Robust Classification of MHC Class II Peptides

Dina A. Salem[1], Rania A. Abul Seoud[2], and Yasser M. Kadah[3]

[1]Computer Engineering Department, Misr University of Science and Technology, Giza, Egypt

[2]Electrical Engineering Department, Fayoum University, Fayoum, Egypt

[3]Electrical and Computer Engineering Department, King Abdulaziz University, Jeddah, Saudi Arabia

Correspondence:

Yasser Mostafa Kadah, PhD
Professor of Biomedical Engineering
Electrical and Computer Engineering Department
King Abdulaziz University
P.O. Box 80204
Jeddah 21589
Saudi Arabia
E-mail: ykadah@kau.edu.sa

Abstract

Adaptive immune system is one of the human body's defense mechanisms developed to protect against repeated infection by the same pathogen through immunologic memory. Vaccination uses this concept to design vaccines to protect our bodies from infectious diseases. Some cells of the immune response cannot recognize antigen fragments unless attached to Major Histocompatibility Complex (MHC) molecules. Therefore, predicting peptides that are able to bind to MHC molecules is a key step when designing vaccines. MHC class II is one type of MHC molecules that is characterized by its ability to bind peptides of different length. Machine learning techniques can facilitate discrimination between peptides to classify them into binders or non-binders to MHC class II molecules. However, building a classification model passes through several stages that may influence its final decision. In this study, we design a robust MHC class II peptides classifier using neuro-fuzzy techniques. In particular, we optimize each of the stages involved including construction of training and testing datasets to eliminate bias, mapping variable length peptides into fixed feature vector, mining important features through several feature selection techniques, and choice of neuro-fuzzy classifiers. The experimental results demonstrate the importance of this optimization to obtain objective evaluation and show how bias in the results of such techniques as cross-validation can cause wide variability of outcomes for the same data. This can explain the fluctuations in performance of several techniques and suggests a more robust strategy to use for a more objective comparison of different techniques.

Keywords: Balanced Data, Blind Testing, Filter Feature selection, Fuzzy Feature Selection, MHC class II molecules, Neuro-Fuzzy classifiers, Similarity Reduced Data.

1. Introduction

Cell-mediated immunity is the immune response associated with cells that does not involve antibodies. Major Histocompatibility Complex (MHC) molecules have the main rule to elicit the T cell-mediated immune response. The two main classes of MHC molecules (class I and class II) are cell surface glycoproteins coded on chromosome 6 in humans and called Human Leukocyte Antigen (HLA). T cell antigens cannot be recognized unless its fragments are attached to MHC molecule to be present on the surface of T-cells. After binding to a fragment of a pathogen, MHC class II molecules activate the helper T cells and hence stimulate cellular and humoral immunity. Not all antigenic peptide fractions are able to bind MHC class II molecules [1]. Therefore, predicting which specific peptides are able to bind MHC class II molecules is an important step in vaccine design. Computational immunology is an emerging branch of bioinformatics (also called immuno-informatics) that has potential to play a major rule in this task by reducing time and cost by targeting more precise binding peptide prediction.

Several epitope databases are available to serve the goal of predicting peptides that are able to bind MHC class II molecules. Different allele-specific datasets (that is, a separate dataset for each allele) consisting of binding and nonbinding peptides can be extracted. Similarity between sequences in the resulting dataset may cause biased evaluation measures due to the likely presence of similar peptides in both the training and testing dataset. This is particularly even more problematic when measuring performance using cross-validation, which is a common practice in most previous work [2]. To address this problem, several studies performed similarity reduction on the dataset to remove redundant sequences [3] [4]. However, this approach may result in removing important information from the training dataset. A change in one position between two sequences may be a good reason to cause a change of its state from binder to non-binder or vice-versa. Therefore, the results of each method will highly depend on the similarity reduction technique used, which hinder objective evaluation of its performance. For example; according to the same database –MHCBN [5] – the sequence “AAFAAAKAAAAAA” is classified as a binder while “AASAAAKAAAAAA” is classified as a non-binder to MHC class II although the two sequences are exactly the same for all the positions except for the third one.

To obtain realistic performance results without bias or loss of information, similar sequences from those found in the training data are removed from the testing data used in a blind testing procedure instead of cross-validation. An optimal local alignment approach is used to find the score of the best alignment between each two sequences in the binders and non-binders datasets. Testing data then will only contain a set of sequences that do not show similarity, and all other sequences will form the training data.

To accomplish the prediction goal using machine learning techniques, peptides in the dataset must be represented by a set of features of fixed length (unlike actual MHC class II molecules lengths). MHC class II binding grooves at its end are open which explains their ability to bind different length peptides [6]. Consequently, variable length peptides are to be converted to fixed length ones using feature representation. To reach this goal, different features are extracted from a well-known physicochemical amino acids data repository and averaged over the peptide length. Increasing number of features along with thousands of peptides results in a high dimensional data that are difficult to deal with in addition to their prohibitive computation time and memory requirements. To overcome this problem, feature selection techniques are utilized to select only the most informative features for our problem.

Feature selection techniques are classified into three types namely; filter, wrapper and embedded. The former is characterized over the other two types by its easy and fast computing procedure. In addition, they do not depend on the used classifier so they are implemented once and then integrated with any classifier. Their main duty is to calculate a score for each feature using data intrinsic properties. Features with highest scores remain in the feature vector and the rest are discarded [7]. Filter techniques are either parametric or non-parametric tests. Parametric tests constrain the sample data to have a specific probability distribution and a predefined value of distribution parameters. On the other hand, nonparametric tests need more data to reach the same conclusion using fewer assumptions [8]. Classifying using only the top ranked features offer lower computational and memory requirements while increasing performance due to the lower data dimensionality.

Neuro-fuzzy classifier (NFC) is a type of machine learning network based classifier that is able to build a fuzzy system using neural network learning capabilities. The learning procedure is data driven and operates on local information. A three layer neuro-fuzzy classifier expresses its first layer as inputs, second layer (hidden) corresponds the fuzzy rules (i.e., if-then rules) and outputs form the third layer. Connection weights are the definition of the fuzzy sets. Fuzzy rules are considered as prototypes of training data and so, neuro-fuzzy classifier has the advantage of the possibility of its construction by using training data or fuzzy rules. Neuro-fuzzy have a main characteristic of the ability of being interpreted in linguistics rules [9].

When using neuro-fuzzy classifiers (NFC) on large-scale data sets, nonlinear network parameters often cause significantly higher computation time, which may not be practical in many cases. Scaled conjugate-gradient (SCG) algorithm is known to consume less memory and presenting high convergence rate when training type 1 fuzzy systems [10]. Here, three different implementations of the adaptive NFC based on the work of Bayram [11] [12] are employed and compared in our study with a proposed modification to maintain stable results.

Accordingly, one of the objectives of this study is to highlight the role of some data mining techniques on the prediction goal. Data mining algorithms used are represented in two main categories; outlier detection and classification. Outlier detection is achieved through feature selection techniques which are employed to discard features that have negative influence on the model performance in addition to those having no influence at all. In this context, six different feature selection techniques are compared and their effect on the prediction accuracy is studied. The machine learning technique involved in the classification process is a hybridization between fuzzy inference systems and neural network. Several parameters of the merged techniques are tuned and modifications are proposed to overcome negative issues appeared during implementation.

2. Previous Work

A broad study of available web servers serving the function of predicting peptides binding to MHC class II molecules reported poor performance and recommended focusing on collecting adequate data and enhancing predictive models [13]. Although some previous work showed somewhat accurate prediction, results were still unsatisfactory. One related study aimed to increase accuracy in addition to minimizing the time consumed in the prediction phase when training using Fuzzy neural network (FNN). FNN was recommended as a suitable predictor with a slow processing problem. A proposed solution was to use boosted fuzzy classifier with a SWEEP operator method (BFCS) [14]. Their model was successful when trained on a dataset of 1050 peptides for HLA-DRB1*0401 in the context of fast processing, easily obtaining linguistic rules and a slight increase in accuracy. This study, however, suffered from three main drawbacks. Data were downloaded from two databases only resulting in low number of peptides. Features selected for discrimination were only three without a rationale for the criteria of choice. Moreover, evaluation of model involved cross-validation not blind testing [2]. The last drawback was solved later in another study by examining its model using blind testing on a separate data [4]. However, this work did not remove similar peptides from the blind dataset and hence is prone the risk of bias.

A study that adopted the idea of comparing results when evaluating a model with full data against a similarity-reduced data was reported [4]. Three different databases were the source of fifteen constructed datasets, five reduced datasets for each specific-length allele downloaded from each database. Prediction was carried out using three different techniques and results are recorded on several MHC class II alleles. Unfortunately, their datasets lacks collectivity as data downloaded from three databases are not put together but rather used separately, which is likely to decrease amount of information needed for training the classifier. Furthermore, the study did not involve feature selection, which leads to the problem of higher data dimensionality and its computation time and memory complications.

A study was carried out to demonstrate the effect of using forty-two different combinations of six features extracted from the Chou's PseAAC [15] along with Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). Datasets of allele HLA-DRB1*0301 constructed in [4] are used for evaluation in a 5-fold cross-validation procedure. The importance of using balanced data through Synthetic Minority Oversampling Technique (SMOTE) was emphasized and reported to enhance prediction performance [3]. Unfortunately, same previously stated drawbacks were observed with absence of blind testing, use of only one database, and use of six features with no feature selection. A review study concentrating on the tools available through the Immune Epitope Database and analysis Resource [16] was held out to highlight the importance of the practical use of those tools. Two case studies are investigated to figure out the immunogenicity of erythropoietin and timothy grass pollen [17].

Some tools were designed to serve both MHC class I and II binding peptides prediction. These tools are mainly characterized by their independence on the peptide length where amino acid sequences were treated as time series data. This method overcame the drawback of peptides length-dependent machine learning techniques and offered an opportunity to predict uncommon length epitopes [18]. A recent study proposed a new classification algorithm using ℓ1-minimization techniques operated on a sparse representation of peptides. The research concluded that physicochemical properties encoding of features is preferred over binary encoding scheme. Prediction of peptides binding to five different alleles were examined using 10-fold cross-validation [19].

Analysis of previous work defines main prediction shortcomings that will be addressed in this work by enhancing each step of the process. First, the data collection is done using three most recent updated epitope databases. Different datasets are then constructed to decide which is more appropriate for training a classifier, full data or similarity reduced data. The effect of data resampling is monitored through the establishment of a balanced dataset. Second, feature selection techniques are utilized to automate choosing features based on well-defined criteria. Third, neuro-fuzzy classifier is used for classification given its excellent performance in similar problems while incorporating a speed up scaled conjugate-gradient algorithm to boost their speed. Finally, the evaluation of the proposed model is performed using different measurement metrics. A comparison of the results from cross-validation with blind testing is performed for the developed system. In contrast to previous work that carried out blind testing on unpublished datasets that most probably contain similar peptides to the ones used in training, we ensure removing any similar peptides in the testing set for blind testing. The new system has the potential to provide more robust prediction results by eliminating all sources of bias in the evaluation process, which is important for objective evaluation.

3. Methodology

The objective of this paper is to propose a classification model to predict which peptides within a group of sequences are able to bind MHC class II molecules. An initial influential step is to take a decision concerning which peptides participate in the training phase of the classification algorithm. Since chosen peptides significantly affect the classification model, this stage is thoroughly studied by comparing the model performance when trained with five mixed datasets formed from three downloaded datasets. Two datasets are subjected to similarity reduction based on an optimal local alignment procedure. Fig. (1) illustrates all the data aggregation, filtering, processing and splitting steps.

Classifying peptides into binders and non-binders using a machine learning technique needs informative features. Therefore, features must be first collected and calculated to have a global peptides representation. Then, feature selection techniques must be implemented to choose the most informative features. As a final step, a neuro fuzzy classifier with three distinct implementations classifies the peptides using datasets with the assigned features.

3.1. Data aggregation and processing

The most common web available databases that contain information about binding capabilities of peptides to MHC class II molecules are; Immune Epitope DataBase (IEDB) [16], MHCBN [5] [20], SYFPEITHI [21], MHCPEP [22] and AntiJen [23]. Last update for AntiJen database was in 2003 and that for MHCPEP was in 1998. Therefore, the data available through these two databases are not included in the data collected for our study. That is because correctness cannot be guaranteed and there is a possibility that the state of any of the peptides is changed later. Furthermore, MHCPEP is one of the data sources of the MHCBN database (last updated 2009). On the other side, IEDB is characterized by being the largest container for MHC molecules data and is the most frequently updated (last updated 2015). Also, SYFPEITHI database (last updated August 2012) contains quite a bit good list of peptides binding to some MHC alleles.

In accordance to the previous quick analysis, data used in this study will be a collection of binding and non-binding peptides to MHC class II alleles withdrawn from the three databases; IEDB, MHCBN and SYFPEITHI. These data are filtered first according to the specifications of its source database and then collected to form one dataset. IEDB data are filtered by; first, removing peptides with no reported IC50 value then omitting any peptide does not meet the database construction conditions. MHCBN data are filtered by removing peptides with no or uncertain qualitative values. SYFPEITHI data are not in a need to any filtration since it contains binding peptides only. This new collected dataset is further filtered by removing any repeated sequence such that the resulting dataset contains only unique peptides (FUD).

3.2. Similarity reduction by optimal local alignment

To obtain a similarity reduced dataset, FUD is then split into two sub-datasets; namely, binders and non-binders datasets. Clusters are formed for each sub-dataset such that each cluster contain all sequences sharing a similarity of 80% or more. Similarity is detected using optimal local alignment algorithm [24] by calculating a similarity score between each two sequences in the same sub-dataset. Optimal local alignment is a dynamic programming algorithm developed from that of Smith and Waterman [25]. Pairwise alignment is included along with a gap penalty value of 8 when executing this algorithm.

To guarantee a fair blind testing, 15% of each sub-dataset is set aside then merged to form a testing dataset. This is done under a constraint that clusters that contain only one peptide are the only allowed as a part of the testing dataset. This ensures that the testing dataset can never contain a peptide similar to any of the ones found in the training dataset. That is because a cluster with only one peptide means that there is no similar peptide in its sub-dataset (up to at least 80% similarity). A similarity reduced training dataset is a one that contains no two peptides share similarity in sequences more than 80%. This is achieved by taking only one peptide from each cluster. The chosen peptide is the one that share the maximum similarity with other peptides in the same cluster.

3.3. Feature representation

The variable length characteristic of the MHC class II molecules raises the importance of expressing peptides by a fixed length feature set. This ensures converting a variable length vector into a fixed length one to be computationally convenient for the classification process. Features contributing to the fixed length feature vector construction are extracted from the Amino Acid Index (AAindex) database [26]. AAindex is a huge repository of physicochemical properties of all amino acids expressed in numerical indices. AAindex contains 545 properties reduced to 531 after removing those appearing in undefined values at specific amino acids. The remaining properties are all used as features in the feature vector representation to have a feature vector for each peptide of size 1×531. Each peptide feature is an averaged value of each physicochemical property calculated by Eq. (1).

$$PFV = \frac{\sum_{1}^{L} AAPV}{L}\,, \tag{1}$$

where PFV is the feature value of each peptide, AAPV is the property value of each amino acid, and L is the peptide length. Therefore, for every single peptide, the previous equation is repeated 531 times to result in a 531 PFVs representing the peptide feature vector of size 1×531.

For example to calculate the feature value of the property named “Hydrophobicity index” for the 9-mer peptide sequence “'ARSMAAAAA'”, the hydrophobicity index value of each amino acid (AAPV) in the sequence is fetched from the AAindex database, then the 9 values are summed up and divided by L (9) to have one representative hydrophobicity index value of the mentioned sequence. Fig. (2) illustrates this numerical example. This process is repeated 531 times for all the amino acid physicochemical properties.

3.4. Feature selection

Since each peptide has a feature vector of 531 values, this results in a feature matrix of size n×531 where n is the total number of peptides in the dataset. This clearly poses a high dimensional data problem, which is challenging for the classification problem in addition to its computation time cost. Feature selection techniques are used to extract the most informative features to address this problem. Here, we use four filter-type feature selection techniques with both parametric and non-parametric tests in addition to a proposed hybrid technique. In particular, ranking features by t-test and entropy represent parametric tests while using receiver operating characteristics (ROC) and Wilcoxon calculations are non-parametric. For parametric tests, data are normalized along each feature around zero mean and unit variance. This is in addition to proposing the hybrid and the fuzzy filter feature selection techniques.

- The four filter feature selection techniques are; t-Test [27], Relative Entropy (RE) [28], ROC (Receiver Operating Characteristics) [29] and Wilcoxon Mann Whitney test [30]. For example, when calculating the values of each of these four tests on the 531 physicochemical properties (using FUD data), the “polar requirement” property had the highest entropy value which is equal to 0.65. Another property named “Knowledge-based membrane-propensity scale from 3D_Helix in MPtopo databases” had the highest t-test and ROC values which are equal to 28.4 and 0.225 respectively. As for the Wilcoxon test, the “Averaged turn propensities in a transmembrane helix” property had the highest rank with a value equals to 0.5.
- Hybrid filter chooses the best informative highest scoring feature according to each of the four previous techniques to form a feature vector of size 1×4 for each peptide. Since four filter techniques are applied, the hybrid filter

chooses the first ranked feature from each of these four. As a result, four features only are used in the hybrid technique. If any feature is repeated among the chosen ones, the algorithm skips it to the next unrepeated feature to ensure having four different ones. The hybrid filter technique diagram is shown in Fig. (3).

- Fuzzy ranking [31] selects features using powers of fuzzy sets expressed by their linguistic hedge values. Adaptive neuro fuzzy classifiers are used to define classification fuzzy sets whose linguistic hedge values describe the importance of features. A feature is considered informative if its corresponding linguistic hedge value of classes is greater than 0.5 and more close to 1. Otherwise, features are omitted from the informative features list.

3.5. Neuro-fuzzy classifier (NFC)

A neuro-fuzzy classifier is a hybridization between fuzzy systems and neural networks. Such type of merged technique has the ability to produce decisions across their built-up fuzzy rules with their membership functions tuned by neural network. The NFC algorithm defined in our employed implementations are based on the zero-order Sugeno fuzzy model shown in Fig. (4) [32]. The model rule is stated as; *if x is A and y is B then z = C*, where; *x* and *y* are the input variables, *A* and *B* are the antecedent fuzzy sets, and *C* is the class to which *z* belongs. This model comprises five layers where, layer 1 nodes always have outputs specifying the degree of satisfaction (membership grade) between the node and its linguistic label. Each node in Layer 2 (rule node) represents a rule with an output expressing the degree of fulfillment of that rule (firing strength). Normalized firing strengths are the output of Layer 3 which is known as the normalization layer. Normalized firing strengths are multiplied by each individual rule to give the output of layer 4.

Finally, all incoming outputs of layer 4 are summed up in layer 5 to give the model output [32] [33].

Three adaptive neuro-fuzzy classifiers are utilized based on partial modification of Bayram's classifiers [11] [12]. The original implementations were shown to have good performance on medium and large-scale data. However, they have not been applied to our type of data before [34] [35]. All three algorithms initialize their fuzzy rules by clustering using K-means where number of clusters per class is user dependent. Also, they share the same fuzzy sets description method which is based on Gaussian membership function. Number of clusters are set to 10 for a number of epochs of 50 when examining datasets and feature selection techniques. Then, number of clusters and epochs are changed to study their effect on the chosen dataset and feature selection techniques. A description of the three adopted NFC methods and the proposed modification is as follows:

- Scaled conjugate-gradient Neuro-Fuzzy Classifier (SCG-NFC) [11]: SCG algorithm possess an acceptable convergence rate and low memory usage when training neuro-fuzzy classifiers. SCG stands on the second-order gradient supervised learning procedure. A combined trust-region method eliminates the step size calculation learning time problem of the line-search method. SCG-NFC is executed when comparing the effect of different datasets and feature selection techniques.
- Speedup SCG-NFC (SSCG-NFC) [11]: Gradients of the SCG are calculated twice for each iteration to each training instance. A method to further decrease time while preserving convergence rate is to use gradients estimation instead of calculation. This method allows decreasing computation time by 20 to 50% when applied to different applications.

- Power of Fuzzy sets-SCG-NFC [12]: PF-SCG-NFC is a modification of the SCG-NFC that enhances the recognition rate and positively contribute to resolving overlapped classes misleading issue. Linguistic hedges of the power of fuzzy sets are proposed to add one more layer of the underlying network and is trained with other network parameters.
- Integrated K-means modification: K-means clustering coded in Bayram classifiers initialize its centroids randomly. Consequently, multiple processing classify instances differently according to the final built-up clusters that always depend on the initial cluster centroids. Alternatively, we propose uniform choice of centroids based on sorting the summation of all features for each instance. Then, sample points are chosen starting from the smallest until the largest with a calculated step. The step is determined from the data size and the number of clusters.

4. Experimental Verification

Here, the detailed evaluation of the proposed model is presented. One of the main points under study is the effect of training a classification model with different datasets and choosing a reliable evaluation strategy. Therefore, this work aims to answer two important questions. The first is about the type of data to be used to train a classifier. Data downloaded from a database can be utilized in two different forms, either to consider removing repeated peptide sequences as sufficient or opting to continue to have similarity-reduced data. The second important issue is about the evaluation strategy that is able to express results in more realistic results and hence more robust assessment of the methodology. Therefore, comparison of model evaluation using cross-validation against blind testing is performed.

4.1. Evaluation criteria

Four evaluation metrics are recorded to assess the model performance on predicting binding peptides to human MCHII molecule HLA-DRB1*0101; namely, area under receiver operating characteristics curve (AUC), accuracy, sensitivity and specificity. AUC differs from accuracy in that the result of the former depends on all thresholds discriminating between the two classes. Whereas, the later gives a result using one threshold value (cut-off point) detected by the classifier. Sensitivity and specificity are mainly used to figure out the model performance on the positive (binders) and negative (non-binders) instances separately. All evaluation metrics are recorded for each of the six implemented feature selection techniques on five datasets. Table (1) lists details of datasets construction of the HLA under investigation.

- FUD: Filtered Unique Dataset is a filtered collection of binders and non-binders extracted from the three previously chosen databases.
- SRD: Similarity Reduced Dataset is constructed from FUD by removing peptides sharing a similarity of 80% or more.
- TrFUD-R: Training FUD is the result of randomly dividing FUD into 85% training data and 15% testing data constitutes the Testing Dataset (TestD-R).
- TrFUD-C: Training FUD by Clustering is a training dataset of 85% of FUD but without any similar peptide to the ones found in the 15% forming the Testing Dataset (TestD-C).
- TrSRD-C: Training Similarity Reduced Dataset by Clustering is derived from TrFUD-C by keeping only one peptide from a set of peptides sharing 80% or more similarity. Its testing data (TestD-C) is the same of TrFUD-C to compare the effect of training a classifier with all available peptides against similarity reduced peptides.

- BSRD: Balanced Similarity Reduced Dataset is constructed to eliminate the unbalancing data property. Down sampling of the majority class (binders) results in nearly equal number of peptides to those of the minority class (non-binders). Similarity clustering is the proposed resampling criteria where only one peptide is captured from a set of peptides that share sequence similarity of 60% or more. Fig. (5) shows its construction steps.

The first two datasets are the used ones when evaluating the proposed model using K-fold cross-validation that creates disjointed evaluation sets with a K value of 5. Cross-validation with K of 10 is only carried once on FUD to show the variability in accuracy on different testing folds. The next three datasets are the blind testing datasets used to differentiate between random and planned choices of testing data. The last dataset evaluates the three classifiers' implementations change in performance due to enrolling a balanced dataset on a 5-fold cross-validation basis.

5. Results and Discussion

Proposed work through this study involved several steps that should be assessed separately to have a fair evaluation for each. Section 5.1 aims to provide a comprehensive feedback on the effect of using similarity-reduced data by its comparison with the usage of unique peptides data. The preference of using blind testing over cross-validation is explained in section 5.2. Comparing the performance of the employed feature selection techniques and detecting the appropriate number of selected features is the objective of section 5.3. Section 5.4 displays the results of the three classifier implementations in addition to evaluating them on the BSRD.

5.1. Comparison of similarity-reduced data against raw data

The performance of the classification model when using similarity reduced datasets was compared to full unique peptides dataset to decide the choice of dataset construction criteria. Tables (2)-(7) record the values of all the evaluation metrics of the five datasets for the six feature selection techniques at constant classifier parameters. Random choice of testing data (TestD-R) AUC values were the highest for five out of six feature selection techniques. The other three measurements were dependent on the features ranking method choice as its highest value differs from one to another. That observation meets the proposed idea that training without similarity reduction gives overly optimistic results. The reason behind is TestD-R (as it is randomly selected) may contain peptides similar in sequences to any in TrFUD-R. A situation which despite being more close to reality does not give a reliable model assessment.

SRD is extracted from FUD after removing all similar peptides. FUD showed higher three measurements (AUC, accuracy and sensitivity) than that of SRD for all feature selection techniques. Specificity is always lower except that of hybrid filter which was a little bit higher (1%). This cannot be interpreted as FUD being better than SRD when evaluating classifiers because of the similarity and the randomness choice of data problems. Again, TrSRD-C is compared to TrFUD-C as the former contains peptides of the later with similar ones removed. Specificity of TrSRD-C showed a significant increase in four of the feature selection techniques (reached 8%) on the account of decrease in sensitivity (5%). The reason behind this is that reduction by 27% in the binders dataset corresponds to only 19% in the non-binders dataset. TrFUD-C and TrSRD-C AUC and accuracy results are mostly comparable. TrSRD-C had the highest AUC in most cases while TrFUD-C accuracy values were better. To make a decision about the dataset to use, we choose TrSRD-C to have a sort of balanced results according to its specificity and AUC values.

5.2. Comparison of cross-validation against blind testing

To compare between cross-validation and blind testing results, an average AUC value of SRD and FUD (cross-validation datasets) is compared against that of TrFUD and TrSRD across all feature selection techniques in Fig. (6). TrFUD-R results are excluded from the comparison to have an ultimate fair blind testing. Results demonstrate that cross-validation results are always higher due to random choice of data shared in its folds causing bias. On the contrary, blind testing with clustering testing dataset did not contain any similar peptides and hence provide more robust, bias-free results.

To understand the effect of randomness, the results of applying 10 fold cross-validation on FUD are presented in Fig. (7). This figure shows the big variance in the model performance on each testing fold of the specified 10 folds. Each fold is a random choice contributing a 10% of the original dataset. Testing accuracy values varies from 67% to 74% keeping all classification model parameters fixed and only peptides forming training and testing data are changed. This gives an indication of the dependency of the model performance when evaluating using cross-validation on random split of data. On the other hand, blind testing gives more realistic results. This indicates that cross-validation cannot serve as realistic model evaluation.

5.3. Feature selection

The model proposed in this work is first trained by five different combinations of the downloaded datasets. Each dataset is represented by a group of features chosen by six feature selection techniques. The most repeated features names and code extracted from the AAindex database with a letter given as abbreviation for further listing simplification is listed in Table (8). Table (9) mentions the top significant feature picked up by each technique for every dataset. The most repeated feature is the polar requirement (Woese, 1973). This feature is chosen by the entropy and fuzzy tests for all the datasets and so it is a basic part of the hybrid selection. While t-test had the same selected feature, ROC and Wilcoxon tests change selection two times for the five datasets. Therefore, parametric tests (t-test and entropy) were not affected by removing similar peptides from the datasets. Non-parametric tests (ROC and Wilcoxon) ranking were subject to change when data construction is varied.

By analyzing measurement values for each filter selection technique, we found that none of the six feature selection techniques is considered the best due to varying results with data. But, ranking by hybrid and fuzzy tests values is seen to be the best performing on clustered datasets. Thus, their selected features values will assemble the features vectors for the rest of evaluation. What is really was interesting that the four hybrid filter chosen features are enclosed within the some of the other feature selection tests chosen ones. Accordingly, effect of changing the number of top ranked features constituting feature vectors is visualized in Fig. (8). The results showed that the number of features is not directly proportional to the performance values. Thus, different features number ought to be under trial first to detect the best suitable composition for a specific classifier.

5.4. Classifier performance

A first step was to show the effect of changing number of clusters and epochs on the classifier assessment. Fig. (9) displays the values of the evaluation metrics assuming different numbers of clusters in the SCG-NFC design one time by trying 30 epochs and another time by 10. Number of epochs have no big influence while, increasing number of clusters above a certain limit mostly reduces AUC values. Best AUC value happened at 10 clusters and 30 epochs (10/30) and so these are parameter values when evaluating the difference in performance of the three classifiers shown in Table (10). To ensure the comparison output, we changed the number of clusters to be 6 and number of epochs to be 50 (6/50) and documented the results in the same table. SCG-NFC had the highest AUC, accuracy and sensitivity for both cases. SSCG-NFC showed comparable results with characteristic of reducing time to the third. PF-SCG-NFC always had the smallest AUC for these two cases.

The effect of the three classifiers is further examined on the balanced dataset (BSRD) which was constructed specifically for this purpose after noticing the specificity low values in different stages. Results when using the same previous two cases of epochs and clusters numbers are present in Table (11). Balanced data had a great effect on the specificity values for the three classifier implementations (8% increase from the highest achieved before). Sensitivity was negatively affected by down-sampling of data which in turn affects accuracy values. Then, SCG-NFC outperforms PF-SCG-NFC in most cases which contradicts its proposed idea of increasing performance for our type of data.

6. Conclusions

In this study, we constructed a robust MHC class II peptides neuro-fuzzy classification system. We outlined the effect of different parameters on the performance of a model designed to classify various peptides into MHC class II binders and non-binders. Datasets of HLA-DRB1*0101 were used to compare and evaluate different model stages including dataset construction, feature selection and representation, classifier modeling, and evaluation criteria. Similarity reduced datasets were found to be the most appropriate for classification due to treatment of the over-fitting and overly optimistic issues. Variable length peptides were mapped into fixed length feature vectors where features are automatically selected according to their ranking scores. Fuzzy selection and integrating the top ranked by different tests into a hybrid one were shown to be the best feature selection techniques. Comparing results from cross-validation and blind testing methods was performed and indicated that blind testing eliminates the bias in the results that result from similarity of data in cross-validation. Three implementations of adaptive NFC were developed. Among these methods, SSCG-NFC reduced computation time by 30% with comparable classification performance. The developed system outlines the importance of eliminating similarity in data sets and bias in evaluation measures in order to maintain robustness and objective assessment of methodology.

References

[1] J. C. Tong and S. Ranganathan, "MHC and T cell responses," in *Computer-Aided Vaccine Design*, Philadelphia, Woodhead Publishing Limited, 2013, pp. 1-12.

[2] H. Takahashi and H. Honda, "Prediction of Peptide Binding to Major Histocompatibility Complex Class II Molecules through Use of Boosted Fuzzy Classifier with SWEEP Operator Method," *Journal of Bioscience and Bioengineering,* vol. 101, no. 2, pp. 137-141, 2006.

[3] F. K. Faramarzi, M. M. Beigi, Y. Botorabi and N. Mousavi, "Prediction of Peptides Binding to Major Histocompatibility Class II Molecules Using Machine Learning Methods," *Engineering,* vol. 5, pp. 513-517, 2013.

[4] Y. M. El-manzalawy, D. Dobbs and V. Honavar, "On Evaluating MHC-II Binding Peptide Prediction Methods," *PLoS One,* vol. 3, no. 9, p. e3268, 2008.

[5] M. Bhasin, H. Singh and G. P. S. Raghava, "MHCBN: a comprehensive database of MHC binding and non-binding peptides.," *BIioinformatics,* vol. 19, no. 5, p. 665–666, 2003.

[6] F. Castellino, G. Zhong and R. N. Germain, "Antigen presentation by MHC class II molecules: invariant chain function, protein trafficking, and the molecular basis of diverse determinant capture," *Hum Immunol,* vol. 54, p. 159–169, 1997.

[7] Y. Saeys, I. Inza and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, no. 19, pp. 2507-2517, 2007.

[8] T. Neideen and K. Brasel, "Understanding Statistical Tests," *Journal of Surgical Education,* vol. 64, no. 2, pp. 93-96, 2007.

[9] D. Nauck and R. Kruse, "What are Neuro-Fuzzy Classifiers?," in *Proc. Seventh International Fuzzy Systems Association World Congress IFSA'97*, Pregue, 1997.

[10] M. V. Ribeiro, C. A. Duque and J. M. T. Romano, "An interconnected type-1 fuzzy algorithm for impulsive noise cancellation in multicarrier-based power line communication systems," *IEEE J Sel Areas Communitications,* vol. 24, no. 7, p. 1364–1376, 2006.

[11] B. Cetisli and A. Barkana, "Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training," *Soft Comput,* vol. 14, p. 365–378, 2010.

[12] B. Cetisli, "Development of an adaptive neuro-fuzzy classifier using linguistic hedges: Part 1," *Expert Systems with Applications,* vol. 37, no. 8, p. 6093–6101, 2010.

[13] M. Nielsen, O. Lund, S. Buus and C. Lundegaard, "MHC Class II epitope predictive algorithms," *Immunology,* vol. 130, p. 319–328, 2010.

[14] H. Takahashi and H. Honda, "A new reliable cancer diagnosis method using boosted fuzzy classifier with a SWEEP operator method," *Journal of chemical engineering of Japan,* vol. 38, pp. 763-773, 2005.

[15] H.-B. Shen and K.-C. Chou, "PseAAC: a flexible web-server for generating various kinds of protein pseudo amino acid composition," *Analytical Biochemistry,* vol. 373, pp. 386-388, 2008.

[16] "Immune Epitope Database and Analysis Resource," National Institute for Allergy and Infectious Diseases (NIAID), February 2006. [Online]. Available: http://www.iedb.org/. [Accessed 4 March 2015].

[17] S. Paul, R. V. Kolla, J. Sidney, DanielaWeiskopf, W. Fleri, Y. Kim, B. Peters and A. Sette, "Evaluating the Immunogenicity of Protein Drugs by Applying In Vitro MHC Binding Data and the Immune Epitope Database and Analysis Resource," *Clinical and Developmental Immunology,* pp. 1-7, 2013.

[18] C. Meydan, H. H. Otu and O. U. Sezerman, "Prediction of peptides binding to MHC class I and II alleles by temporal motif mining," *BMC Bioinformatics,* vol. 14, no. 2, pp. S2-S13, 2013.

[19] C. Aguilar-Bonavides, R. Sanchez-Arias and C. Lanzas, "Accurate prediction of major histocompatibility complex class II epitopes by sparse representation via ℓ1-minimization," *BioData Mining,* vol. 7, no. 23, 2014.

[20] "MHCBN version 4.0," Dr Raghava's Group, Bioinformatics Centre, Institute of Microbial Technology, Chandigarh, INDIA, 2009. [Online]. Available: http://www.imtech.res.in/raghava/mhcbn. [Accessed 2 March 2015].

[21] "SYFPEITHI : A Database Of MHC Ligands And Peptide Motifs (Ver. 1.0)," Institute for Cell Biology,Department of Immunology, 27 August 2012. [Online]. Available: http://www.syfpeithi.de/. [Accessed 5 March 2015].

[22] V. Brusic, G. Rudy, A. P. Kyne and L. C. Harrison, "MHCPEP—A Database of MHC-Binding Peptides: Update 1995," *Nucleic Acids Research,* vol. 24, no. 1, pp. 242-244, 1996.

[23] "AntiJen: A Kinetic, Thermodynamic and Cellular Database v2.0," The Edward Jenner Institute for Vaccine Research, 2003. [Online]. Available: http://www.ddg-pharmfac.net/antijen/AntiJen/antijenhomepage.htm. [Accessed 1 March 2015].

[24] G. J. Barton, "An Efficient Algorithm to Locate All Locally Optimal Alignments Between Two Sequences Allowing for Gaps," *Computer Applications in the Biosciences,* vol. 9, pp. 729-734, 1993.

[25] T. F. Smith and M. S. Waterman, "Identification of Common Molecular Subsequences," *Journal of Molecular Biology,* vol. 147, no. 1, p. 195–197, 1981.

[26] "AAindex," Kyoto University Bioinformatics Center, 31 March 2008. [Online]. Available: http://www.genome.jp/aaindex/. [Accessed 15 March 2015].

[27] Student, "The Probable Error of a Mean," *Biometrika,* vol. 6, no. 1, pp. 1-25, 1908.

[28] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics,* vol. 22, no. 1, pp. 79-86, 1951.

[29] A. J. Serrano, E. Soria, J. D. Martin, R. Magdalena and J. Gomez, "Feature selection using ROC curves on classification problems," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, 2010.

[30] H. B. Mann and D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *Annals of Mathematical Statistics ,* vol. 18, no. 1, pp. 50-60, 1947.

[31] B. Cetisli, "The effect of linguistic hedges on feature selection: Part 2," *Expert Systems with Applications,* vol. 37, no. 8, p. 6102–6108, 2010.

[32] J.-S. R. Jang, C.-T. Sun and E. Mizutani, "Chapter 4: Fuzzy Inference System," in *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997, pp. 73-90.

[33] Y. Wu, B. Zhang, J. Lu and K. -L. Du, "Fuzzy Logic and Neuro-fuzzy Systems: A Systematic Introduction," *International Journal of Artificial Intelligence and Expert Systems,* vol. 2, no. 2, pp. 47-80, 2011.

[34] J. C. Tong and S. Ranganathan, "Computational T cell vaccine design," in *Computer-aided vaccine design*, Woodhead Publishing Limited, 2013, pp. 59-86.

[35] L. P. Eng, T. W. Tan and J. C. Tong, "Chapter 4: Building MHC Class II Epitope Predictor Using Machine Learning Approaches," in *Computational Peptidology*, New York, Springer, 2015, pp. 67-73.

Table (1): Number of peptides for HLA-DRB1*0101 starting at download from different databases passing by each separate step. Step1 refers to removing peptides with no reported IC50 value and step2 refers to omitting any peptide does not meet the database construction conditions.

All Downloaded Data for Binders and non-Binders

IEDB			***MHCBN***		***SYFPEITHI***
All	Step1	Step2	All	Filtered	All (binders only)
9834	8231	7299	588	575	21

Total after filtering and collecting in one dataset

7299+575+21=7895

Dataset		Total	Binders	Non-binders
FUD		7571	5253	2318
SRD	Tot. clust.	6012	4067	1945
	1 pep clust.	5022	3357	1665
TrFUD		6435	4465	1970
TrSRD		4876	3279	1597
TestD-C		1136	788	348
BSRD		4815	2497	2318

Table (2): HLA-DRB1*0101 performance values recorded on different testing procedure using the top 10 results of the t-test based feature selection (50 epochs and 10 clusters for SCG-NFC)

Eval.	*Testing Data*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*
CV (K=5)	FUD	0.66	0.75	0.89	0.43
	SRD	0.64	0.70	0.83	0.46
Blind test	TestD-R	0.66	0.74	0.90	0.41
	TestD-C (TrFUD-C)	0.64	0.73	0.87	0.41
	TestD-C (TrSRD-C)	0.64	0.71	0.82	0.45

Table (3): HLA-DRB1*0101 performance values recorded on different testing procedure using the top 10 results of the entropy based feature selection.

Eval.	*Testing Data*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*
CV (K=5)	FUD	0.65	0.75	0.89	0.41
	SRD	0.64	0.72	0.85	0.43
Blind test	TestD-R	0.68	0.74	0.88	0.48
	TestD-C (TrFUD-C)	0.62	0.71	0.84	0.41
	TestD-C (TrSRD-C)	0.65	0.72	0.84	0.47

Table (4): HLA-DRB1*0101 performance values recorded on different testing procedure using the top 10 results of the wilcoxon based feature selection.

Eval.	*Testing Data*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*
CV (K=5)	FUD	0.65	0.75	0.90	0.40
	SRD	0.65	0.73	0.87	0.43
Blind test	TestD-R	0.66	0.73	0.88	0.44
	TestD-C (TrFUD-C)	0.60	0.66	0.75	0.45
	TestD-C (TrSRD-C)	0.61	0.65	0.71	0.51

Table (5): HLA-DRB1*0101 performance values recorded on different testing procedure using the top 10 results of the roc based feature selection.

Eval.	*Testing Data*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*
CV (K=5)	FUD	0.67	0.75	0.89	0.44
	SRD	0.66	0.74	0.87	0.46
Blind test	TestD-R	0.66	0.73	0.87	0.46
	TestD-C (TrFUD-C)	0.62	0.68	0.78	0.46
	TestD-C (TrSRD-C)	0.63	0.72	0.85	0.42

Table (6): HLA-DRB1*0101 performance values recorded on different testing procedure using the hybrid feature selection.

Eval.	*Testing Data*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*
CV (K=5)	FUD	0.65	0.73	0.87	0.43
	SRD	0.64	0.72	0.86	0.42
Blind test	TestD-R	0.66	0.74	0.89	0.44
	TestD-C (TrFUD-C)	0.65	0.73	0.86	0.44
	TestD-C (TrSRD-C)	0.65	0.73	0.86	0.44

Table (7): HLA-DRB1*0101 performance values recorded on different testing procedure using the fuzzy feature selection.

Eval.	*Testing Data*	*AUC*	*Accuracy*	*Sensitivity*	*Specificity*
CV (K=5)	FUD	0.67	0.74	0.85	0.49
	SRD	0.66	0.72	0.84	0.48
Blind test	TestD-R	0.67	0.74	0.89	0.44
	TestD-C (TrFUD-C)	0.65	0.72	0.83	0.47
	TestD-C (TrSRD-C)	0.67	0.71	0.78	0.55

Table (8): List of the first top ranked features according to all the employed feature selection techniques. Features names and codes are those used by AAindex database. The last column contains an abbreviation letter to be easily used in the next table. In addition to the number of repetitions of each between brackets for all feature selection techniques excluding the hybrid one.

Name	*Code*	*Abbrev.*
Polar requirement (Woese, 1973)	WOEC730101	W (10)
Knowledge-based membrane-propensity scale from 3D_Helix in MPtopo databases	PUNT030102	P (8)
Averaged turn propensities in a transmembrane helix (Monne et al., 1999)	MONM990201	M (4)
AA composition of MEM of multi-spanning proteins (Nakashima-Nishikawa, 1992)	NAKH920108	N (3)
Negative charge (Fauchere et al., 1988)	FAUJ880112	F (3)
Membrane-buried preference parameters (Argos et al., 1982)	ARGP820103	A (1)
Principal component I (Sneath, 1966)	SNEP660101	S (1)

Table (9): The most significant feature as specified by each filter technique when ranking features using different datasets. In case the first top ranked feature in a test is recognized by another test for the same dataset, the next two top ranked are shown in order between brackets.

	t-Test	*Entropy*	*Wilcoxon*	*ROC*	*Hybrid*	*Fuzzy*
FUD	P (W, N)	W	M	P	P,W,M,N	W, F
SRD	P (W, N)	W	A	P	P,W,F,A	W, S
TrFUD-R	P (W, N)	W	M	P	P,W,M,N	W
TrFUD-C	P	W	M	F	P,W,M,F	W
TrSRD-C	P	W	M	F	P,W,M,F	W

Table (10): Performance of the three NFC implementations trained with TrSRD-C and tested over TestD-C. Top ten features selected by the Fuzzy feature selection represents the feature vectors. The first column indicate number of clusters and number of epochs of the classifier in the form (clusters/epochs)

	Classifier	*AUC*	*Acc.*	*Sens.*	*Spec.*	*Time*
10/30	SCG-NFC	0.67	0.72	0.79	0.55	3.79
	PF-SCG-NFC	0.64	0.70	0.79	0.49	5.94
	SSCG-NFC	0.64	0.70	0.79	0.50	1.96
6/50	SCG-NFC	0.66	0.71	0.80	0.52	5.94
	PF-SCG-NFC	0.66	0.71	0.80	0.52	19.23
	SSCG-NFC	0.65	0.70	0.76	0.54	1.89

Table (11): Performance of the three NFC implementations trained and tested over the balanced data BSRD on a 5-fold cross-validation basis. Feature vectors representation used the most informative ten features according to the Hybrid feature selection.

	Classifier	*AUC*	*Acc.*	*Sens.*	*Spec.*	*Time*
10/30	SCG-NFC	0.66	0.66	0.70	0.62	2.1
	PF-SCG-NFC	0.66	0.65	0.69	0.63	6.97
	SSCG-NFC	0.65	0.65	0.68	0.61	1.49
6/50	SCG-NFC	0.66	0.66	0.71	0.61	2.27
	PF-SCG-NFC	0.67	0.67	0.71	0.62	6.09
	SSCG-NFC	0.67	0.67	0.72	0.62	1.36

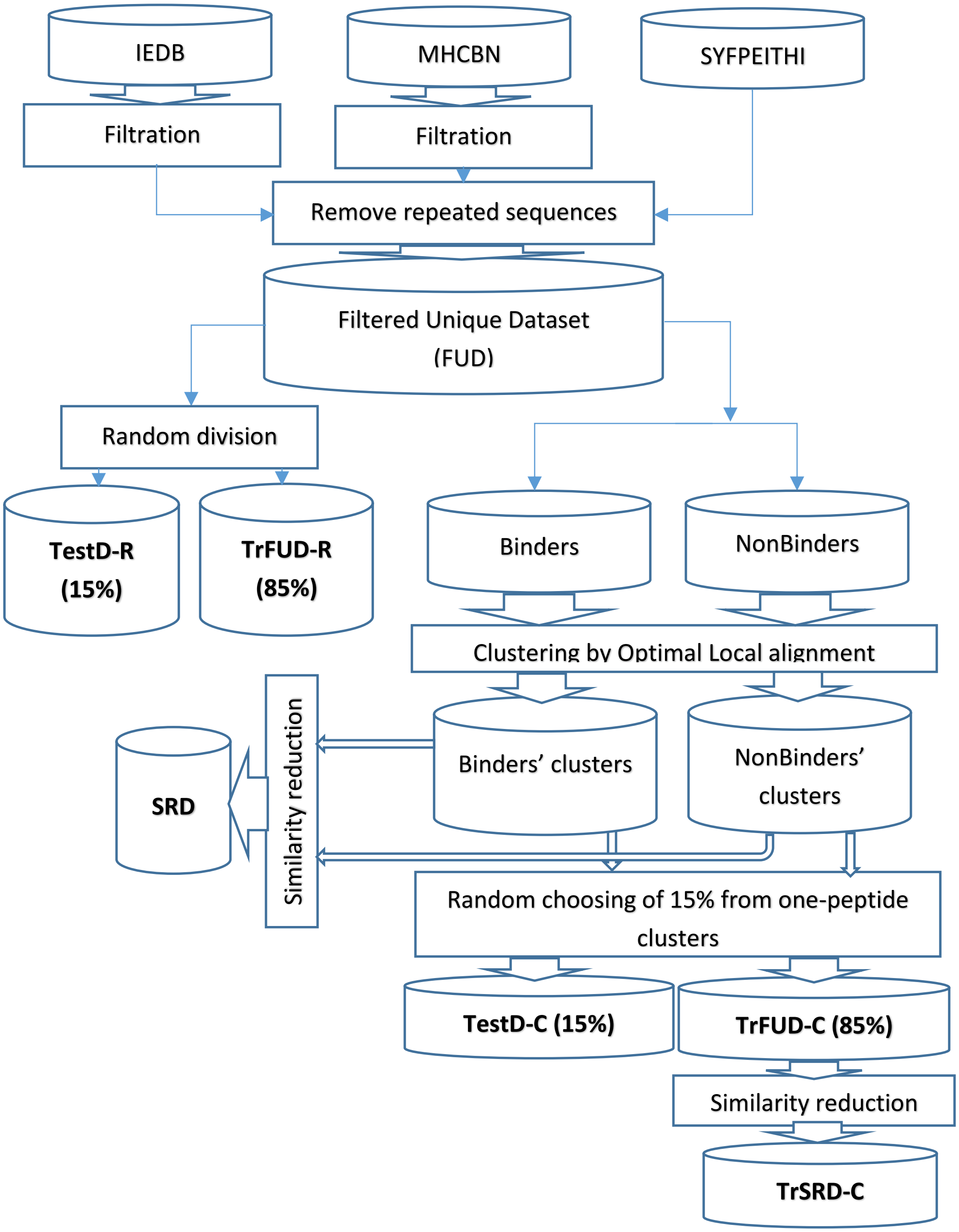
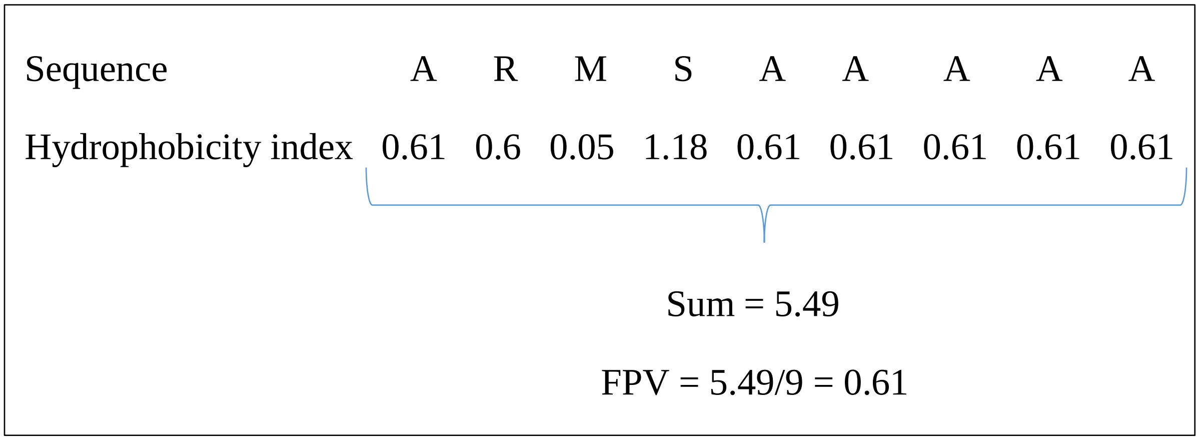


Fig. (1): Datasets aggregation, filtration, processing and splitting workflow.



Sequence	A	R	M	S	A	A	A	A	A
Hydrophobicity index	0.61	0.6	0.05	1.18	0.61	0.61	0.61	0.61	0.61

Sum = 5.49

FPV = 5.49/9 = 0.61

Fig. (2): An illustration of a numerical example on calculating a feature value of a peptide sequence

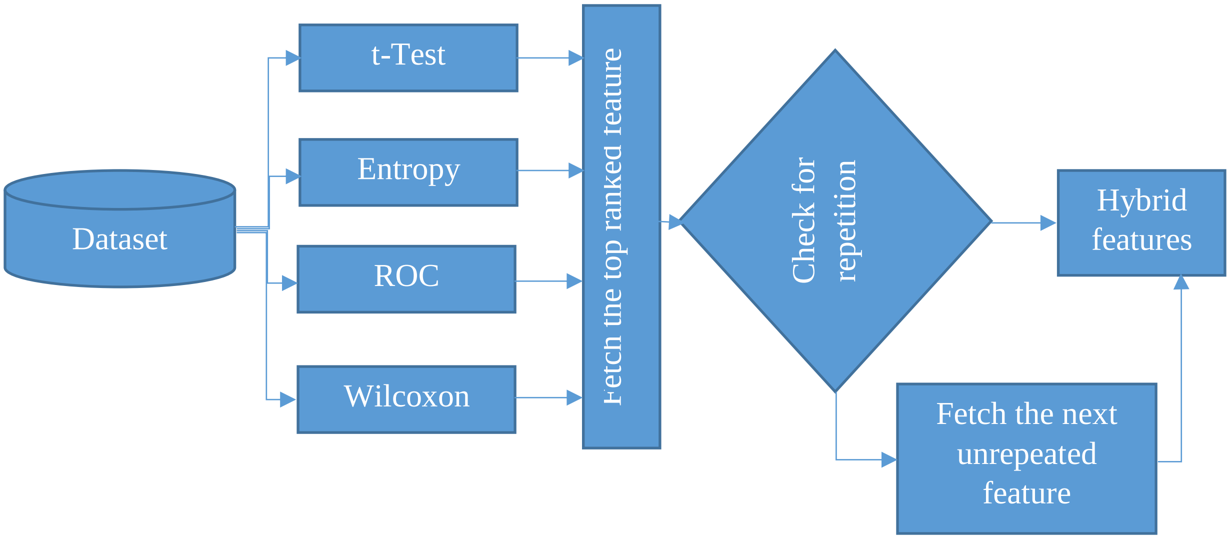


Fig. (3): The Hybrid Technique Block Diagram

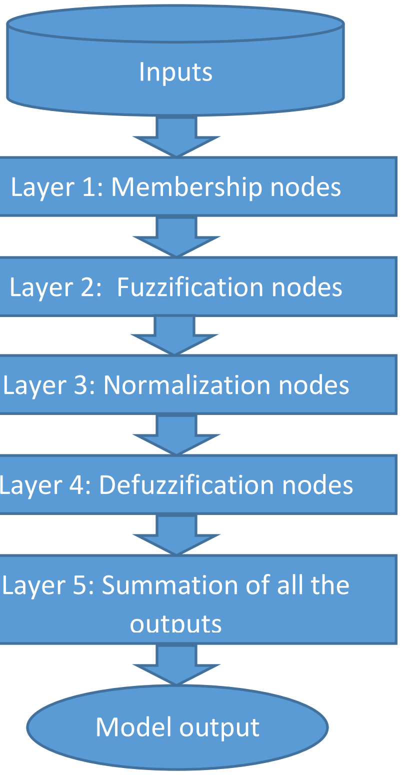

Inputs
Layer 1: Membership nodes
Layer 2: Fuzzification nodes
Layer 3: Normalization nodes
Layer 4: Defuzzification nodes
Layer 5: Summation of all the outputs
Model output

Fig. (4): Block diagram of the NFC model

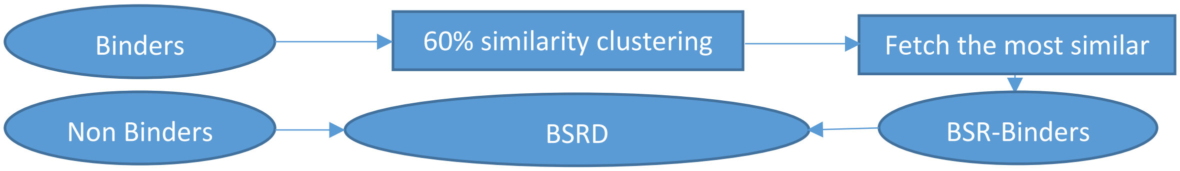


Fig. (5): Balanced Similarity Reduced Dataset construction

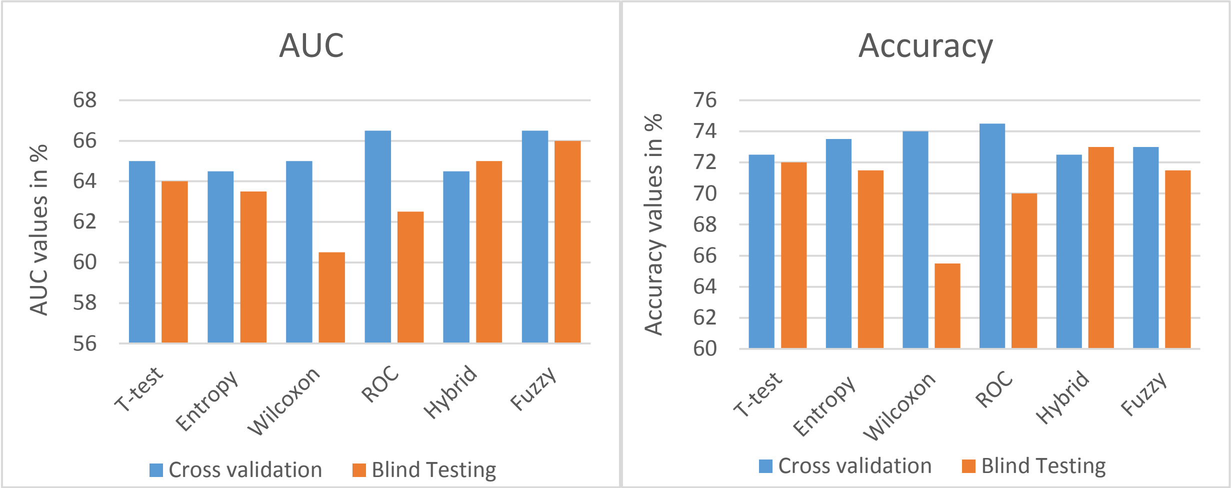


Fig. (6): AUC and accuracy values averaged over each two datasets contributing in cross-validation and blind testing comparison. Values are extracted from tables (2)-(7) to simplify the results analysis. Cross-validation results always exceeds blind testing results by 0.5% to 4.5% for AUC and by 0.5% to 8.5% for accuracy except for the hybrid feature selection technique.

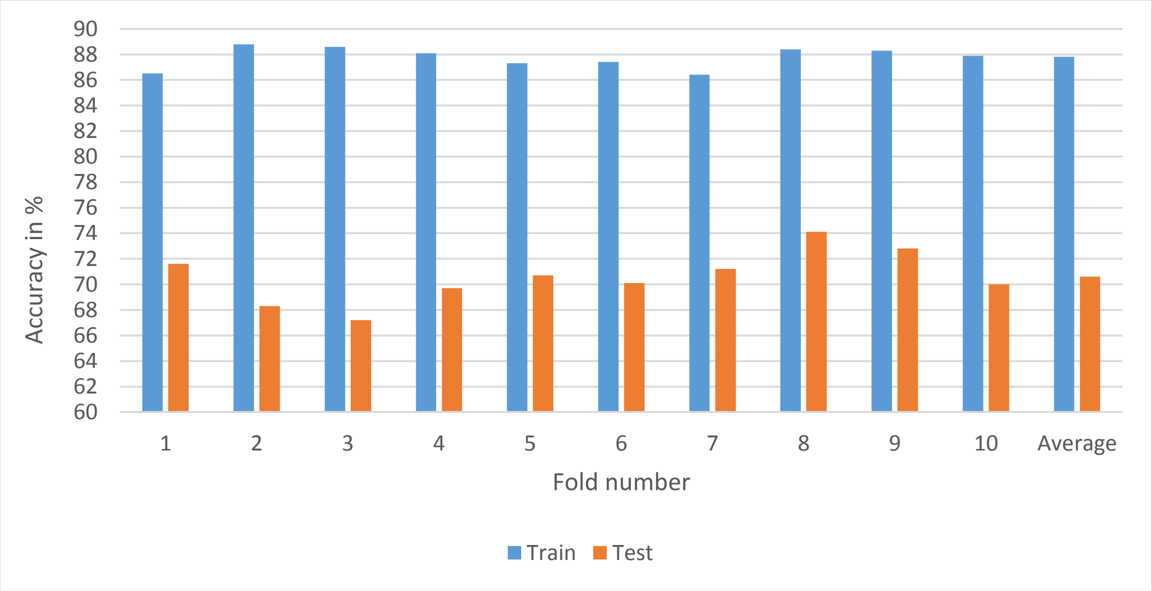


Fig. (7): Accuracy values on the training and testing folds on a 10-fold cross-validation testing using entropy for feature selection and FUD as the dataset. The horizontal axis is only an indication of the fold number.

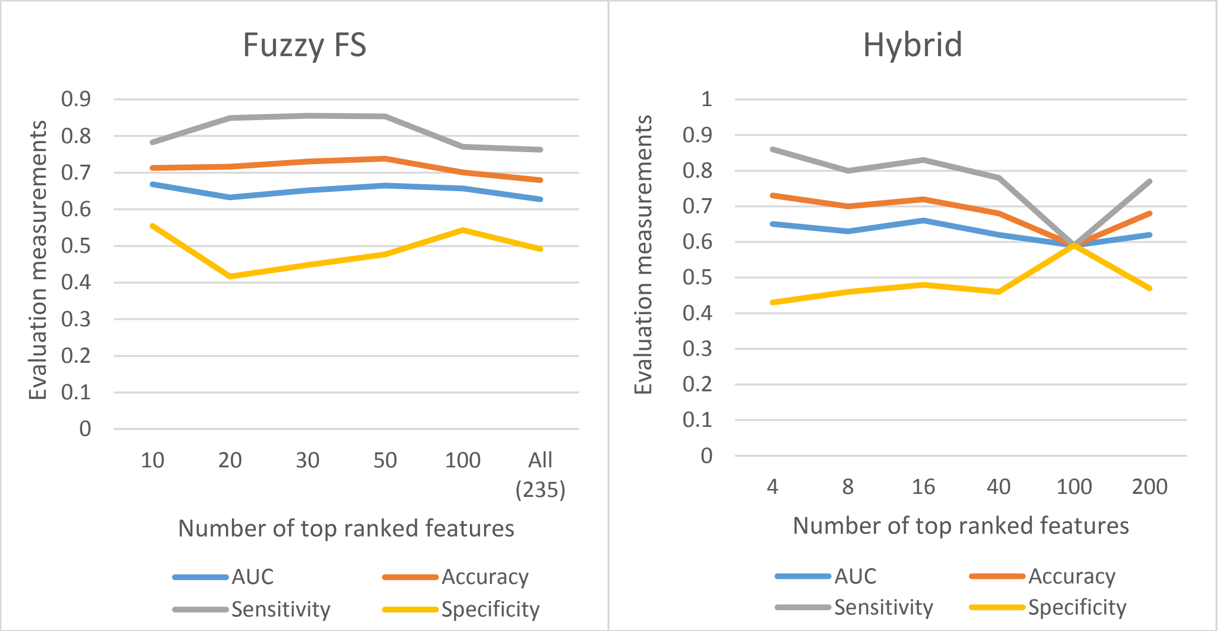


Fig. (8): evaluation of the SCG-NFC (10 clusters & 50 epochs) when trained on TrSRD-C and tested over TestD-C assuming different number of top ranked features of the fuzzy and hybrid feature selection techniques. For fuzzy selection, best AUC (66.9%) achieved when using top 10 ranked features (acc. = 71%). Highest accuracy (73.4%) was at the choice of the top 50 features with AUC=66.5%. For hybrid selection 4 and 16 features shared the highest accuracy and AUC respectively. Using less features has the advantage of less processing time.

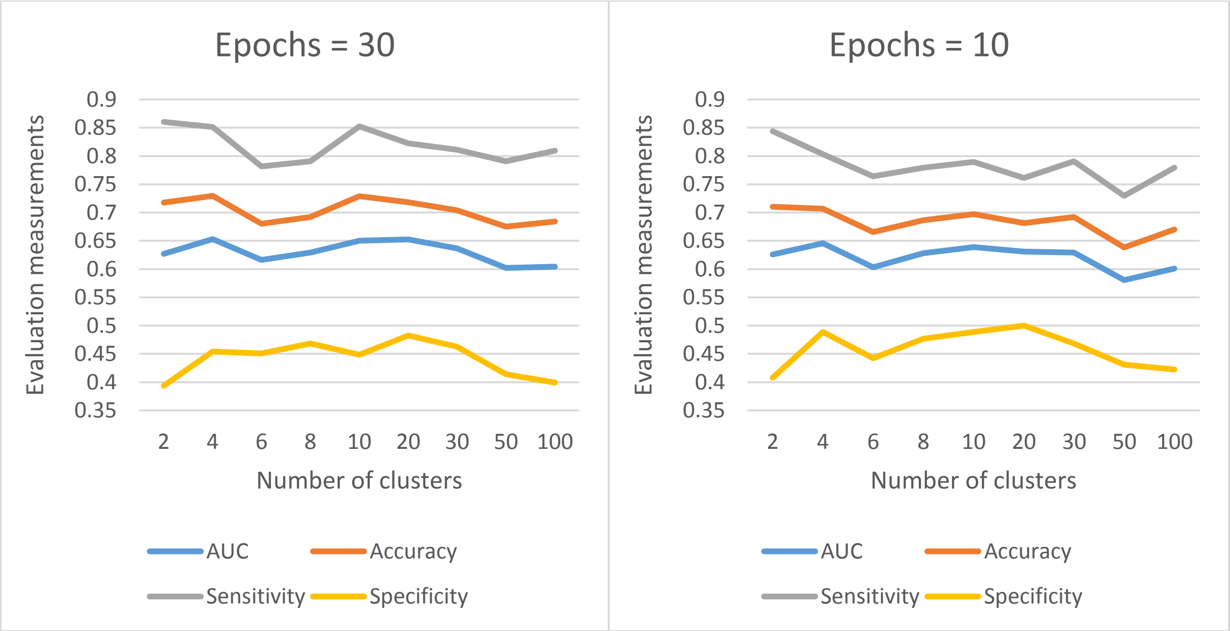


Fig. (9): Different measurements recorded on TestD-C when training the SCG-NFC with TrSRD-C. Hybrid algorithm is the selected feature selection technique. The left graph is the performance of 30 epochs using different clusters number while, the right one is for 10 epochs only. Best AUC and accuracy for both happened at 4 and 10 clusters. Highest AUC for 10 epochs equals 64% (70% accurate) while that for 30 epochs equals 65% (73% accurate).