generated, and thus there is no room for improvement.
- We do not wish to select low accuracy classifiers that will confuse the prediction obtained and thus result in predictions that are more erroneous than a single classifier.

It would appear that these two conditions oppose each other, since high accuracy classifiers would tend to agree on the same predictions, against what we require for point 1. A trade-off between the classifier accuracy and the level of agreement between classifiers is then ideally what is required. This introduces the need for a formal definition of agreement between classifiers.

We use the definition of agreement considered by Petrakos et al [16], and the mathematical description that follows is generally known as kappa analysis. We define the agreement between any two classifiers  $\kappa$  based on the error matrix of the two classifiers [17]. The error matrix shows the number, and for which classes the two classifiers agree on a prediction. Table I shows the format for an error matrix between two classifiers. In the above table,  $Q$  is the number of classes in the data.  $x_{11}$  in the table is the number of test patterns that both classifier 1 and 2 agreed belonged to class  $C_1$ .  $x_{21}$  is the number of test patterns that classifier 1 predicted belonging to class  $C_2$ , but that classifier 2 predicted belonged to class

TABLE I  
DATA FORMAT FOR ERROR MATRIX BETWEEN CLASSIFIERS

Classifier 1	Classifier 2				Totals
	$C_1$	$C_2$	...	$C_Q$	
$C_1$	$x_{11}$	$x_{12}$	...	$x_{1Q}$	$x_{1+}$
$C_2$	$x_{21}$	$x_{22}$	...	$x_{2Q}$	$x_{2+}$
...	...	...	...	...	...
$C_Q$	$x_{Q1}$	$x_{Q2}$	...	$x_{QQ}$	$x_{Q+}$
Totals	$x_{+1}$	$x_{+2}$	...	$x_{+Q}$	

$C_1$ . Similarly, the entire error matrix can be generated using the prediction made by any two classifiers. We determine the error matrices for 15 of the best classifiers in terms of predictions with respect to the elite classifier. The agreement is calculated using the following set of equations, where  $N$  is the number of training patterns used in generating the error matrix [17].

$$\theta_1 = \sum_{i=1}^N x_{ii} \quad (2)$$

$$\theta_2 = \sum_{i=1}^N x_{i+} \cdot x_{+i} \quad (3)$$

$$\kappa = \frac{N\theta_1 - \theta_2}{N^2 - \theta_2} \quad (4)$$

The selection of classifiers from this population, which must essentially minimise both the error of the individual classifiers and the agreement of the classifiers with the elite classifier, is an optimisation problem. We have chosen to implement a Genetic Algorithm as the optimisation tool for this system. The GA implemented for the selection of classifiers is designed to minimise both the agreement and the error of the selected combination of classifiers. The GA is based on a floating point representation, with the arithmetic crossover and non-uniform mutation operators. The GA will select 4 classifiers resulting in two vectors:

$$\epsilon_{GA} = \{\epsilon_1; \epsilon_2; \epsilon_3; \epsilon_4\}$$

$$\kappa_{GA} = \{\kappa_1; \kappa_2; \kappa_3; \kappa_4\}$$

We use a linear combination of these two matrices to define the cost value of a particular selection of classifiers. It is this cost that the GA will attempt to minimise. The cost function is defined by equation 5.  $\lambda$  is introduced as a scalar constant to allow the relative importance of the agreement in the system to be adjusted. In this study  $\lambda = 1$ , which gives equal importance to both the error and the agreement.

$$f(\epsilon, \kappa) = \lambda \sum_{i=1}^4 \kappa_i + \sum_{i=1}^4 \epsilon_i \quad (5)$$

The GA selects the 4 best classifiers that minimises the cost function of equation 5. The Genetic Algorithm was designed to produce 50 generations of solutions with each generation being a population of 30 possible solutions. The crossover rate was set to a high value of 0.8 and a mutation rate of 0.4, and were empirically determined to be the

best values for the experiment. The crossover functions are modified from the standard crossover functions in this case, to ensure that unique classifiers are selected during each generation, that is, preventing the same classifier from being selected twice in a particular generation.

These selected classifiers are then used in parallel, with each of the five classifiers in the system producing an independent set of predictions. These predictions must then be fused together to form the final decision. A number of decision fusion techniques exist. Some of these include the majority and weighted majority voting, trained combiner fusion, median, min and max combiner rules [18]. We adopt the majority voting decision fusion scheme, which simply considers each of the predictions produced by the five classifiers as a vote, with the final prediction for any given pattern given by the prediction that receives the largest number of votes.

#### A. Incremental Learning of Protein Data

The ensemble system is not a useful system if it is not able to accommodate newly discovered sequences that are produced daily. The ability of a classifier to allow this type of knowledge update was also defined as incremental learning. The fuzzy ARTMAP through its instance-based learning is able to incrementally learn new data. This incremental learning can consider two types of data:

- 1) It is possible to add new sequence information for families which the classifier has already been trained with.
- 2) Data of completely new classes can be added to the system, increasing the knowledge that the system has of the general protein domain.

The base system will in general be trained with data of a number of classes. Once new data becomes available, incremental learning of the system is based on incrementally training each of the 5 FAM classifiers in the system with the new data. The system can now be tested with data from all classes it has been trained with, including classes which have been incrementally added to the system.

### VIII. SYSTEM TESTING AND EXPERIMENTAL RESULTS

#### A. Testing Using GPCR Data

The GPCR data is also divided into 6 separate databases  $\mathcal{D}_1, \dots, \mathcal{D}_6$ , with a validation set for database  $\mathcal{D}_1$ . In this case, the datasets have data of all 8 classes which are available. This specific partitioning is used to demonstrate data incremental learning, where new data of classes which the system has already been trained with is added to the system. This case is more appropriate for use with GPCR data where the families are established. The separation of data into these databases is shown in table II.  $\mathcal{D}_v$  and  $\mathcal{D}_t$  represent the validation and testing datasets respectively.

#### B. Comparative Performance

We compare the Fuzzy ARTMAP with other more common machine learning tools such as the Support Vector

TABLE II  
SEPARATION OF DATA INTO INDIVIDUAL DATABASES

Family	$\mathcal{D}_1$	$\mathcal{D}_v$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$	$\mathcal{D}_6$	$\mathcal{D}_t$
Type 1	32	10	43	43	43	43	43	43
Type 2	23	8	30	30	30	30	30	30
Type 3	16	6	22	22	22	22	22	22
Type 4	6	2	9	9	8	8	8	8
Fz/Smo	12	4	16	15	16	16	16	16
MLO	3	1	4	5	5	5	5	4
Class H	32	11	43	43	43	43	43	43
Pheromone 2	20	6	26	26	26	26	27	27

(SVM) Machines and Multi-layer perceptron (MLP). These have been chosen since they have found widespread use in the literature [1], [2], [15]. Table III shows the performance of the classifiers that were considered in the experiment. The parameters that are used for each of the classifiers is included in the table. The classifiers are trained with all the training data combined into a single training set and tested on the test set  $\mathcal{D}_t$ , using the features that were described in section 5. The table shows that the FAM has comparable accuracy

TABLE III  
COMPARATIVE PERFORMANCE OF FAM VERSUS OTHER CLASSIFIERS

Classifier	Error (%)
Generalised Linear Model	25.91
Multi-layer Perceptron, $n_{hid} = 15$ , $cyc = 200$	15.03
Fuzzy ARTMAP $\rho = 0.75$	11.90
SVM - RBF $\gamma = 2.3$	17.10
SVM-Polynomial 2.23 degree	10.36

when compared to many other classification systems.

#### C. Base Classifier Training and Incremental Performance

The base classification system was trained using database  $\mathcal{D}_1$ . The GA for this data set selected classifiers 2, 3, 4, 12 to form the final ensemble system. Again, the system consisting of the elite classifier and the four classifiers selected by the GA are incrementally trained using databases  $\mathcal{D}_2, \dots, \mathcal{D}_6$ , with the ensemble being tested after each increment with the testing database  $\mathcal{D}_t$ . The performance of the system is shown in table IV. This data shows that the

TABLE IV  
TRAINING AND GENERALISATION PERFORMANCE OF SYSTEM

Set	Train 1	Train 2	Train 3	Train 4	Train 5	Train 6
$\mathcal{D}_1$	0	0	0	0	0	0
$\mathcal{D}_2$	—	0	0	0	0	0
$\mathcal{D}_3$	—	—	0	0	0	0
$\mathcal{D}_4$	—	—	—	0	0	0
$\mathcal{D}_5$	—	—	—	—	0	0
$\mathcal{D}_6$	—	—	—	—	—	0
$\mathcal{D}_v$	25.00	22.92	22.92	27.08	25.00	27.08
$\mathcal{D}_t$	22.79	18.65	19.17	19.69	18.65	16.58

system is extremely capable of remembering data that has been trained upon, as shown by the many 0% which appear in the table for the training databases. The many zeros are not an indication of overtraining. The FAM is trained so that it

learns all its training data with a 0% error. What the results show is that after it has learnt its initial training data, the memory is not degraded by the addition of additional data. The system also shows that the performance does increase as more data of each of the classes is added to the system.

## IX. ANALYSIS OF RESULTS

The results presented indicate that the Fuzzy ARTMAP is a suitable machine learning tool for the classification of protein sequences into structural families, which is comparable to many of the more established tools. An analysis of the sequences also shows that the system is able to classify proteins of varying lengths from 32 to 350 amino acids in length, and thus the length of the protein sequences used are not important. The accuracy of the classification could be improved if some form of dimensionality reduction or feature selection is applied. These techniques have been applied by many authors using numerous techniques. Principal Component Analysis has been used as a technique of dimensionality reduction and Cheng et al [1] uses the chi-squared test as a means of feature selection. Feature selection can also be applied using various sub-optimal feature selection techniques such as the floating forward selection search or the Genetic Algorithm can be used as demonstrated by Mohamed *et al* [19].

The agreement  $\kappa$  was used to measure diversity of the system. This might not be the best measure of the relationship between predictions of classifiers to the elite classifier. The use of the correlation coefficient could be explored or the use of a disagreement [17] should also be explored, to determine if this measure gives some degree of refinement in the selection of the classifiers. The genetic algorithm is also important in the committee. Due to the stochastic nature of the GA, it is possible that different GA optimisations produce a different selection of classifier members. This though is not as likely in the case of the data presented here, since many of the classifiers had the same agreement or error, resulting in the GA converging to the same selection choice. That said, the optimisation of the GA is efficient and runs very fast due to the fact that it uses pre-calculated results such as the error matrix and agreement values.

## X. CONCLUSION

The algorithm presented is applicable in general to all classification problems. Where the case exists that any new information that may be obtained will not significantly improve the classification ability of the system, then the batch training approach may be more suitable. Where this is not the case such as families whose sequences have low sequence similarity, then the incremental approach may be better and will be more desirable, especially if prior training data is no longer available. The techniques presented here are also not limited to the problem of structural family classification, and can be easily extended to secondary and tertiary structure prediction, functional annotations and the prediction of protein-protein interaction sites.

## REFERENCES

- [1] B. Y. M. Cheng, J. G. Carbonell, and J. Klein-Seetharaman, "Protein classification based on text document classification techniques," *Proteins: Structure, Function, and Bioinformatics*, vol. 58, pp. 955–970, 2005.
- [2] J. T. L. Wang, Q. Ma, S. Shasha, and C. H. Wu, "New techniques for extracting features from protein sequences," *IBM Systems Journal*, vol. 40, no. 2, pp. 426–441, 2001.
- [3] F. Horn, J. Weare, M. W. Beukers, S. Horsch, A. Bairoch, W. Chen, Ø. Edvardsen, F. Campagne, and G. Vriend, "GPCRDB: an information system for G-protein coupled receptors," *Nucleic Acids Research*, vol. 26, no. 1, pp. 277–281, 1998.
- [4] K. Lundstrom, "Structural genomics of GPCRs," *Trends in Biotechnology*, vol. 23, pp. 103–108, February 2005.
- [5] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, pp. 698–713, 1992.
- [6] A. Koufakou, M. Georgiopoulos, A. Anagnostopoulos, and T. Kasparis, "Cross-validation in fuzzy ARTMAP for large databases," *Neural Networks*, vol. 14, pp. 1297–1291, 2001.
- [7] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer, third, revised and extended ed., 1999.
- [8] P. A. Vijaya, M. N. Murty, and D. K. Subramanian, "An efficient incremental protein sequence clustering algorithm," in *Proceedings of the IEEE Conference on Convergent Technologies for Asia-Pacific Region*, pp. 409–413, October, 2003.
- [9] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *Advances in Neural Information Processing Systems (NIPS 2000)*, vol. 13, pp. 409–415, 2001.
- [10] R. Andonie and L. Sasu, "Fuzzy ARTMAP with input relevances," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 929–941, 2006.
- [11] C. Huang, C. T. Lin, and N. R. Pal, "Hierarchical learning architecture with automatic feature selection for multi-class protein fold classification," *IEEE Transactions on Nanobioscience*, vol. 4, pp. 221–232, December 2003.
- [12] X. Zhao, Y. Cheung, and D. Huang, "A novel approach to extracting features from motif content and protein composition for protein sequence classification," *Neural Networks*, vol. 18, pp. 1019–1028, October 2005.
- [13] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. Kim, "Prediction of protein folding class using global description of amino acid sequence," in *Biophysics*, vol. 92, (USA), pp. 8700–8704, National Academy of Science, September, 1995.
- [14] A. Tomovic, P. Janicic, and V. Keselj, "N-gram-based classification and unsupervised hierarchical clustering of genome sequences," *Computer Methods and Programs in Biomedicine*, vol. 81, pp. 137–153, 2006.
- [15] D. Huang, X. Zhao, and G. Huang, "Classifying protein sequences using hydropathy blocks," *Pattern Recognition*, vol. 30, pp. 2293–2300, December 2006.
- [16] M. Petrakos, J. A. Benediktsson, and I. Kannelopoulos, "The effect of classifier agreement on the accuracy of the combined classifier in decision level fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 2539 – 2546, November 2001.
- [17] Y. M. Bishop, S. E. Fienberg, and P. W. Holland, *Discrete Multivariate Analysis*. Massachusetts: MIT Press, 1977.
- [18] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [19] S. Mohamed, D. Rubin, and T. Marwala, "Multi-class protein classification using Fuzzy ARTMAP," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, (Taipei, Taiwan), pp. 1676–1681, 8 - 11 October 2006.