Genome-scale classification of recombinant and non-recombinant HIV-1 sequences using artificial neural network ensembles

Ashok Kumar Dwivedi* and Usha Chouhan

Department of Mathematics, Bioinformatics and Computer Applications, Maulana Azad National Institute of Technology, Bhopal 462 003, India

Genetic recombination and high rate of mutations in the HIV-1 genome increase the diversity of HIV-1, which allows viruses to escape more easily from host immune system or develop resistance for antiretroviral drugs. Consequently, it is indispensable to devise an effective method for recognition of recombination in HIV-1 strains. This article presents ensemble models of artificial neural network for the classification of recombinant and non-recombinant sequences of HIV-1 genome. We have evaluated the performance of these ensemble models using different classification measurements like specificity, sensitivity and classification accuracy. Furthermore, model performance was measured on receiver operating curve and using calibration graph. High classification accuracy up to 93.43% was achieved on tenfold cross validation.

Keywords: Artificial neural network, bagging, boosting, ensemble, HIV-1 genome.

MILLIONS of people have been infected with HIV which severely affects life and economy of several countries, especially developing countries that do not have proper resources to combat the AIDS epidemic. It is difficult to cure HIV/AIDS because these viruses change strains. The recent approach is to treat patients with multiple drugs[1]. Nevertheless, the drug combinations must be designated to cure a particular strain of virus and it is critical to know the strain which has infected a patient in order to recommend effective course of treatment. HIV-1 strains have been categorized into three major groups known as major (M), outlier (O), and non-M, non-O (N) group. Additionally, there are circulating recombinant forms (CRFs) which are the result recombination of two or more HIV1-subtypes[2], this further complicates the development for effective treatment for HIV/AIDS. It is therefore indispensable to develop competent and effective classification techniques for the determination of HIV-1 subtypes. Most of the techniques developed targetted to classify HIV-1 and HIV-2 sequences or for the classification of subtypes[3]; however no significant work has been done for the effective classification of CRF and non-CRF sequences. Moreover, these techniques are based on finding pairwise distance between sequences or on the basis of phylogenetic distances. It can be envisioned that phylogenetic analysis based on whole genomes is more trustworthy than those based on small segments of the HIV-1 genome. However, such an analysis with is infeasible and intractable due to inherent computational complexity of these techniques.

In this study we propose supervised machine learning technique for the effective classification of recombination and non-recombination HIV-1 sequences. To further enhance the discerning ability of the classifiers, we use artificial neural network (ANN) ensemble and two approaches for ensemble learning – bagging and boosting – have been evaluated for their performance.

The objective of this work is manyfold. First, utilizing entire genome sequences of HIV-1 strains available at Los Alamos National Laboratory[4], for the classification of recombinant and non-recombinant strains. Secondly, to apply ANNs and their ensembles for classification and comparing their performance on various performance indices.

ANNs have been applied in various field of life sciences, including biomedical science, computational biology and bioinformatics[5]. Recently, ANN ensemble has become an encouraging technique of machine learning. In ANN ensemble learning, a group of networks are used to perform the same task and then their predictions are combined. There are different ensemble techniques, but the most frequently used techniques are bagging and boosting. Both use a base classifier for constructing different classifiers for classification. In this study, we use ANN as a base classifier. We show that the ANN ensemble displays significant improvement in the performance[6]. The central idea of the ensemble is to use multiple ANNs which contain more information than any single ANN. This information can be used to improve dependability and generalization of performance[7]. Commonly, several ANNs in the last generation are joined to create an ensemble that has improved generalization performance[7]. Every net contained by the ensemble has in theory dissimilar weight in the output of the ensemble[8,9]. It has been shown by various researchers that the ANN ensemble has a smaller error of generalization than that obtained

*For correspondence. (e-mail: dwivedi.ashok@gmail.com)

by a single ANN[8]. In the direction of maximization of the consequence of coalescing numerous ANNs, an enormous assortment of ANNs should be used. When input x is presented to the network, the output from a characteristic ensemble having K component networks is given by eq. (1)[8,10]

$$y(x) = \sum_{i=1}^{k} w_i y_i(x), \tag{1}$$

where output of network i is y_i and weight connected with the network is w_i.

Only the adequate solution of the previous generation is used in the next generation. When such solutions are combined, it results in greater accuracy than using the best option amongst them. A number of studies have focused on different approaches for combining these solutions, including negative correlation learning, back-propagation and using multi-objective criteria[11–13]. In this article multiple ANNs are produced independently and then combined using two different ensemble techniques, bagging and boosting. Finally the performances of ANN, boosting[14] and bagging[15] ensemble of ANN are compared on different classification performance indices. Moreover, performance is evaluated using receiver operating curve (ROC)[16] and calibration graph[17]. Experiment is designed and framework developed using data mining toolbox in python programing language[18,19].

Materials and methods

Datasets

Genome sequences of all available strains of HIV-1 and HIV-1 CRFs were retrieved from GIV database at Los Almos Laboratory[4]. A total of 4233 complete genomes of HIV-1 were retrieved and clustered in two sets, CRF (1206) and NON-CRF (3027), according to their strain types. The composition of every sequence on their oligonucleotide use (nucleotide and dinucleotide composition) was calculated using software DAMBE[20], and used to construct the training set for the classification.

For genome-level studies we used nucleotide composition, i.e. the percentage of all nucleotides in each genome (four attributes) and similarly, dinucleotide composition, i.e. percentage of dimer frequency in each genome (16 attributes). These two features were used as genome signatures. We also used trinucleotide frequency, but results did not significantly improve in this study in comparison to the time and memory demand; so we have incorporated only nucleotide and dinucleotide composition.

Total number of (4233) records with (20) features was incorporated into the training set. All records were labelled as CRF or NON-CRF according to the class they belongs.

Artificial neural network for classification

There are different architectures of ANN, but in this study we have used combinations of multi-layer perceptron (MLP)[21,22]. We have selected MLP because it is the most suitable choice for non-linear data[23,24]. Our objective was also to show that MLP can be used to classify this data. Our result indicates that it is one of the best choices because we achieved classification accuracy up to 94.19% using MLP and its ensemble.

An MLP is arranged in layers like a multistage directed graph. Each node at each layer receives an input from the connected node of previous layer. Then it calculates the value of a function and provides input to the connected node in the next layer. The layers are designated as 'input layers', 'hidden layers' and 'output layers'. The intermediate layers which do not have direct connection with input and output are called hidden layers. The activation of hidden layers and output layers is calculated by a function, which is a weighted summary of the inputs it collects. This is then passed through an activation function. In a particular layer the activation of node j is defined as

$$a_i = \sum_i w_{ij} o_i, \tag{2}$$

where o_i represents the yield of node i in the preceding layer and w_{ij} is the weight on the link from node i to the present node j. The weights are real-valued numbers and typically initialized randomly in a small range, for example, [–1, 1]. For particular input and output pairs, there is a set of weight values which will decrease the mean squared error. The output o_j is produced by the activation function

$$o_j = \frac{1}{1+\exp(-a_j)}. \tag{3}$$

This is known as the logistic function which has been used in this study. Several methods can be used to find values of the weights[21,22].

Methods for updating the weights: Back-propagation is common algorithm for updating the weights[25], which uses a gradient descent approach in which the weight changes in ratio to the gradient of the error function. The following equation is used to calculate the error single input for output f

$$e_{\text{net}} = \frac{1}{2}(f-d)^2. \tag{4}$$

Equation (5) is used to calculate updates to a weight w_{ij} within the system that yields output f

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} + \Delta w_{ij},$$

$$\Delta w_{ij} = -\alpha \frac{\partial e_{\text{net}}}{\partial w_{ij}}, \quad (5)$$

where α is known as the learning rate. The updates are accomplished iteratively by making small changes to every weight in the system until a minimum of the error function is achieved[22].

Approaches for joining a set of predictors: The amalgamation approach for a collection of predictors is of ultimate importance which can decide the performance of the entire system by managing the limitations of each base component estimator[26]. The modest possible organization strategy is one which takes a weighted summation of the diverse predictor outputs in a linear fashion. Then the output of this arrangement is

$$f_{\text{ens}} = \sum_{i=1}^{M} w_i f_i, \quad (6)$$

where the number of predictors is represented by M, output of the ith predictor is f_i, and w_i is the analogous real-valued non-negative weight.

Constructing ANN ensembles using bagging: Nilson[27] first provided the idea of ANN ensembles in which perceptions are organized in layers, and outputs are joined with a second layer based on vote ranking. In this study we used, bagging and boosting, two of the most extensively used approaches for constructing ANN ensembles.

In bagging[15], a new training set is generated randomly based on the original set for each ensemble member. For generating a training set of size N, we take random sample of N items uniformly with replacement, then the ensemble member is trained with this resample. This procedure is repeated for any new ensemble members. These resampled sets are frequently called bootstrap replicates[28]. Breiman[15] showed that on average 63.2% of the original training set will exist in each replicate. Bagging has demonstrated to be a prevalent technique, appropriate to many problems. Friedman[29] proposed that bagging thrives by decreasing the variance constituent of the error and leaving the bias unaffected; while Grandvalet[30] shown that bagging can converge without reducing variance.

In this study we have used bagging procedure for the construction of ANN ensemble. Several ANNs were trained independently using bootstrap technique and then they were aggregated using suitable grouping method. We need to construct K training sample sets from a give single training set $TR = \{(x_i; y_i) \mid i = 1, 2, ..., l\}$ with K independent ANNs. In order to obtain greater improvement of the aggregation results, it is necessary to create different training sample sets. To do this, we frequently use the bootstrap procedure. Bootstrapping constructs K duplicate training datasets $\{TR_k^{\text{B}} \mid k = 1, 2, ..., K\}$ by random resampling, but with replacement, from the given training dataset TS repeatedly. Each instance x_i in a certain training set, TS may appear several times or not at all in any specific duplicate training dataset. Then these replicated training sets are used to train different ANNs.

Constructing ANN ensembles using boosting: Boosting[31] resamples the datasets with non-uniform distribution as opposed to bagging, which resamples datasets with uniform distribution. Several boosting algorithms have been developed since the preliminary work by Schapire[31], which include cost-sensitive versions[32,33] and those which can deliver confidence approximations in their forecasts[34]. The most widely used variant of boosting methods is AdaBoost[35]. This technique works in rounds, where in a new network is trained in each round.

Initially a network is trained with equal weight on all training patterns. The misclassified patterns are identified at the end of each round and their weights are increased in a new training set. These new weights are fed back for training in the next round. Subsequently, when the required number of networks has been trained, the weighted votes are used to combine them based on their error in training. In case of boosting (AdaBoost.M1)[36], each sample from the training set is assigned with weights. If m classifiers are to be generated, they are done so sequentially such that one classifier is generated in a single iteration. The classifier C_i is generated by updating the weights of training samples based on classification results of classifier C_{i-1}.

This indicates that boosting retains a weight for each instance instead of drawing a sequence of independent bootstrap samples from the original instances. At every test, the vector of weights is attuned to reveal the performance of the resultant classier, with the consequence that the weights of misclassified instances are improved. The final classier is constructed by aggregating the results of learned classifiers by voting in which each vote of the classifiers is calculated as a function of its accuracy[37].

Bauer and Kohavi[38] compared the performance of bagging and boosting in a large experimental study. The analysis indicates that even though bagging yields an ensemble which is superior to any of its constituent classifiers, and is comparatively less vulnerable to noise, it is on average not considerably better than a simple ensemble. The authors[38] found boosting to be a suitable method, but it can quickly overfit and can be vulnerable to noise; similar difficulties of overfitting have been observed by several authors in AdaBoost.

Estimation of classification performance

The following indices were used to measure the classification performance. Here, the number of actual CRFs predicted as CRF is true positive (TP), the number of NON-CRFs predicted as NON-CRF is true negative (TN), the number of NON-CRFs predicted as CRF is false positive (FP) and the number of CRFs predicted as NON-CRF is false negative (FN).

Classification accuracy: The proportion of instances which are correctly classified by the classification learner.

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{FP} + \text{FN} + \text{TP} + \text{TN}).$$

Sensitivity: The ratio of detected positive sample with the total positive samples; for instance, the proportion of CRFs classified as CRFs to total CRFs

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}).$$

Recall: This is similar to sensitivity, but commonly used in text mining.

Specificity: This is measured by finding the proportion of discovered negative samples with all negative samples, for instance, the proportion of NON-CRFs correctly classified as NON-CRFs

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}).$$

Precision: The ratio of true positive instances with all instances classified as positive.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}).$$

Rate of true positive:

$$\text{Rate of TP} = \frac{\text{TP}}{\text{Total positive}}.$$

Rate of false positive:

$$\text{Rate of FP} = \frac{\text{FP}}{\text{Total negative}}.$$

Area under receiver-operating curve (AUC): This is defined as the total area under ROC[39].

Brier score: This defines the degree of accuracy of likelihood calculations, which calculate the average aberration between the forecast probabilities of measures and the real events.

The Matthews correlation coefficient (MCC): The eminence of binary classifier is measured using MCC in machine learning. It is considered as a composed measure which take all cases of true positive, false positive, true negative and false negative into account and can be applied to datasets of diverse sizes. MCC can be considered as a correlation coefficient between the predicted and observed binary classifiers. The value of MCC is between −1 and +1. The value +1 illustrates a perfect estimate, 0 represents random prediction, and −1 specifies total disparity between observation and prediction.

The following formula is used to calculate MCC from confusion matrix

$$\text{MCC} = \frac{\text{TN} \times \text{TP} - \text{FN} \times \text{FP}}{((\text{FN} + \text{TP})(\text{FP} + \text{TP})(\text{FP} + \text{TN})(\text{FN} + \text{TN}))^{1/2}}.$$

Results and discussion

We have used ANN as a base classifier in the ensemble learning techniques, bagging and boosting (AdaBoost.M1), for classification of recombinant and non-recombinant HIV-1 genome sequences. A total of 4233 genomes of HIV-1, including 1206 CRF and 3027 NON-CRF genome strains were used in the classification. We have used tenfold cross-validation techniques for validation of the model. The model was also used for classification of test datasets.

We found that the base classifier (ANN) is itself adequate for classification HIV-1 genome strains with classification accuracy of 93.36% in tenfold cross-validation and 99.88% classification accuracy for the test on training data (Tables 1–4).

Classification accuracy of ANN model and its ensemble models are more or less the same (Table 4) when techniques are applied on the training set. However, merits differ significantly, when tenfold cross-validation technique is used to validate the model (Table 3). While using ANN ensembles the classification accuracy and other merits were improved on tenfold cross validation. Though classification can be done using classic ANN model, the objective of this study is to use ensemble technique for improvement of the result, which is implicated by the result on tenfold cross validation (Table 3). However, using an ensemble does not add any complexity to the model because the difference is in training and only training sets are chosen differently which does not add any further complexity to the MLP model.

However, close resemblance of CRF sequence with multiple HIV-1 strains, causes misclassification of 164, 161, and 159 CRFs by ANN, ANN AdaBoost.M1 and ANN bagging respectively (Tables 1 and 2). Nevertheless, only a small fraction of non-CRFs was misclassified, 0.038%, 0.038% and 0.028% by ANN, AdaBoost.M1 and bagging respectively, on tenfold cross validation (Table 1). A high

Table 1. Confusion matrix for tenfold cross validation using artificial neural network (ANN), ANN AdaBoostin.M1 and ANN bagging (with 100 layers, regularization factor – 0.1, maximum number of iteration – 1000, number of created classifiers for bagging and boosting – 50) for classification of circulating recombinant form (CRF) and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes

		Predicted class					
		ANN		ANN–AdaBoost.M1		ANN bagging	
		CRF	NON-CRF	CRF	NON-CRF	CRF	NON-CRF
Actual class	CRF	1042	164	1045	161	1047	159
	1206	89.9%	5.3%	89.9%	5.2%	92.3%	5.1%
	NON-CRF	117	2910	117	2910	87	2940
	3027	10.1%	94.7%	10.1%	94.8%	7.7%	94.9%
Total	4233	1159	3074	1162	3071	1134	3099

Table 2. Confusion matrix for testing on training data using ANN, ANN AdaBoostin.M1 and ANN bagging (with 100 layers, regularization factor – 0.1, maximum number of iterations – 1000, number of created classifiers for bagging and boosting – 50) for the classification of CRF and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes

		Predicted class					
		ANN		ANN–AdaBoost.M1		ANN bagging	
		CRF	NON-CRF	CRF	NON-CRF	CRF	NON-CRF
Actual class	CRF	1201	5	1203	3	1199	7
	1206	100.0%	0.2%	100.0%	0.1%	100.0%	0.2
	NON-CRF	0	3027	0	3027	0	3027
	3027	0.0%	99.8%	0.0%	99.9%	0.0%	99.8%
Total	4233	1201	3032	1203	3030	1199	3034

Table 3. Classification performance measure indices for tenfold cross validation using ANN, ANN AdaBoostin.M1 and ANN bagging (with 100 layers, regularization factor – 0.1, maximum number of iterations – 1000, number of created classifiers for bagging and boosting – 50) for classification of CRF and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes

	Classification accuracy	Sensitivity	Specificity	Area under curve	Information score	F1 measure	Precision	Brier score	Matthews correlation coefficient
ANN	0.9336	0.8640	0.9613	0.9671	0.6909	0.8812	0.8991	0.1032	0.8355
ANN AdaBoost.M1	0.9343	0.8665	0.9613	0.9400	0.7134	0.8826	0.8993	0.1174	0.8373
ANN bagging	0.9419	0.8682	0.9713	0.9720	0.7020	0.8949	0.9233	0.0904	0.8555

level of classification accuracy (≈100%) is achieved by all three models when training data are used for testing (Table 4). On tenfold cross validation, ANN bagging is better than ANN AdaBoost.M1 and ANN in the sense that only 7.7% of predicted CRFs was misclassified in comparison of 10.1% by the other two classifiers and only 5.1% of predicted non-recombinants were misclassified (Table 1). However, all three models have classification accuracy of 99%, when tested on nontraining set; ANN AdaBoost.M1 is better than the other two models in the number of incorrectly predicted NON-CRFs (Table 2).

Performance evaluation using ROC

ROC[39] plots false positive rate (*X*-axis) and true positive rate (*Y*-axis). It is independent of number of positive and negative cases. ROC is useful when the number of positive and negative cases varies during the training. For the best classifier, area under ROC must be near to 1. Figures 1 and 2 indicate that ANN bagging outperforms the other two techniques. AUC for ROC is nearly equal to one (0.9720) for this technique, which is better than ANN and ANN AdaBoost.M1 (0.9671 and 0.9400 respectively) on

Table 4. Classification performance measure indices for testing on training data using ANN, ANN AdaBoostin.M1 and ANN bagging (with 100 layers, regularization factor – 0.1, maximum number of iterations – 1000, number of created classifiers for bagging and boosting – 50) for classification of CRF and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes

	CA	Sens.	Spec.	AUC	IS	F1	Prec.	Brier	MCC
ANN	0.9988	0.9959	1.0000	1.0000	0.8310	0.9979	1.0000	0.0059	0.9971
ANN AdaBoost.M1	0.9993	0.9975	1.0000	0.9988	0.8604	0.9988	1.0000	0.0014	0.9983
ANN Bagging	0.9983	0.9942	1.0000	1.0000	0.8166	0.9971	1.0000	0.0127	0.9959

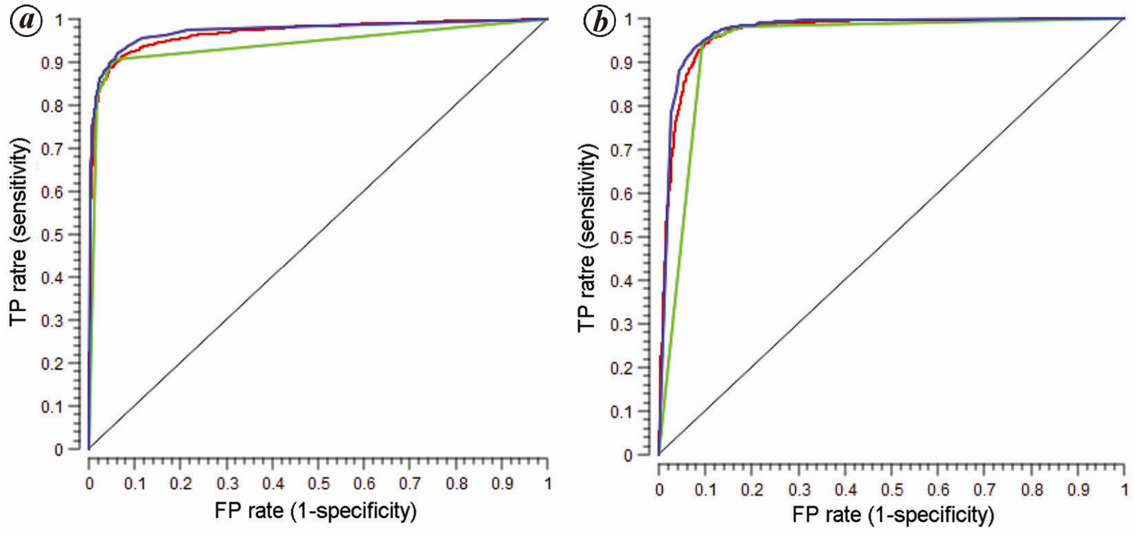

Figure 1. Recover operating curve (ROC) on tenfold cross validation using artificial neural network (ANN), ANN AdaBoostin.M1 and ANN bagging (with 100 layers, regularization factor – 0.1, maximum number of iterations – 1000, number of created classifiers for bagging and boosting – 50) for classification of circulating recombinant form (CRF) and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes. Red, green and blue curves represent ROC for ANN, AdaBoost.M1 and Bagging respectively. (***a***) Predicted class – CRF and (***b***) predicted class – NON-CRF.

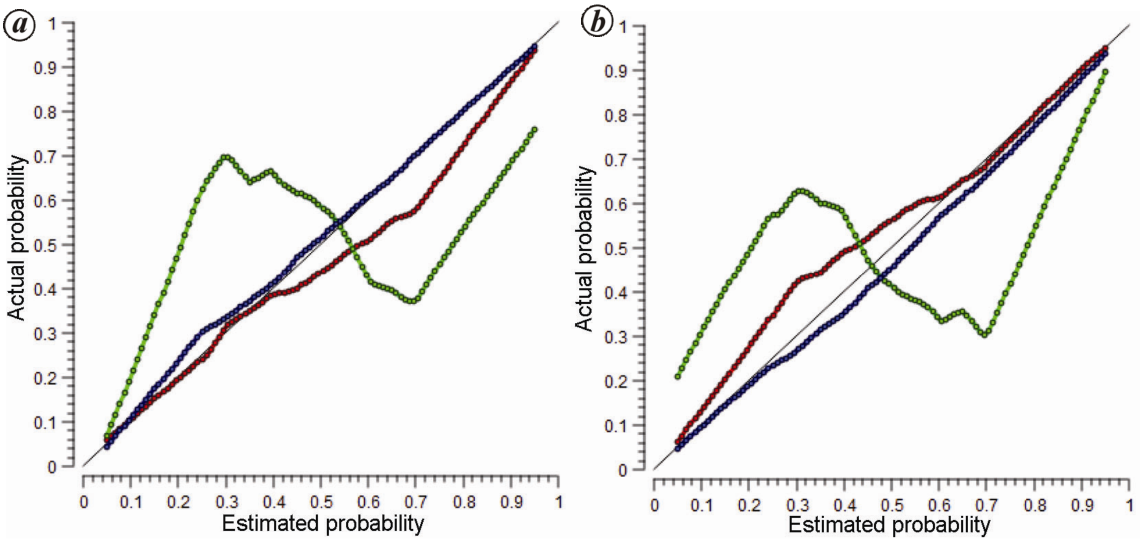

Figure 2. Calibration graph on tenfold cross validation using ANN, ANN AdaBoostin.M1 and ANN Bagging (with 100 layers, regularization factor – 0.1, maximum number of iterations – 1000, number of created classifiers for bagging and boosting – 50) for classification of CRF and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes. Red, green and blue curves represents ROC for ANN, AdaBoost.M1 and bagging respectively. (***a***) Predicted class – CRF and (***b***) predicted class – NON-CRF.

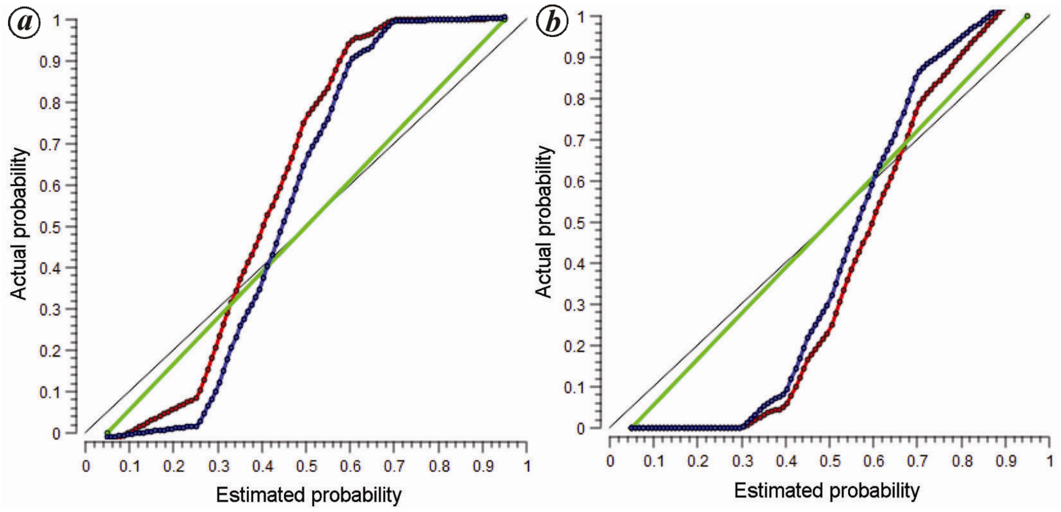

Figure 3. Calibration graph for testing on training data using ANN, ANN AdaBoostin.M1 and ANN bagging (with 100 layers, regularization factor – 0.1, maximum number of iterations – 1000, number of created classifiers for bagging and boosting – 50) for classification of CRF and NON-CRF HIV-1 genome sequences using nucleotide and dinucleotide composition as sequence attributes. Red, green and blue curves represents ROC for ANN, AdaBoost.M1 and bagging respectively. (***a***) Predicted class – CRF and (***b***) Predicted class – NON-CRF.

tenfold cross validation (Tables 3 and 4). Moreover, AUC for all three algorithms is nearly equal when techniques are applied on training datasets (Table 4).

Performance evaluation on calibration graph

Calibration graph[17] plots estimated probabilities (*X*-axis) against actual probabilities (*Y*-axis) and is different compared to ROC. A suitable classifier must also have the property that its predicted probabilities are well calibrated. Nonetheless, even after improvement in the calibration ability, ROC properties and classification ability remain unchanged[17]. A perfect calibration graph has the property that it is represented on the diagonal of the graph, which indicates no difference between the estimated and actual probabilities. Calibration graph for three classifiers on tenfold cross validation is shown in Figure 2 *a* and *b* for CRF and NON-CRF as target class respectively, which clearly indicates that ANN bagging is far better calibrated than ANN and ANN AdaBoos.M1. Furthermore, the calibration ability is not much affected when CRF or NON-CRF is used as target class. Figure 3 *a* and *b* indicates significantly different results with all three techniques when training set is used for testing. The figure shows that ANN AdaBoost.M1 is far calibrated on training set than ANN and ANN bagging. It also indicates that the calibration abilities of ANN and ANN bagging are quite different for CRF and NON-CRF genome sequences.

Thus ANN and its ensemble can effectively and efficiently classify recombinant and non-recombinant genomes of HIV-1 strains. Moreover, their characteristics are different on different evaluation measures.

Conclusion

Effective treatment of HIV-AIDS becomes complicated due to the presence of CRFs. CRFs are the consequence of recombination between two or more HIV-1 subtypes. For this reason conventional phylogenetic classification is not appropriate for HIV-1 classification and genome-scale phylogenetic methods are infeasible. In this study we have devised supervised machine learning technique for effective classification of recombinant HIV-1 sequences. This study reveals the capability of ANN and its ensemble for the classification of non-recombinant and recombinant HIV-1 sequences genome-scale. ANN provides good generalization competencies, but the proficiencies are limited to availability of positive datasets. In such a scenario we propose using ANN ensemble and the performance of ANN ensemble training algorithms, namely bagging and boosting, have been evaluated. Additionally recital of these ensembles was corroborated using tenfold cross validation and tested on training data. On tenfold cross validation we achieved substantial classification correctness of 99.93%. Our results are also significant because almost all available HIV-1 strain genome sequences have been used in the study and highest classification accuracy ≈100% achieved on training sets. Moreover, our study also demonstrates that different machine learning techniques have different characteristics depending on the nature of the data and evaluation criteria. Thus techniques should be preferred according to the requirement of the problem on hand. This method can be extended to classify different sub-types of non-recombinant HIV-1 sequences, i.e. for subtyping of HIV-1 sequences. However, classification accuracy and other merits may be different.

1. Palella Jr, F. J. *et al.*, Declining morbidity and mortality among patients with advanced human immunodeficiency virus infection. *N. Engl. J. Med.*, 1998, **338**, 853–860.
2. Robertson, D. L., Hahn, B. H. and Sharp, P. M., Recombination in AIDS viruses. *J. Mol. Evol.*, 1995, **40**, 249–259.
3. Rambaut, A., Robertson, D. L., Pybus, O. G., Peeters, M. and Holmes, E. C., Human immunodeficiency virus: phylogeny and the origin of HIV-1. *Nature*, 2001, **410**, 1047–1048.
4. Kuiken, C. *et al.*, *HIV Sequence Compendium 2009*, Theoretical Biology and Biophysics, Los Alamos National Laboratory, Los Alamos, New Mexico, 2009.
5. Baxt, W. G., Shofer, F. S., Sites, F. D. and Hollander, J. E., A neural computational aid to the diagnosis of acute myocardial infarction. *Ann. Emer. Med.*, 2002, **39**, 366–373.
6. Hansen, L. K. and Salamon, P., Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intel.*, 1990, **12**, 993–1001.
7. Liu, Y., Yao, X. and Higuchi, T., Evolutionary ensembles with negative correlation learning. *Evol. Comput., IEEE Trans.*, 2000, **4**, 380–387.
8. García-Pedrajas, N., Hervás-Martínez, C. and Ortiz-Boyer, D., Cooperative coevolution of artificial neural network ensembles for pattern classification. *Evol. Comput., IEEE Trans.*, 2005, **9**, 271–302.
9. Ortiz-Boyer, D., Hervás-Martínez, C. and García-Pedrajas, N., Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE Trans. Evol. Comput.*, 2005, **9**(3), 271–302.
10. Yao, X. and Liu, Y., Making use of population information in evolutionary artificial neural networks. *IEEE Trans. Syst. Man Cyber., Part B* (*Cybernetics*), 1998, **28**(3), 417–425.
11. Chan, Z. and Kasabov, N., Fast neural network ensemble learning via negative-correlation data correction. *IEEE Trans. Neural Network*, 2005, **16**(6), 1707–1710.
12. Yang, J. Y., Li, G.-Z., Liu, L.-X. and Yang, M. Q., Classification of brain glioma by using neural networks ensemble with multi-task learning. In BIOCOMP 2007. Proceedings of the International Conference on Collective Dynamics: Topics on Competition and Cooperation in the Biosciences, Vietri, Italy, 24–28 September 2007, pp. 515–522.
13. Li, G.-Z., Liu, T.-Y. and Wu, G.-F., Improving generalization ability of neural networks ensemble with multi-task learning. *J. Comput. Inf. Syst.*, 2006, **2**, 1235–1239.
14. Schapire, R. E., Theoretical views of boosting and applications. In *Algorithmic Learning Theory*, Springer, pp. 13–25.
15. Breiman, L., Bagging predictors. *Mach. Learn.*, 1996, **24**, 123–140.
16. Centor, R. M., Signal detectability the use of ROC curves and their analyses. *Med. Decis. Making*, 1991, **11**, 102–106.
17. Cohen, I. and Goldszmidt, M., Properties and benefits of calibrated classifiers. In 8th European Conference on Principles of Knowledge Discovery in Databases, Pisa, Italy, 20–24 September 2004, pp. 125–136.
18. Demšar, J. *et al.*, Orange: data mining toolbox in python. *J. Mach. Learn. Res.*, 2013, **14**, 2349–2353.
19. Demšar, J., Zupan, B., Leban, G. and Curk, T., *Orange: from Experimental Machine Learning to Interactive Data Mining*, Springer, 2004.
20. Xia, X. and Xie, Z., Dambe: software package for data analysis in molecular biology and evolution. *J. Hered.*, 2001, **92**, 371–373.
21. Bishop, C. M., Neural Networks for Pattern Recognition, 1995.
22. Haykin, S. and Lippmann, R. L., Neural networks: a comprehensive foundation. *Int. J. Neural Systems*, 1994, **5.4**, 363–364.
23. Sozou, P. D., Cootes, T. F., Taylor, C. J., Di Mauro, E. and Lanitis, A., Non-linear point distribution modelling using a multi-layer perceptron. *Image Vision Comput.*, 1997, **15**, 457–463.
24. Asoh, H. and Otsu, N., Nonlinear data analysis and multilayer perceptrons. In *Neural Networks*, 1989 IJCNN, International Joint Conference on Neural Network, *IEEE*, pp. 411–415.
25. Chauvin, Y. and Rumelhart, D. E., *Backpropagation: Theory, Architectures, and Applications*, Psychology Press, 1995.
26. Sharkey, A. J. C. (ed.), *Combining Artificial Neural Nets: Ensemble and Modular Multi-net Systems*, Springer, 2012.
27. Nilsson, N. J., *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, 1925.
28. Efron, B. and Tibshirani, R. J., *An Introduction to the Bootstrap*, CRC Press, 1994.
29. Friedman, J. H., On bias, variance, 0/1 – loss, and the curse-of-dimensionality. *Data Mining Knowledge Discov.*, 1997, **1**, 55–77.
30. Grandvalet, Y., Bagging can stabilize without reducing variance. In *Artificial Neural Networks*, icann 2001 Springer, pp. 49–56.
31. Schapire, R. E., The strength of weak learnability. *Mach. Learn.*, 1990, **5**, 197–227.
32. Fan, W., Stolfo, S. J., Zhang, J. and Chan, P. K., Adacost: misclassification cost-sensitive boosting. In Proceedings – ICML '99, Proceedings of the Sixteenth International Conference on Machine Learning, 1999, pp. 97–105.
33. Joshi, M. V., Kumar, V. and Agarwal, R. C., Evaluating boosting algorithms to classify rare classes: comparison and improvements. In Proceedings IEEE International Conference on Data Mining, 2001, pp. 257–264.
34. Schapire, R. E. and Singer, Y., Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 1999, **37**, 297–336.
35. Freund, Y. and Schapire, R. E., A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Leaving Theory*, Springer, 2005, pp. 23–37.
36. Freund, Y. and Schapire, R. E., A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, Springer, pp. 23–37.
37. Quinlan, J. R., Bagging, boosting, and c4. 5. In Innovative Applications of Artificial Intelligence Conference (IAAI), vol. 1, pp. 725–730.
38. Bauer, E. and Kohavi, R., An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Mach. Learn.*, 1999, **36**, 105–139.
39. Metz, C. E., Basic principles of ROC analysis. In *Seminars in Nuclear Medicine*, Elsevier, 1978, pp. 283–298.

ACKNOWLEDGEMENTS. We thank the Department of Biotechnology (DBT), New Delhi for providing support for this work under Bioinformatics Infrastructure Facility of DBT at Maulana Azad National Institute Technology, Bhopal.

Received 9 November 2014; revised accepted 21 March 2016

doi: 10.18520/cs/v111/i5/853-860