J Med Syst (2016) 40: 69
DOI 10.1007/s10916-015-0428-7

SYSTEMS-LEVEL QUALITY IMPROVEMENT

Lopinavir Resistance Classification with Imbalanced Data Using Probabilistic Neural Networks

Letícia M. Raposo[1] · **Mônica B. Arruda**[2] · **Rodrigo M. de Brindeiro**[2] · **Flavio F. Nobre**[1]

Received: 7 July 2015 / Accepted: 23 December 2015 / Published online: 6 January 2016

Abstract Resistance to antiretroviral drugs has been a major obstacle for long-lasting treatment of HIV-infected patients. The development of models to predict drug resistance is recognized as useful for helping the decision of the best therapy for each HIV+ individual. The aim of this study was to develop classifiers for predicting resistance to the HIV protease inhibitor lopinavir using a probabilistic neural network (PNN). The data were provided by the Molecular Virology Laboratory of the Health Sciences Center, Federal University of Rio de Janeiro (CCS-UFRJ/Brazil). Using bootstrap and stepwise techniques, ten features were selected by logistic regression (LR) to be used as inputs to the network. Bootstrap and cross-validation were used to define the smoothing parameter of the PNN networks. Four balanced models were designed and evaluated using a separate test set. The accuracies of the classifiers with the test set ranged from 0.89 to 0.94, and the area under the receiver operating characteristic (ROC) curve (AUC) ranged from 0.96 to 0.97. The sensitivity ranged from 0.94 to 1.00, and the specificity was between 0.88 and 0.92. Four classifiers showed performances very close to three existing expert-based interpretation systems, the HIVdb, the Rega and the ANRS algorithms, and to a k-Nearest Neighbor.

Keywords Classifier · Resistance · HIV · Probabilistic neural networks

This article is part of the Topical Collection on *Systems-Level Quality Improvement*

✉ Letícia M. Raposo
raposo@peb.ufrj.br

[1] Biomedical Engineering Program, Federal University of Rio de Janeiro - UFRJ, Ilha do Fundão, Rio de Janeiro, RJ, Brazil

[2] Laboratory of Molecular Virology, Federal University of Rio de Janeiro - UFRJ, Ilha do Fundão, Rio de Janeiro, RJ, Brazil

Introduction

The human immunodeficiency virus (HIV) is a retrovirus from the Retroviridae family and is responsible for the acquired immunodeficiency deficiency syndrome (AIDS), which was first documented in 1981 [1]. The control of this infection is achieved using antiretroviral drugs, which help reduce the mortality and morbidity as well as promote increased patient lifespans [2]. However, several patients show or develop resistance to some of the available drugs, which is a major limiting factor of HIV therapy effectiveness.

Different statistical techniques and machine learning algorithms have been developed to predict HIV resistance. Such studies have used statistical modeling [3–5], neural networks [6–8], support vector machines [9] and decision trees [10].

Most of these studies used data provided by genotyping, a test that identifies genetic mutations associated with resistance to antiretroviral drugs. Although it is not considered the gold standard test, genotyping is faster and cheaper than phenotyping and provides a direct quantitative measure of the susceptibility of HIV strains to drugs [11].

Several authors recognize that different classifiers perform poorly with imbalanced data sets [12, 13]. The imbalanced problem is characterized when there are greater instances of some classes than others. Developing classifiers using imbalanced data may result in solutions with a good overall performance due to the tendency of overfitting for the majority class [14].

In this study, we present a classifier for predicting resistance to lopinavir, an HIV protease inhibitor, using genotypic information. Due to the high number of resistant mutations,

we used logistic regression (LR) to select the features and a probabilistic neural network (PNN) to design the classifiers by using balanced data in relation to the classes of resistance. In addition, a comparison of the performance was conducted between the PNN models and three well-known HIV-1 genotyping interpretation systems: HIV db, Rega, and ANRS.

Methods

Genotype dataset

The data were made available from the Molecular Virology Laboratory, Health Science Center, Federal University of Rio de Janeiro (UFRJ /CCS / Brazil), a member of the Brazilian Network for HIV-1 Genotyping (RENAGENO), responsible to perform and analyze genotyping tests for all HIV-infected patients within the public system. The Brazilian HIV data are accessible to the RENAGENO laboratory members and general data are publicly available at www2.aids.gov.br/ final/dados/dados_aids.asp.[1] For this study, 625 amino acid sequences of the protease enzyme of the pol gene of HIV-1 subtype B from infected patients were analyzed.

Modeling

Outcome variable

The outcome was a binary variable that indicated whether the patient was resistant to lopinavir. For patients who were susceptible or had an intermediate resistance to this drug in the last regimen of the therapy, the variable was coded as 0 (non-resistant), whereas those who developed resistance to lopinavir were coded as 1 (resistant). Patient classification was obtained using the HIV Genotyping Test—Brazilian Interpretation Algorithm (version 05:12) [15], which uses a set of predefined rules to identify if there is a particular drug resistance.

Explanatory variables

The explanatory variables were obtained from a set of positions in the HIV-1 protease gene (PR) known to influence drug resistance. The initial positions included here were those obtained from an updated list of mutations associated with resistance to antiretroviral drugs provided by the International Antiviral Society (IAS-USA) [16]. The PR positions with the corresponding amino acid code for the original sequence are: L (amino acid) 10 (position), K20, L24, V32, L33, M46, I47, I50, F53, I54, L63, A71, G73, L76, V82, I84 and L90.

[1] For further information on the Brazilian HIV data banks, contact the co-author Rodrigo Brindeiro (robrinde@biologia.ufrj.br).

Training and test sets

The set of 625 available amino acid sequences was divided into a training set of 500 sequences, and a test set of 125 sequences. In the training group, 400 patients had no resistance to lopinavir, whereas 100 were resistant. In the test group, 30 patients were resistant, and 95 showed no resistance. The training set was used for feature selection and to obtain an optimal smoothing parameter of the PNN. The test set was only used to evaluate the performance of the final models.

Feature selection

The selection of input variables is an important step to enhance the classification ability of the models and to reduce the training and test computing time. Because no feature selection method designed for the PNN was found in the literature, we proposed using the bootstrap method and LR to select the amino acid mutation positions to be the inputs for the PNN model. Bootstrapping is a resampling technique with a replacement proposed by Efron [17]. Each resample has the same sample size as the original data.

From the training set, a total of 1000 bootstrap samples of equal size ($n = 100$) were obtained from the 100 patients with therapeutic failure. Each one of these bootstrap samples was combined with 100 patients randomly selected from the 400 non-resistant individuals, generating a balanced set used for model estimation.

One LR model was designed for each bootstrap sample. The variables of each of the 1000 models were selected by the stepwise method, using the Akaike information criterion (AIC) [18]. For this method, a sequence of regression models is obtained by adding or removing variables at each step. Non-significant variables are excluded, and the procedure is repeated until no other variable improves the model [19]. The AIC penalizes models with many variables, and lower values of AIC are preferred. The final chosen variables used as input to the PNN were those selected in 50 % of the LR models.

PNN modeling

A PNN is an artificial neural network (ANN) used in different classification problems [20–24]. This particular ANN proposed by Specht [25] has a faster training than the multilayer perceptron network. It generates accurate predicted target probability scores, and it is relatively insensitive to outliers.

A PNN uses Bayesian decision to classify the input vectors. The optimal decision rule that minimizes the average cost of misclassification is called the Bayes optimal decision rule [26]. The architecture of a classic PNN is shown in Fig. 1.

The input layer has as many neurons as the number of explanatory variables and only distributes the input to the

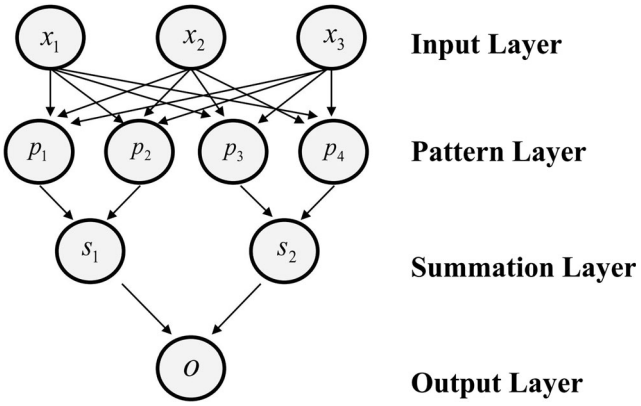

Fig. 1 Basic architecture of a probabilistic neural network

neurons in the pattern layer. The pattern layer contains one neuron for each case in the training data set. The neurons of the pattern layer are divided into i groups, one for each class. Each neuron receives the input vector and estimates its probability density function (PDF) using the Parzen window method [27]. In this study, the Gaussian function was used as the Parzen window. The ith kernel node in the jth group is defined as a Gaussian basis function:

$$p_{i,j}(x) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left(-\frac{\left\|x - x_{i,j}\right\|^2}{2\sigma^2}\right) \quad (1)$$

where xi,j is the vector of sample stored in the pattern unit of class i or j (the center of the kernel), d is the number of input variables and σ is a smoothing (spread) parameter.

The summation layer sums the output from pattern units associated with a given class. This layer has as many processing units as the number of classes to be recognized. Each of these units estimates a class-conditional PDF using a mixture of Gaussian kernels according to equation 2:

$$f_{i,j}(x) = \sum_{i=1}^{N_j} \alpha_{i,j} p_{i,j}(x), \quad 1 \le j \le n \quad (2)$$

where $\alpha_{i,j}$ satisfies:

$$\sum_{i=1}^{N_j} \alpha_{i,j} = 1, \quad 1 \le j \le n \quad (3)$$

The output layer makes the decision based on the maximum probability of Bayes' rule. A competitive transfer function on output neurons selects the node with the highest output and assigns a 1 (positive identification) to that class and a 0 (negative identification) to non-targeted classes.

After the feature selection step, we use a combination of bootstrap and cross-validation to choose the smoothing parameter of the PNN. Following a similar procedure as previously described for feature selection, 100 balanced subsets with size 200 were obtained, and for each subset, a PNN model was implemented. The smoothing parameter should be chosen to obtain the highest accuracy of the classifier. Therefore, we varied the parameter from 0.1 to 1 in steps of 0.1, and for each smoothing parameter, a 10-fold cross validation to evaluate the model was applied.

The data were partitioned into 10 equal sub-samples. For each smoothing value, a PNN was trained with 90 % of the data and was evaluated with the remaining sub-sample. The area under the receiver operating characteristic (ROC) curve (AUC) was the accuracy criterion. Ten computed AUCs from the folds were averaged to produce a single estimation for that particular value of the smoothing constant, and the smoothing parameter was selected as the value that provided the best average AUC. This procedure was repeated for each one of the 100 balanced subsets. The final smoothing parameter was defined as the average of the smoothing parameters associated with the best AUCs of each subset.

The variables selected by LR and the estimated smoothing parameter were employed to develop four PNN models over four balanced test sets, which were later used in the validation step. The four balanced data sets were obtained by dividing the 400 nonresistant samples into four sub-samples of size 100 and combining each one with the available 100 resistant samples from the training set.

Evaluation of the PNN models

The performance of the four final models was evaluated using ROC curve analysis, AUC, accuracy, sensitivity, and specificity. The models were applied to the test set with 125 samples, which was not used at any other stage of the analysis.

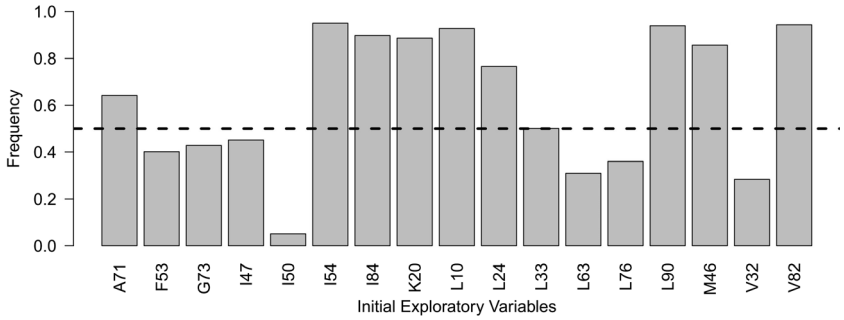

Fig. 2 Frequency of variables selected in the 1000 bootstrap samples in logistic regression

Table 1 Test set evaluation results with 95 % confidence interval (CI) of PNN classifiers

	AUC	Acc	Se	Sp
Classifier 1	0.96 (0.92–0.99)	0.89 (0.83–0.94)	0.94 (0.81–0.99)	0.88 (0.79–0.94)
Classifier 2	0.96 (0.93–0.99)	0.92 (0.86–0.96)	1.00 (0.85–1.00)	0.89 (0.81–0.95)
Classifier 3	0.96 (0.93–0.99)	0.91 (0.85–0.96)	1.00 (0.85–1.00)	0.88 (0.79–0.94)
Classifier 4	0.97 (0.94–0.99)	0.94 (0.88–0.97)	0.97 (0.85–1.00)	0.92 (0.85–0.97)
Mean	0.96 (0.95–0.97)	0.91 (0.89–0.94)	0.98 (0.94–1.00)	0.89 (0.85–0.92)

Acc Accuracy, Se Sensitivity, Sp Specificity

An ROC curve characterizes the performance of a binary classification model across all possible cut-offs and depicts the tradeoff between sensitivity and the false-positive rate. The AUC represents the expected performance as a single scalar.

Accuracy (Acc) is defined as the proportion of correct classification by the model over the total sample. This measure is given by the following formula:

$$\mathrm{Acc} = (\mathrm{TP} + \mathrm{TN}) \Big/ (\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}) \quad (4)$$

where TP, FP, TN, and FN are true positives, false positives, true negatives and false negatives, respectively.

Sensitivity (Se) measures the proportion of true positives compared to the total positive class, and specificity (Sp) comprises the proportion of true negatives in relation to the total negative class.

$$\mathrm{Se} = \mathrm{TP} \Big/ (\mathrm{TP} + \mathrm{FN}) \quad (5)$$

$$\mathrm{Sp} = \mathrm{TN} \Big/ (\mathrm{TN} + \mathrm{FP}) \quad (6)$$

Classification algorithms for comparison

The classifiers were compared to three rule-based genotypic interpretation systems, HIVdb (version 7.0) [28], Rega (version 9.1.0) [29] and ANRS (Agence Nationale de Recherches sur le SID) (version 2013.09) [30].

In addition to the PNN, the k-Nearest Neighbors (k-NN), a non-probabilistic algorithm, was implemented to provide a comparison of diagnostic performance. The k-NN algorithm classifies each test case by a majority vote of its neighbors, with the case being assigned to the class most common amongst its k nearest neighbors as measured by Euclidean distance. The dataset used was the same applied in the PNN, and the input variables were those selected by LR.

Software

PNN classifiers were implemented using MATLAB® software package (MATLAB version R2009b with neural networks' toolbox) [31]. Statistical analysis and LR were performed using the open source R software version 3.0.1 [32].

Results

We initially selected the 16 positions of lopinavir resistance provided by the IAS-USA as the input for the variable selection approach using bootstrap and stepwise LR. The percentages of selection of each input variable in 1000 bootstrap samples are shown in Fig. 2. The final selected features, those that were selected in at least 50 % of the models, were the ten following mutation positions: A71, I54, I84, K20, L10, L24, L33, L90, M46, and V82. The PNN smoothing parameter was set to 0.63, which is the average of 100 smoothing parameters with the best AUCs of each model as described previously.

The four PNN classifiers developed using the same 100 resistant samples combined with random samples of the same size from the 400 non-resistant patients were evaluated with the test set. Because the test set emulates a real situation, with resistant and non-resistant patients arriving at random and without knowledge of whether it was balanced or not, we had in this test set 30 resistant and 95 nonresistant patients. Table 1 shows the performance of the PNN classifiers. The mean AUC equals 0.96, accuracy equals 0.91, and sensitivity and specificity equal 0.98 and 0.89, respectively. The ROC curves for the four classifiers are shown in Fig. 3.

The k-NN algorithm resulted in classifiers with a mean AUC equal to 0.93, an accuracy equal to 0.91 and a sensitivity

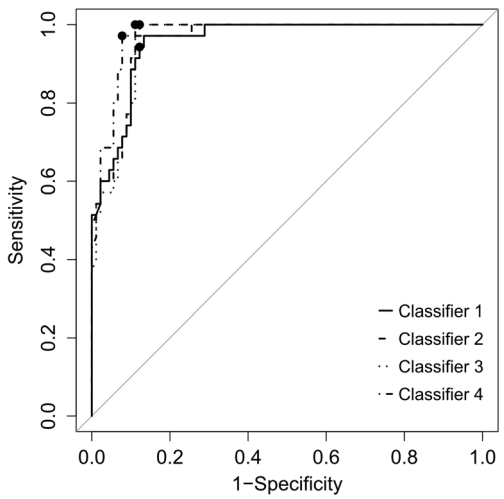

Fig. 3 ROC curve for the PNN classifiers. The *black points* are the threshold of 0.5 used to predict class

Table 2 Test set evaluation results with 95 % confidence interval (CI) of k-NN classifiers

	AUC	Acc	Se	Sp
Classifier 1	0.91 (0.87–0.96)	0.90 (0.83–0.94)	0.94 (0.81–0.99)	0.88 (0.79–0.94)
Classifier 2	0.94 (0.90–0.97)	0.91 (0.85–0.96)	1.00 (0.85–1.00)	0.88 (0.79–0.94)
Classifier 3	0.94 (0.91–0.98)	0.92 (0.86–0.96)	1.00 (0.85–1.00)	0.89 (0.81–0.95)
Classifier 4	0.93 (0.89–0.97)	0.91 (0.85–0.96)	0.97 (0.85–1.00)	0.89 (0.81–0.95)
Mean	0.93 (0.91–0.95)	0.91 (0.85–0.95)	0.98 (0.94–1.00)	0.88 (0.85–0.91)

Acc Accuracy, Se Sensitivity, Sp Specificity

and specificity equal to 0.98 and 0.88, respectively. Table 2 summarizes the performance of k-NN classifiers.

HIVdb, Rega and ANRS algorithms classified the data at three levels of resistance: susceptible, intermediate resistance and high level of resistance. To compare the ability of these algorithms to our models, the outputs of the algorithms were classified according to the following criteria: susceptible or intermediate resistance were classified as non-resistant, and samples classified as a high resistance were classified as resistant. Table 3 summarizes the performance of these three algorithms.

Discussion

In the present study, we used an approach combining bootstrap and LR stepwise variable selection followed by the prediction of resistance to the antiretroviral lopinavir with a PNN neural network. Only those variables that appeared in 50 % of logistic regression models were used in final models. If the cutoff point was increased to 60 %, only position 33 (50.1 %) did not appear in PNN models and an increase to 70 % would also exclude position 71 (64.2 %). Using a cutoff equals to 70 % the classifiers had a lower overall performance, with mean AUC equal to 0.65, accuracy of 0.46 and sensitivity and specificity equal to 0.99 and 0.27, respectively. For the test data, the PNN classifier showed predictive performances greater or comparable to three well-known interpretation systems.

In this study, feature selection and model development used balanced data sets. Most classification procedures assume balanced training data sets in its learning stage [14]. When these methods are trained on highly imbalanced data sets, they often tend to overpredict the presence of the majority class [33]. For example, if the data have a large number of negative cases, it is possible that the classifier shows a higher specificity than sensitivity, which results in an overestimated accuracy.

The available data set had fewer instances of resistance class compared to susceptible or non-resistance class. We addressed this problem by using random undersampling of the majority class. Accuracy alone is not a good measure of the performance of a classifier because it is strongly biased in favor of the majority class. Moreover, this measure considers different classification errors as equally important. It would be more attractive if we used a performance measure that disassociates the errors that occur in each class. In addition to global performance metrics such as AUC or Acc, other parameters should be considered to evaluate classifiers, such as sensitivity and specificity. The absence of these parameters may impair a proper evaluation of the model as well as a misinterpretations of the results.

Several studies report only accuracy, which reduces the interpretation of their results. In a recent study, Pasomsub et al. [8], with a feed-forward artificial neural network showed that developed classifier had an AUC equal to 0.92 (95 % IC: 0.88–0.95) for lopinavir. However, they did not mention other indices, such as sensitivity and specificity, and there is no indication if their data set was balanced or not.

In a study developed by Rhee et al. [34], a feed-forward network was used in the development of models, using a complete set of 70 positions in HIV-protease and a set of selected positions by the list of IAS. For lopinavir, the accuracy was 0.76 for the full set of positions and 0.73 for the list of IAS, which was lower than those found in our study.

The four classifiers showed very similar performances, with accuracies ranging from 0.89 to 0.94 and an average AUC equal to 0.96. When applying the variables selected by the approach proposed in this present study, the k-Nearest Neighbor exhibited results similar to PNN models, demonstrating that this feature selection method could be applied to

Table 3 Test set evaluation results with 95 % confidence interval (CI) of HIVdb, Rega and ANRS algorithms

	AUC	Acc	Se	Sp
HIVdb	0.91 (0.84–0.97)	0.93 (0.87–0.97)	0.86 (0.70–0.95)	0.96 (0.89–0.99)
Rega	0.74 (0.66–0.83)	0.86 (0.78–0.91)	0.49 (0.31–0.66)	1.00 (0.94–1.00)
ANRS	0.94 (0.89–1.00)	0.97 (0.92–0.99)	0.89 (0.73–0.97)	1.00 (0.94–1.00)

Acc Accuracy, Se Sensitivity, Sp Specificity

probabilistic and non-probabilistic algorithms. In all cases, they were at least comparable or superior to some metrics to HIVdb, Rega, and ANRS algorithms, three well-known rule-based genotypic interpretation systems used for many clinicians to predict resistance to specific antiretrovirals. Compared with these prediction algorithms, our approach requires fewer features—10 positions as input to the PNN model to classify lopinavir resistance in contrast to 17 positions proposed by IAS. Additionally, feature selection can be revised and PNNs re-trained without difficulties when new data are made available or new resistance positions are reported.

The limitations of this approach for predicting HIV resistance deserve consideration. First, this approach can only predict drug resistance that is included in the training set, which in our case was lopinavir. Although this is a limitation, the method can be trained with available data for other drugs, but here, we did not have enough samples to properly develop models for other drugs. Second, the choice of features and smoothing parameter of the PNN neural network requires some computational effort. However, once this stage is accomplished, the prediction speed is very high.

With specificity and sensitivity of 0.98 and 0.89, respectively, the PNN classification developed here may serve as a useful tool to support decision making regarding the prediction of resistance of HIV+ patients, thus assisting physicians in their treatment of HIV+ patients. Additional applications of this approach using other antiretroviral drugs in therapeutic practice are needed to better evaluate the impact and the usefulness of the proposed PNN model.

Acknowledgments The authors would like to acknowledge FAPERJ (Carlos Chagas Filho Foundation for Research Support of the State of Rio de Janeiro), CNPq Brazil (National Counsel of Technological and Scientific Development) and CAPES (Coordination for the Improvement of Higher-Education Personnel) for the financial support provided for this research.

References

1. Rambaut, A., Posada, D., Crandall, K., and Holmes, E., The causes and consequences of HIV evolution. *Nat. Rev. Genet.* 5:52–61, 2004. doi:10.1038/nrg1246.
2. WHO (2015) Progress report 2011: Global HIV/AIDS response. http://www.who.int/hiv/pub/progress_report2011/en/. Accessed 28 Oct 2014.
3. Prosperi, M. C. F., Altmann, A., Rosen-Zvi, M., et al., Investigation of expert rule bases, logistic regression, and non-linear machine learning techniques for predicting response to antiretroviral treatment. *Antivir. Ther.* 14:433–442, 2009.
4. Van der Borght, K., Verheyen, A., Feyaerts, M., et al., Quantitative prediction of integrase inhibitor resistance from genotype through consensus linear regression modeling. *Virol. J.* 10:8, 2013. doi:10.1186/1743-422x-10-8.
5. Raposo, LM, Arruda, MB, Brindeiro, RM et al., Logistic regression models for predicting resistance to HIV protease inhibitor nelfinavir. In: Romero LMR (ed) XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013, IFMBE Proceedings, vol 41. Springer International Publishing 1237–1240, 2014.
6. Bonet, I., García, M. M., and Saeys, Y., Predicting Human Immunodeficiency Virus (HIV) drug resistance using recurrent neural networks. In: Mira, J. (Ed.), *Bio-inspired Modeling of Cognitive Tasks, Lectures Notes in Computer Science, vol 4527*. Springer Berlin, Heidelberg, pp. 234–243, 2007.
7. Larder, B., Wang, D., Revell, A., et al., The development of artificial neural networks to predict virological response to combination HIV therapy. *Antivir. Ther.* 12:15–24, 2007.
8. Pasomsub, E., Sukasem, C., Sungkanuparph, S., et al., The application of artificial neural networks for phenotypic drug resistance prediction: Evaluation and comparison with other interpretation systems. *Jpn. J. Infect. Dis.* 63:87–94, 2010.
9. Beerenwinkel, N., Geno2pheno: Estimating phenotypic drug resistance from HIV-1 genotypes. *Nucleic Acids Res.* 31:3850–3855, 2003. doi:10.1093/nar/gkg575.
10. Beerenwinkel, N., Schmidt, B., Walter, H., et al., Diversity and complexity of HIV-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype. *Proc. Natl. Acad. Sci.* 99: 8271–8276, 2002. doi:10.1073/pnas.112177799.
11. Wang, D., and Larder, B., Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks. *J. Infect. Dis.* 188:653–660, 2003. doi:10.1086/377453.
12. Chawla, N., Japkowicz, N., and Kotcz, A., Editorial. *ACM SIGKDD Explor. Newslett.* 6:1, 2004. doi:10.1145/1007730.1007733.
13. Sun, Y., Kamel, M., Wong, A., and Wang, Y., Cost-sensitive boosting for classification of imbalanced data. *Pattern Recogn.* 40:3358–3378, 2007. doi:10.1016/j.patcog.2007.04.009.
14. He, H., and Garcia, E., Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* 21:1263–1284, 2009. doi:10.1109/tkde.2008.239.
15. Algoritmo Brasileiro, Interpretação—Genotipagem do HIV-1. http://forrest.ime.usp.br:3001/resistencia 2012. Accessed 15 Sep 2014.
16. Wensing, A. M., Calvez, V., Günthard, H. F., et al., 2014 Update of the drug resistance mutations in HIV-1. *Top Antivir. Med.* 22:642–650, 2014.
17. Efron, B., Bootstrap methods: Another look at the jackknife. *Ann. Stat.* 7:1–26, 1979. doi:10.1214/aos/1176344552.
18. Akaike, H., A new look at the statistical model identification. *IEEE Trans. Autom. Control* 19:716–723, 1974. doi:10.1109/tac.1974.1100705.
19. Krzanowski, WJ., An Introduction to Statistical Modelling. Reprint edition, John Wiley & Sons, 2010.
20. Budak, F., and Übeyli, E., Detection of resistivity for antibiotics by probabilistic neural networks. *J. Med. Syst.* 35:87–91, 2009. doi:10.1007/s10916-009-9344-z.
21. Bascil, M. S., and Oztekin, H., A study on hepatitis disease diagnosis using probabilistic neural network. *J. Med. Syst.* 36(3):1603–6, 2013. doi:10.1007/s10916-010-9621-x.
22. Singh, K., Gupta, S., and Rai, P., Predicting carcinogenicity of diverse chemicals using probabilistic neural network modeling approaches. *Toxicol. Appl. Pharmacol.* 272:465–475, 2013. doi:10.1016/j.taap.2013.06.029.
23. Kumar, H. P., and Srinivasan, S., Classification of ovary abnormality using the probabilistic neural network (PNN). *Technol. Health Care: Off. J. Europ. Soc. Eng. Med.* 22:857–865, 2014.
24. Hirschauer, T. J., Adeli, H., and Buford, J. A., Computer-aided diagnosis of Parkinson's disease using enhanced probabilistic neural network. *J. Med. Syst.* 39(11):179, 2015. doi:10.1007/s10916-015-0353-9.
25. Specht, D., Probabilistic neural networks. *Neural Netw.* 3:109–118, 1990.

26. Berrar, DP, Downes, CS, and Dubitzky, W., Multiclass cancer classification using gene expression profiling and probabilistic neural networks. *Pac. Symp. Biocomput*. 5–16, 2003.
27. Parzen, E., On estimation of a probability density function and mode. *Ann. Math. Statist.* 33:1065–1076, 1962. doi:10.1214/aoms/1177704472.
28. Liu, T., and Shafer, R., Web resources for HIV type 1 genotypic-resistance test interpretation. *Clin. Infect. Dis.* 42:1608–1618, 2006. doi:10.1086/503914.
29. Rega Instituut KU Leuven., Rega Algorithm. https://rega.kuleuven.be/cev/avd/software/rega-algorithm. Accessed 20 Oct 2014.
30. HIV French Resistance., HIV-1 genotypic drug resistance interpretation's algorithms http://www.hivfrenchresistance.org/index.html. Accessed 20 Oct 2014.
31. The MathWorks, Inc., *MATLAB and Statistics Toolbox Release* 2009b, Massachusetts.
32. R Development Core Team., R: A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing, 2013.
33. Wei, Q., and Dunbrack, R., The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS ONE* 8: e67863, 2013. doi:10.1371/journal.pone.0067863.
34. Rhee, S.-Y., Taylor, J., Wadhera, G., et al., Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proc. Natl. Acad. Sci. U. S. A.* 103:17355–17360, 2006.