Bioinformatics Advance Access published June 21, 2013

Multilabel classification for exploiting cross-resistance information in HIV-1 drug resistance prediction

Dominik Heider [1],* Robin Senge [2], Weiwei Cheng [2] and Eyke Hüllermeier [2]
[1]Department of Bioinformatics, University of Duisburg-Essen, Germany
[2]Department of Mathematics and Computer Science, University of Marburg, Germany

Associate Editor: Prof. Alfonso Valencia

ABSTRACT

Motivation: Antiretroviral treatment regimens can sufficiently suppress viral replication in HIV infected patients and prevent the progression of the disease. However, one of the factors contributing to the progression of the disease despite ongoing antiretroviral treatment is the emergence of drug resistance. The high mutation rate of HIV can lead to a fast adaptation of the virus under drug pressure, thus to failure of antiretroviral treatment due to the evolution of drug-resistant variants. Moreover, cross-resistance phenomena have been frequently found in HIV-1, leading to resistance not only against a drug from the current treatment, but also to other not yet applied drugs. Automatic classification and prediction of drug resistance is increasingly important in HIV research as well as in clinical settings, and to this end, machine learning techniques have been widely applied. Nevertheless, cross-resistance information was not taken explicitly into account, yet.

Results: In our study, we demonstrated the use of cross-resistance information to predict drug resistance in HIV-1. We tested a set of more than 600 reverse transcriptase sequences and corresponding resistance information for six nucleoside analogues. Based on multilabel classification models and cross-resistance information, we were able to significantly improve overall prediction accuracy for all drugs, compared to single binary classifiers without any additional information. Moreover, we identified drug-specific patterns within the RT sequences that can be used to determine an optimal order of the classifiers within the classifier chains. These patterns are in good agreement with known resistance mutations and support the use of cross-resistance information in such prediction models.

Contact: dominik.heider@uni-due.de

1 INTRODUCTION

The human immunodeficiency virus (HIV) is one of the major human diseases leading to about 2 million deaths yearly. Although antiretroviral treatment is working quite well in principle, the high mutation rate of HIV frequently leads to a fast adaptation of the virus and thus to the development of drug resistant viral strains. Tripathi *et al.* (2012) performed stochastic simulations of the within-host evolution of HIV-1. By estimating the structure of the HIV-1 quasispecies, they were able to calculate an error threshold of HIV-1. They discovered that HIV-1 has a mutation rate that is very close to error catastrophe and that the error threshold depends heavily on the recombination rate of HIV-1 (Tripathi *et al.*, 2012). Pennings (2012) analyzed the evolution of resistance in HIV-1 due to standing genetic variation. She ascertained that, depending on the treatment, probabilities of evolution of drug resistance due to standing genetic variation vary between 0 and 39%.

The most important parameters for the evolution of drug resistance are the effective population size of the virus before treatment and the fitness of the resistant virus during treatment. Evolution of drug resistance finally leads to a failure of antiretroviral treatment and thus to the progression of disease. Experimental testing of viral resistance in patients have been widely used in research as well as in clinical settings to gain information about the way drug resistance evolve. In contrast to these experimental phenotypic assays, computational approaches offer the possibility to predict drug resistance in HIV-1 in a very easy and fast way based on short sequence information of the viral genotype, e.g. the sequence of the viral reverse transcriptase. Such computational models for predicting drug resistance in HIV-1 have been developed and widely applied. These computational models are mainly based on statistical or machine learning methods that try to find a mapping from the sequence information to a “resistance factor“. Usually, the IC_{50} ratios[1] are used in these models to define resistance. In general, drug resistance means reduced inhibition of viral replication by antiretroviral drugs, resulting in increased IC_{50} values. The IC_{50} values of the drug resistant isolates and HIV wildtype are used to calculate resistance factors

$$\frac{IC_{50}(\text{drug concentration for resistant strain})}{IC_{50}(\text{drug concentration for wild type})},$$

as a standardized measure of HIV drug resistance. The data is separated into the classes “susceptible“ and “resistant“ based on a drug-specific cutoff of the resistance factor. A computational model is then trained based on these data pairs (sequence and corresponding class), which can then be used to predict the resistance factor or the resistance class for new unseen sequences. For instance, Rhee *et al.* (2006) used five different statistical and machine learning methods (decision trees, artificial neural networks, support-vector machines, least-squares regression and least angle

*to whom correspondence should be addressed

[1] The concentration of a specific drug inhibiting 50% of viral replication compared to cell culture experiments without the drug is defined as IC_{50} (50% inhibitor concentration).

regression) to predict drug resistance in HIV-1 for 16 drugs, including protease- and reverse transcriptase inhibitors. Kierczak *et al.* (2009) developed a rough set-based model considering physico-chemical changes of mutated sequences compared to the wildtype strain to predict reverse transcriptase inhibitor resistance in HIV-1. Moreover, the same group developed the first systems biology approaches to HIV-1 drug resistance, showing networks of interacting positions (Kierczak *et al.*, 2010).

One very important finding, however, has not been exploited in these models so far, namely the occurrence of *cross-resistance*. In the context of HIV-1, cross-resistance means that mutations leading to a resistance against a specific drug, which is currently part of the antiretroviral treatment of a specific patient, also leads to resistance[2] against other drugs (that may or may not be part of the same treatment). In the current study, we analyzed cross-resistance in HIV-1 for a dataset of more than 600 reverse transcriptase sequences and six nucleoside analogues (NAs), namely Lamivudine (3TC), Abacavir (ABC), Zidovudine (AZT), Stavudine (d4T), Didanosine (ddI) and Tenofovir (TDF). Moreover, we developed a model that exploits knowledge about cross-resistance to improve the overall prediction accuracy for the whole repertoire of drugs used in this study. To this end, we made use of novel methods for so-called *multilabel classification*, a generalization of conventional (polychotomous) classification that has recently gained increasing attention in machine learning (Tsoumakas and Katakis, 2007).

To the best of our knowledge, this is the first time information about reverse transcriptase inhibitors cross-resistance has been explicitly integrated in HIV-1 drug resistance prediction models. In contrast to protease inhibitors (PIs) as well as non-nucleoside reverse transcriptase inhibitors (NNRTIs), NAs have less side-effects (Stürmer *et al.*, 2007). Moreover, the combination of different drug-classes during therapy may lead to unpredictable interactions of PIs and NNRTIs with the cytochrome P450 system and thus may complicate the therapy. Therefore, one option might be the use of quadruple nucleoside therapy (Stürmer *et al.*, 2007).

2 METHODS

2.1 Data

For our analyses, we used a publicly available dataset consisting of reverse transciptase (RT) sequences of HIV-1 with corresponding resistance factors (IC_{50} ratios) for six nucleoside analogues (NAs), namely Lamivudine (3TC), Abacavir (ABC), Zidovudine (AZT), Stavudine (d4T), Didanosine (ddI) and Tenofovir (TDF) (Rhee *et al.*, 2006). A summary of the dataset is shown in Table 1. For our method, we needed RT sequences for which IC_{50} ratios for all mentioned drugs are available, so we discarded the information about TDF (due to the low number of sequences) and merged the other sequences and IC_{50} ratios of the remaining drugs. Strains lacking phenotypic results for any drug analyzed in the current study were discarded prior to analysis. Eventually, this yields 614 RT sequences with IC_{50} ratios for 3TC, ABC, AZT, d4T and ddI. The class ratio (positive samples / negative samples) for all drugs ranges between 0.83 and 2.39. All sequences originated from subtype B strains.

Table 1. Data overview

Drug	number of sequences	IC_{50} ratio cutoff	class ratio*
3TC	629	≥ 3	2.18
ABC	624	≥ 2	2.39
AZT	626	≥ 3	0.91
ddI	628	≥ 1.5	1.03
d4T	626	≥ 1.5	0.83

*:ratio of resistant vs. susceptible sequences

2.2 Predictive Modeling

The goal of our study was to build models that can be used to predict whether there are resistance mutations within an RT sequence of a specific virus for different nucleoside analogues. Thus, we used drug-specific cutoffs for the IC_{50} ratios to separate the sequences into the classes “resistant” and “susceptible” for the different drugs. For 3TC and AZT, the cutoff was set to 3, for d4T and ddI it was set to 1.5, and for ABC it was set to 2. This means that sequences having a corresponding IC_{50} ratio above or equal to the cutoff are defined as “resistant” (see Table 1), whereas sequences having an IC_{50} ratio below the cutoff are defined as “susceptible“. For instance, an RT sequence with an ratio of 10.3 for 3TC is defined as “resistant”, whereas another RT sequence with an ratio of 2.5 is defined as “susceptible”.

In some studies, e.g. Rhee *et al.* (2006), a third class was defined as “intermediate resistant”. Nevertheless, since “intermediate resistant” strains are somehow resistant, too, we simply subsumed those sequences under the “resistant” category. The resulting classes are rather balanced: for each drug (except for 3TC and ABC), around 50% of the RT sequences belong to the class “resistant” and 50% to the class “susceptible”. Concretely, the fraction of RT sequences categorized as “resistant” are, respectively, 50.81%, 47.72%, 45.44%, 68.57% and 70.52% for ddI, AZT, d4T, 3TC and ABC.

As already demonstrated in several protein classification studies, e.g. (Chowriappa *et al.*, 2008; Dybowski *et al.*, 2010; Heider *et al.*, 2009, 2010), the use of physico-chemical properties and especially the use of hydrophobicity characteristics (Kyte and Doolittle, 1982), lead to very good prediction results. Thus, we encoded the RT protein sequences into hydrophobicity vectors and normalized them to length 240, which represents the average sequence length in the dataset, using Interpol (Heider and Hoffmann, 2011). Thus, we eventually end up with a dataset consisting of 614 instances (RT sequences), each of which is characterized in terms of a normalized hydrophobicity vector of length 240. Moreover, each instance is associated with five binary class labels, namely resistance for ddI, AZT, d4T, 3TC and ABC. Our goal, now, is to train a model that generalizes beyond these examples, i.e., that allows for accurately predicting each of the five outputs on the basis of any normalized hydrophobicity vector given as input information.

2.3 Multilabel Classification

The above problem obviously falls in the realm of (binary) *classification*, a well-established and thoroughly explored subfield of statistics and machine learning. In fact, the arguably most simple way to solve it is to train one binary classifier for each of the five outputs, thereby splitting the original *multi-output* problem into five *single-output* problems. Each of these problems can then be solved individually, using the large repertoire of existing methods for binary classification.

This approach has an important disadvantage, however: It cannot take any advantage of possible dependencies between the different outputs. Modeling and exploiting such dependencies in order to improve prediction accuracy is one of the key goals of *multilabel classification* (MLC). Intuitively, if the

[2] In some cases, albeit less frequently, it could also lead to a re-sensitization for other drugs.

value of one output may (statistically) depend on the value of others, then predicting all outputs simultaneously should indeed be better than predicting them separately. This is the main argument against simple decomposition techniques like the one proposed above, called *binary relevance* (BR) learning in the context of MLC.

More formally, let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_m\}$ be a finite set of class labels (in our case the resistance for the five drugs), and let $\mathcal{X}$ be an instance space (in our case the 240-dimensional hydrophobicity vectors). An MLC task assumes a training set $S = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, generated independently and identically according to a probability distribution $\mathbf{P}(\mathbf{X}, \mathbf{Y})$ on $\mathcal{X} \times \mathcal{Y}$. Here, $\mathcal{Y}$ is the set of possible label combinations, i.e., the power set of $\mathcal{L}$. To ease notation, we define a label combination $\boldsymbol{y}$ as a binary vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, in which $y_j = 1$ indicates the presence (relevance) and $y_j = 0$ the absence (irrelevance) of λ_j. Under this convention, the output space is given by $\mathcal{Y} = \{0,1\}^m$. The goal in MLC is to induce from S a model $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$ that correctly predicts the subset of relevant labels for unlabeled query instances $\boldsymbol{x} \in \mathcal{X}$. Recall that, in our context, "relevance of a label" stands for "resistance against a drug"; thus, a prediction $\boldsymbol{y} = (1, 0, 1, 0, 0)$ would suggest that the instance (RT sequence) at hand is resistant against the first and the third drug while being susceptible to the others.

Performance metrics The prediction of label subsets (vectors) instead of single labels suggests different types of performance metrics for MLC. Commonly used examples of such metrics, that also seem to be meaningful in the context of our application, include the Hamming loss, the subset 0/1 loss and the F-measure. Let $\mathcal{X}_{test} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of test instances. Moreover, let $\boldsymbol{y}_{i\bullet} = (y_{i,1}, \ldots, y_{i,m}) \in \mathcal{Y}$ be the labeling of test instance $\boldsymbol{x}_i$, where $y_{i,j}$ is the value of label λ_j for $\boldsymbol{x}_i$, and denote by $\hat{\boldsymbol{y}}_{i\bullet} = (\hat{y}_{i,1}, \ldots, \hat{y}_{i,m})$ the corresponding prediction produced by the classifier. The Hamming loss of the prediction $\hat{\boldsymbol{y}}_{i\bullet}$ is then defined as the fraction of labels whose relevance is incorrectly predicted:[3]

$$L_H(\boldsymbol{y}_{i\bullet}, \hat{\boldsymbol{y}}_{i\bullet}) = \frac{1}{m} \sum_{j=1}^{m} [\![y_{i,j} \neq \hat{y}_{i,j}]\!] \in [0,1] \tag{1}$$

The subset 0/1 loss simply checks whether the complete label subset is predicted correctly or not:

$$L_S(\boldsymbol{y}_{i\bullet}, \hat{\boldsymbol{y}}_{i\bullet}) = [\![\boldsymbol{y}_{i\bullet} \neq \hat{\boldsymbol{y}}_{i\bullet}]\!] \in \{0,1\} \tag{2}$$

The F-measure essentially corresponds to the harmonic mean of the precision and recall of the prediction; it is defined as follows:

$$L_F(\boldsymbol{y}_{i\bullet}, \hat{\boldsymbol{y}}_{i\bullet}) = \frac{2 \sum_{j=1}^{m} y_{i,j} \hat{y}_{i,j}}{\sum_{j=1}^{m} y_{i,j} + \sum_{j=1}^{m} \hat{y}_{i,j}} \in [0,1] \enspace , \tag{3}$$

where $0/0 = 1$ by definition. The above metrics are used to evaluate the prediction $\hat{\boldsymbol{y}}_{i\bullet}$ for an individual instance $\boldsymbol{x}_i$, i.e., they are computed *instance-wise*. Correspondingly, in an experimental study, the average accuracy would be reported as the average over all instances $\boldsymbol{x}_i$ in the test data $\mathcal{X}_{test}$.

Apart from that, another option is of course to report accuracy in a *label-wise* manner, namely to compute standard performance metrics such as classification rate (percentage of correct predictions) or AUC (Area under the ROC Curve) separately for each label $\lambda_i \in \mathcal{L}$. Note that some metrics can be computed instance-wise as well as label-wise. For example, the label-wise version of the F-measure is given by

$$L_F(\boldsymbol{y}_{\bullet j}, \hat{\boldsymbol{y}}_{\bullet j}) = \frac{2 \sum_{i=1}^{N} y_{i,j} \hat{y}_{i,j}}{\sum_{i=1}^{N} y_{i,j} + \sum_{i=1}^{N} \hat{y}_{i,j}} \in [0,1] \enspace , \tag{4}$$

where $\boldsymbol{y}_{\bullet j} = (y_{1,j}, \ldots, y_{N,j})$ is the vector of values for the label λ_j and $\hat{\boldsymbol{y}}_{\bullet j}$ the corresponding vector of predictions.

2.4 Classifier Chains

Until now, several methods for multilabel classification have been proposed in the literature. Here, we shall focus on a method called *classifier chains* (Read *et al.*, 2011), which, despite having been introduced only lately, already enjoys great popularity. This is arguably due to the fact that it is based on a simple and elegant yet effective idea for capturing label dependencies.

The classifier chains (CC) method learns m binary classifiers (each one dealing with the binary relevance problem associated with one label) linked along a chain, each time extending the feature space by all previous labels in the chain. For instance, if the chain follows the order $\lambda_1 \rightarrow \lambda_2 \rightarrow \ldots \rightarrow \lambda_m$, then the classifier h_j responsible for predicting the relevance of λ_j is of the form

$$h_j : \mathcal{X} \times \{0,1\}^{j-1} \longrightarrow \{0,1\} \enspace . \tag{5}$$

The training data for this classifier consists of (expanded) instances $(\boldsymbol{x}_i, y_{i,1}, \ldots, y_{i,j-1})$ labeled with $y_{i,j}$, that is, original training instances $\boldsymbol{x}_i$ supplemented by the relevance of the labels $\lambda_1, \ldots, \lambda_{j-1}$ preceding λ_j in the chain. Thus, the classifier h_j supposed to predict the label of class λ_j makes use of the preceding labels as additional input information, thereby capturing possible dependencies between the labels. Theoretically, the CC approach can be motivated by the product rule of probability (Dembczyński *et al.*, 2010):

$$\mathbf{P}(\boldsymbol{y} \,|\, \boldsymbol{x}) = \prod_{k=1}^{m} \mathbf{P}(y_k \,|\, \boldsymbol{x}, y_1, \ldots, y_{k-1}) \tag{6}$$

Note that, for training the classifier (5), any standard method for binary classification can be used (logistic regression, decision trees, support vector machines, etc.).

At prediction time, when a new instance $\boldsymbol{x}$ needs to be labeled, a label vector $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_m)$ is produced by successively querying each classifier h_j. Note, however, that the inputs of these classifiers are not well-defined, since the supplementary attributes $y_{i,1}, \ldots, y_{i,j-1}$ are not available. These missing values are therefore replaced by their respective predictions: y_1 used by h_2 as an additional input is replaced by $\hat{y}_1 = h_1(\boldsymbol{x})$, y_2 used by h_3 as an additional input is replaced by $\hat{y}_2 = h_2(\boldsymbol{x}, \hat{y}_1)$, and so forth. Thus, the prediction $\boldsymbol{y}$ is of the form

$$\boldsymbol{y} = \big(h_1(\boldsymbol{x}),\, h_2(\boldsymbol{x}, h_1(\boldsymbol{x})),\, h_3(\boldsymbol{x}, h_1(\boldsymbol{x}), h_2(\boldsymbol{x}, h_1(\boldsymbol{x}))), \ldots \big)$$

The process of training a classifier chain and using it for prediction is illustrated in Figure 1.

2.5 Ensembles of Classifier Chains

Realizing that the order of labels in the chain may influence the performance of the classifier, and that an optimal order is hard to anticipate, Read *et al.* (2011) propose the use of an ensemble of CC classifiers. This approach combines the predictions of different random orders and, moreover, uses a different sample of the training data to train each member of the ensemble. *Ensembles of classifier chains* (ECC) have been shown to increase prediction performance over CC by effectively using a simple voting scheme to aggregate predicted relevance sets of the individual chains: For each label λ_j, relevance is predicted by thresholding the proportion $\hat{w}_j$ of classifiers predicting $y_j = 1$ at a level t, i.e., $\hat{y}_j = [\![\hat{w}_j \geq t]\!]$.

3 RESULTS AND DISCUSSION

The major goal of our experimental study was to provide empirical evidence for the conjecture that capturing statistical dependencies between HIV-1 drugs is instrumental in learning classifiers for resistance prediction. Dependencies of that type are biologically plausible and suggested by the observation of cross-resistance; besides, they are also confirmed by our data: Table 2 shows the pairwise associations between the binary class labels (drugs),

[3] For a predicate P, the expression $[\![P]\!]$ evaluates to 1 if P is true and to 0 if P is false.

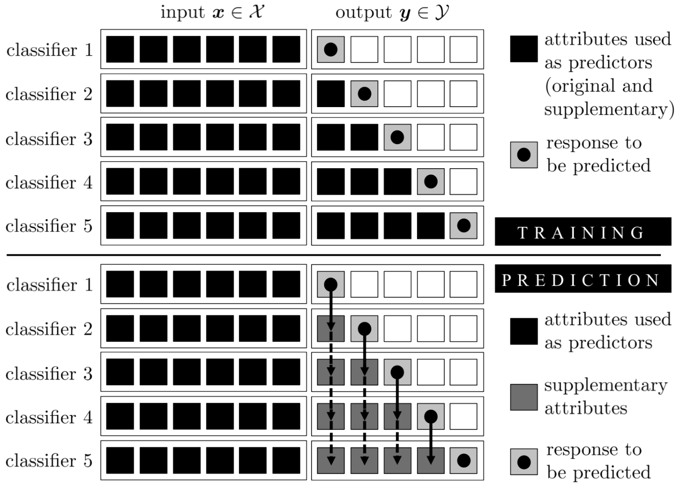

Fig. 1. Illustration of the classifier chains approach: training phase (above) and prediction phase (below).

expressed in the form of the phi coefficient.[4] As can be seen, the dependency between the resistance for different drugs is positive throughout and specifically high for the two pairs 3TC/ABC and AZT/d4T. This observation is in perfect agreement with attribute importance analyses on the basis of random forest classifiers that were trained for each class individually, which are in partial agreement with recent expert-defined resistance mutations (Johnson *et al.*, 2011) and other computational approaches, e.g. Kierczak *et al.* (2010). As can be seen in Figure 2, for 3TC as well as ABC, the most important sequence positions (often selected as discriminative attributes by the classifiers) are found in the C-terminal part with the highest peak at position 184. For the other drugs, namely AZT, d4T and ddI, the highest peaks are spread at the C-terminal as well as at the N-terminal part, e.g. 41, 70, 210 and 215. Some nucleoside analogues resistance patterns are well known (Stürmer *et al.*, 2007), e.g., the so-called thymidine analogue mutations (TAMs) at position 41, 65, 67, 70, 210, 215 and 219, leading to varying levels of AZT and d4T resistance (Garcia-Lerma *et al.*, 2003; Lafeuillade and Tardy, 2003; Antinori *et al.*, 2006). Another important mutation at position 184 is also reflected in the importance analyses. The mutation M184V is associated with high-level 3TC resistance as well as with ABC resistance. For ABC resistance also mutations at positions 65, 74 and 115 could be found during ABC therapy. Moreover, mutation patterns at position 151 in combination with mutations at position 62, 69, 75, 77 and 116 are also associated with high-level resistance against AZT, 3TC and ABC (Sirivichayakul *et al.*, 2003). Interestingly, position 65, which is associated with a broad cross-resistance in almost all NAs except for AZT, has also a high importance for the AZT classification. Nevertheless, random forest importance analyses have some limitations, as they only estimate the importance of a sequence position for the classification, but do not provide information whether a specific sequence position is positively or negatively associated with resistance; moreover, they do not provide information about interacting sequence positions that contribute to resistance. For a comprehensive structural analysis and interpretation of resistance mutations in RT see (Kierczak *et al.*, 2010). Interestingly, phylogenetic analyses of the sequences using a neighbor-joining approach (Gouy *et al.*, 2010) as well as principal component analysis on the distance matrix showed that the sequences cannot be easily separated into the different resistance classes based only on the sequence information (see supplementary Figure 1 and supplementary Figure 2).

Table 2. Values of the phi coefficient, a measure of association that ranges between -1 (perfect negative dependency) and $+1$ (perfect positive dependency).

	3TC	ABC	AZT	D4T	DDI
3TC	1.0	0.824	0.274	0.364	0.618
ABC	0.824	1.0	0.381	0.489	0.614
AZT	0.276	0.381	1.0	0.804	0.392
d4T	0.364	0.489	0.804	1.0	0.538
ddI	0.618	0.614	0.392	0.538	1.0

Table 3. Average classification rate of logistic regression models trained on different input information (original and supplemented). The numbers are determined through 10-fold cross validation repeated 5 times. The best result per label is highlighted in bold font.

input	3TC	ABC	AZT	d4T	ddI
$\boldsymbol{x}$	0.821	0.764	0.689	0.702	0.667
$\boldsymbol{x}$ + 3TC	—	0.766	0.696	0.701	0.689
$\boldsymbol{x}$ + ABC	0.833	—	0.698	**0.758**	0.675
$\boldsymbol{x}$ + AZT	0.816	0.769	—	0.725	0.667
$\boldsymbol{x}$ + d4T	0.815	**0.797**	**0.742**	—	**0.694**
$\boldsymbol{x}$ + ddI	**0.852**	0.776	0.711	0.735	—

It is important to note, however, that a positive correlation between labels does not necessarily imply a benefit for prediction. In particular, while the above correlation is an *unconditional* measure of dependence between class labels, a multilabel classifier such as CC seeks to capture *conditional* dependencies, namely the dependence between class labels *given* the input information $\boldsymbol{x}$. Table 3 shows the average misclassification rates of classifiers (logistic regression) that have been trained for the individual class labels λ_i on different input information, namely (i) the original predictor variables $\boldsymbol{x}$ and (ii) this feature vector supplemented by the resistance information of one of the other drugs λ_j; thus, we simply assumed that λ_j was already known when λ_i needs to be predicted. As can be seen, 3TC benefits more from knowing ddI than from knowing ABC, and d4T benefits more from ABC than from AZT. Another possible effect that cannot be excluded and could have an influence on our findings is treatment history. Unfortunately, the treatment histories in the dataset are highly diverse. However, most of the patients have an unknown treatment history or have not been treated yet (see http://hivdb.stanford.edu). Thus we assume that treatment history might play only a minor role in our model.

The current study was related to the idea of classifier chains in so far as class labels λ_j are used as additional predictor variables for other labels λ_i. Here, however, we assumed the true values y_j of the additional predictor to be known, not only for training but also for prediction. In chaining, on the other hand, the true values y_i are only known in the training step, whereas for prediction, they have

[4] This coefficient is equal to the Pearson correlation for binary variables and is also closely connected to the χ^2 statistics.

Table 4. Performance of BR, CC and ECC in terms of instance-wise metrics (mean ± standard deviation), in brackets the rank. Logistic regression was used as base learner.

	Hamming loss	subset 0/1	F-measure
BR	.2695 ± .0235 (2)	.6905 ± .0519 (3)	.6741 ± .0420 (3)
CC	.2697 ± .0266 (3)	.6904 ± .0528 (2)	.6788 ± .0417 (2)
ECC	.2384 ± .0538 (1)	.6312 ± .0538 (1)	.7166 ± .0366 (1)

to be replaced by their estimates $\hat{y}_i$. Thus, although the last study confirms the potential benefit of label information for prediction purposes, it is not clear that label dependencies can indeed be exploited in a practically realistic setting where other labels are not known at prediction time.

3.1 The Effect of Chaining

To analyze the practical usefulness of classifier chaining, we compared the prediction accuracy of the following methods:

- Binary relevance (BR): A single binary classifier is trained independently for each of the five labels.
- Classifier chains (CC): The five classifiers are trained according to the CC approach outlined in Section 2.4. The chain was constructed by sorting the labels in decreasing order according to their individual (BR) prediction accuracy:[5]

$$\text{3TC} \rightarrow \text{ABC} \rightarrow \text{AZT} \rightarrow \text{d4T} \rightarrow \text{ddI}$$

- Ensembles of classifier chains (ECC): The ECC method described in Section 2.5 was implemented with 10 chains, each time choosing the order of labels at random. The threshold t was taken as $1/2$.

All methods were implemented with standard logistic regression as a base learner. Prediction performance was measured in terms of the Hamming loss (1), the subset 0/1 loss (2) and the F-measure (3) as instance-wise metrics, and the classification rate, the AUC and the F-measure (4) as label-wise metrics. Each of these metrics was estimated by means of a 10-fold cross validation repeated 5 times, and results are reported in terms of the mean values and the standard deviations. Moreover, we also indicate the ranking of the three methods, with the best performing method on rank 1 and the worst performing method on rank 3.

The results for the instance-wise metrics are summarized in Table 4, those for the label-wise metrics in Table 5. Although the differences are not always statistically significant, as can be seen from the standard deviations, the overall picture is very clear and obviously in favor of the chaining methods. In fact, chaining achieves systematic (albeit sometimes small) gains in comparison to standard binary relevance learning. Among the two chaining methods, ECC performs even stronger than CC and typically yields the best results.

Table 5. Performance of BR (top), CC (middle) and ECC (bottom) in terms of label-wise metrics (mean ± standard deviation), in brackets the rank. Logistic regression was used as base learner.

	classification rate	AUC	F-measure
3TC	.8192 ± .0512 (2)	.8394 ± .0638 (2)	.8623 ± .0441 (2)
ABC	.7524 ± .0543 (3)	.7551 ± .0613 (3)	.8176 ± .0478 (3)
AZT	.6960 ± .0590 (3)	.6963 ± .0631 (3)	.6819 ± .0534 (3)
d4T	.7004 ± .0483 (3)	.7119 ± .0621 (2)	.6613 ± .0748 (3)
ddI	.6846 ± .0657 (2)	.6779 ± .0888 (3)	.6820 ± .0769 (2)
3TC	.8192 ± .0512 (2)	.8394 ± .0638 (2)	.8623 ± .0441 (2)
ABC	.7584 ± .0538 (2)	.7602 ± .0592 (2)	.8211 ± .0469 (2)
AZT	.7004 ± .0578 (2)	.7025 ± .0612 (2)	.6837 ± .0574 (2)
d4T	.7021 ± .0616 (2)	.7107 ± .0650 (3)	.6665 ± .0790 (2)
ddI	.6716 ± .0450 (3)	.6819 ± .0620 (2)	.6701 ± .0525 (3)
3TC	.8403 ± .0548 (1)	.9119 ± .0472 (1)	.8814 ± .0425 (1)
ABC	.7980 ± .0440 (1)	.8566 ± .0390 (1)	.8541 ± .0375 (1)
AZT	.7488 ± .0565 (1)	.8282 ± .0543 (1)	.7378 ± .0549 (1)
d4T	.7211 ± .0515 (1)	.8115 ± .0506 (1)	.6874 ± .0683 (1)
ddI	.6999 ± .0569 (1)	.7761 ± .0576 (1)	.7058 ± .0514 (1)

Table 6. Performance of BR, CC and ECC in terms of instance-wise metrics (mean ± standard deviation), in brackets the rank. Random forests (of size 16) were used as base learner.

	Hamming loss	subset 0/1	F-measure
BR	.2159 ± .0298 (3)	.5775 ± .0476 (3)	.7455 ± .0366 (3)
CC	.2129 ± .0459 (2)	.5098 ± .0514 (2)	.7631 ± .0459 (2)
ECC	.1947 ± .0255 (1)	.4961 ± .0476 (1)	.7787 ± .0344 (1)

Table 7. Performance of BR (top), CC (middle) and ECC (bottom) in terms of label-wise metrics (mean ± standard deviation), in brackets the rank. Random forests (of size 16) were used as base learner.

	classification rate	AUC	F-measure
3TC	.8289 ± .0515 (2)	.8910 ± .0520 (2)	.8815 ± .0403 (2)
ABC	.8235 ± .0473 (3)	.8575 ± .0530 (3)	.8828 ± .0348 (3)
AZT	.7852 ± .0582 (2)	.8800 ± .0446 (3)	.7827 ± .0589 (3)
d4T	.7655 ± .0451 (3)	.8603 ± .0351 (3)	.7417 ± .0582 (3)
ddI	.7177 ± .0634 (2)	.7962 ± .0483 (3)	.7292 ± .0618 (2)
3TC	.8289 ± .0515 (2)	.8910 ± .0520 (2)	.8815 ± .0403 (2)
ABC	.8295 ± .0473 (2)	.8705 ± .0484 (2)	.8861 ± .0348 (2)
AZT	.7965 ± .0630 (2)	.8726 ± .0527 (3)	.7928 ± .0626 (2)
d4T	.7721 ± .0598 (2)	.8668 ± .0454 (2)	.7603 ± .0652 (2)
ddI	.7085 ± .0497 (3)	.7922 ± .0525 (3)	.7373 ± .0464 (2)
3TC	.8392 ± .0493 (1)	.8942 ± .0439 (1)	.8905 ± .0386 (1)
ABC	.8404 ± .0375 (1)	.8801 ± .0462 (1)	.8943 ± .0281 (1)
AZT	.8144 ± .0420 (1)	.8976 ± .0361 (1)	.8125 ± .0447 (1)
d4T	.7970 ± .0503 (1)	.8901 ± .0341 (1)	.7831 ± .0590 (1)
ddI	.7357 ± .0392 (1)	.8215 ± .0394 (1)	.7573 ± .0420 (1)

To make sure that the results are not too much influenced by the underlying base learner used by all methods, we repeated the same experiments with random forests (Breiman, 2001) instead of logistic regression. These two learners exhibit quite different properties. In particular, while logistic regression fits a linear decision boundary in the instance space, decision trees are much more flexible and able to model highly non-linear concepts; this flexibility is even increased by the aggregation of different trees in the random forest approach. Thus, it comes at no surprise that the performance of all methods is

[5] This is a commonly used rule of thumb, which is motivated by the observation that mistakes of a single classifier tend to be propagated along the rest of the chain (Senge *et al.*, 2013); consequently, strong classifiers should be placed at the top and poor ones more toward the end of the chain.

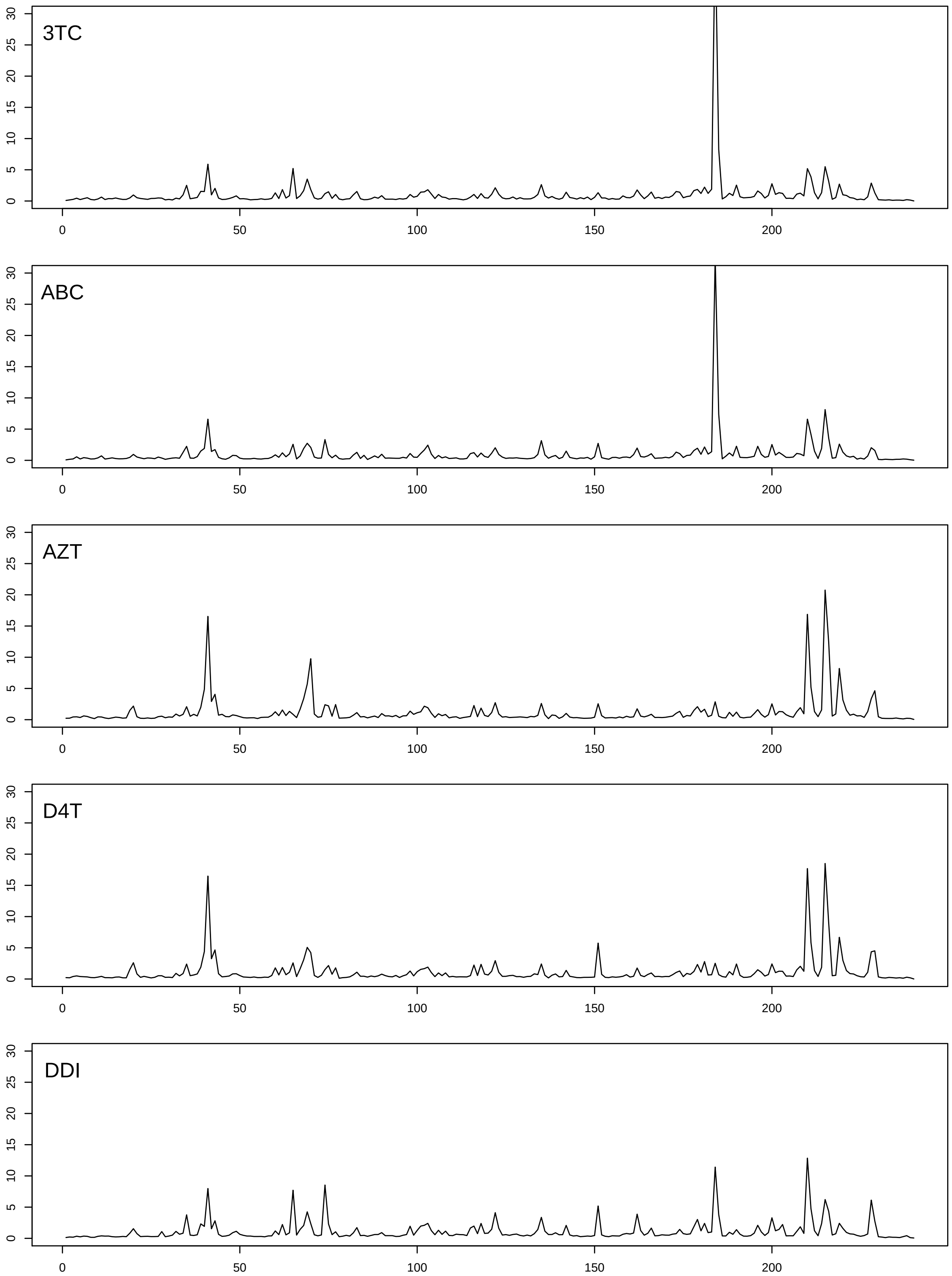

Fig. 2. Importance analyses from single classifiers
On the x-axis the sequence positions are shown, whereas the y-axis represents the sum of all decreases in Gini impurity. Feature importance for five single random forests was assessed using the sum of all decreases in Gini impurity, which has been shown to be more robust compared to the mean decrease in accuracy (Calle and Urrea, 2010).

in general improved. Nevertheless, in terms of relative comparison, the picture is more or less identical to the first experiment with logistic regression: Both chaining methods improve upon BR, with ECC being even better than CC (see Table 6 and Table 7).

4 CONCLUSION

We conclude with an affirmative answer to one of the main questions of our study, namely whether or not cross-resistance information can be used to improve overall accuracy in drug resistance prediction. By using multilabel classification methods, a relatively recent development in machine learning, we were able to exploit cross-resistance information for RT inhibitors. More concretely, our results are based on a specific multilabel classification method called classifier chains.

We consider these results as very promising and, therefore, intend to further explore this direction in future work. On the methodological side, we would like to try alternative MLC methods, including the probabilistic variant of classifier chains proposed by Dembczyński *et al.* (2010) but also approaches that are not based on the idea of chaining. As an interesting property of the former, let us mention that it does not only produce binary predictions, but proper probability estimates of single labels or label combinations. Predictions of that kind are quite interesting, not only for the minimization of various loss functions, but also for the purpose of representing uncertainty. Moreover, we want to include multi-class and regression models to be able to predict more classes, e.g. intermediate resistance, and even the resistance factors.

On the application side, our study has focused on nucleoside analogues so far, although a typical clinical treatment includes drugs from several classes. It might of course be interesting to test our approach for other types of antiretroviral drugs, for example non-nucleoside reverse transcriptase inhibitors, and for other target proteins, such as HIV-1 protease and corresponding protease inhibitors. By now, our approach is limited to very specialized treatment cases and thus is currently not well applicable in clinical settings. However, in the future we plan to adapt our approach for NNRTIs as well as for PIs. Moreover, all sequences used in the current study originated from subtype B strains, thus the results of our model might be misleading if it is applied to other subtypes.

SUPPLEMENTARY FIGURES

Figure 1 - Phylogenetic tree of the RT sequences

Analyses was performed using Seaview 4.2.5 (Gouy *et al.*, 2010). Sequence identifiers are as follows: sequence number - resistance against 3TC (1=yes, 0=no) - ABC (1=yes, 0=no) - AZT (1=yes, 0=no) - d4T (1=yes, 0=no) - ddI (1=yes, 0=no).

Figure 2 - Principal component analysis

PCA was performed on the distance matrix of the RT sequences. First two principal components (PC1 and PC2) are used for plotting. susceptible: virus is not resistant against 3TC/ABC/AZT/d4T/ddI; one: virus is resistant against one drug; two: virus is resistant against two drugs; three: virus is resistant against three drugs; four: virus is resistant against four drugs; five: virus is resistant against all analyzed drugs.

REFERENCES

Antinori, A., Trotta, M. P., Nasta, P., Bini, T., Bonora, S., Castagna, A., Zaccarelli, M., Quirino, T., Landonio, S., Merli, S., Tozzi, V., Perri, G. D., Andreoni, M., Perno, C. F., and Carosi, G. (2006). Antiviral efficacy and genotypic resistance patterns of combination therapy with stavudine/tenofovir in highly active antiretroviral therapy experienced patients. *Antivir Ther*, **11**(2), 233–243.

Breiman, L. (2001). Random forests. *Machine Learning*, **45**, 5–32.

Calle, M. L. and Urrea, V. (2010). Letter to the Editor: Stability of Random Forest importance measures. *Briefings in bioinformatics*, **12**(1), 86–89.

Chowriappa, P., Dua, S., Kanno, J., and Thompson, H. W. (2008). Protein structure classification based on conserved hydrophobic residues. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, **6**(4), 639–51.

Dembczyński, K., Cheng, W., and Hüllermeier, E. (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, pages 279–286.

Dybowski, J. N., Heider, D., and Hoffmann, D. (2010). Prediction of co-receptor usage of HIV-1 from genotype. *PLoS Comput Biol*, **6**(4), e1000743.

Garcia-Lerma, J. G., MacInnes, H., Bennett, D., Reid, P., Nidtha, S., Weinstock, H., Kaplan, J. E., and Heneine, W. (2003). A novel genetic pathway of human immunodeficiency virus type 1 resistance to stavudine mediated by the K65R mutation. *J Virol*, **77**(10), 5685–5693.

Gouy, M., Guindon, S., and Gascuel, O. (2010). SeaView version 4: A multiplatform graphical user interface for sequence alignment and phylogenetic tree building. *Mol Biol Evol*, **27**(2), 221–224.

Heider, D. and Hoffmann, D. (2011). Interpol: An R package for preprocessing of protein sequences. *BioData Min*, **4**, 16.

Heider, D., Appelmann, J., Bayro, T., Dreckmann, W., Held, A., Winkler, J., Barnekow, A., and Borschbach, M. (2009). A computational approach for the identification of small GTPases based on preprocessed amino acid sequences. *Technology in Cancer Research and Treatment*, **8**(**5**), 333–342.

Heider, D., Verheyen, J., and Hoffmann, D. (2010). Predicting Bevirimat resistance of HIV-1 from genotype. *BMC Bioinformatics*, **11**, 37.

Johnson, V. A., Calvez, V., Gnthard, H. F., Paredes, R., Pillay, D., Shafer, R., Wensing, A. M., and Richman, D. D. (2011). 2011 update of the drug resistance mutations in HIV-1. *Top Antivir Med*, **19**(4), 156–164.

Kierczak, M., Ginalski, K., Dramiński, M., Koronacki, J., Rudnicki, W., and Komorowski, J. (2009). A Rough Set-Based Model of HIV-1 Reverse Transcriptase Resistome. *Bioinform Biol Insights*, **3**, 109–127.

Kierczak, M., Dramiński, M., Koronacki, J., and Komorowski, J. (2010). Computational Analysis of Molecular Interaction Networks Underlying Change of HIV-1 Resistance to Selected Reverse Transcriptase Inhibitors. *Bioinform Biol Insights*, **4**, 137–146.

Kyte, J. and Doolittle, R. (1982). A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.

Lafeuillade, A. and Tardy, J.-C. (2003). Stavudine in the face of cross-resistance between HIV-1 nucleoside reverse transcriptase inhibitors: a review. *AIDS Rev*, **5**(2), 80–86.

Pennings, P. S. (2012). Standing Genetic Variation and the Evolution of Drug Resistance in HIV. *PLoS Computational Biology*, **8**(**6**), e1002527.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, **85**(3), 333–359.

Rhee, S.-Y., Taylor, J., Wadhera, G., Ben-Hur, A., Brutlag, D. L., and Shafer, R. W. (2006). Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proc Natl Acad Sci U S A*, **103**(46), 17355–17360.

Senge, R., del Coz, J. J., and Hüllermeier, E. (2013). On the problem of error propagation in classier chains for multi-label classification. In L. Schmidt-Thieme and M. Spiliopoulou, editors, *Data Analysis, Machine Learning and Knowledge Discovery. Proceedings of the 36th Annual Conference of the German Classification Society*, Hildesheim, Germany. Springer.

Sirivichayakul, S., Ruxrungtham, K., Ungsedhapand, C., Techasathit, W., Ubolyam, S., Chuenyam, T., Emery, S., Cooper, D., Lange, J., and Phanuphak, P. (2003). Nucleoside analogue mutations and Q151M in HIV-1 subtype A/E infection treated with nucleoside reverse transcriptase inhibitors. *AIDS*, **17**(13), 1889–1896.

Stürmer, M., Staszewski, S., and Doerr, H. W. (2007). Quadruple nucleoside therapy with zidovudine, lamivudine, abacavir and tenofovir in the treatment of HIV. *Antivir Ther*, **12**(5), 695–703.

Tripathi, K., Balagam, R., Vishnoi, N. K., and Dixit, N. M. (2012). Stochastic Simulations Suggest that HIV-1 Survives Close to Its Error Threshold. *PLoS Computational Biology*, **8**(**9**), e1002684.

Tsoumakas, G. and Katakis, I. (2007). Multi label classification: An overview. *International Journal of Data Warehouse and Mining*, **3**(3), 1–13.