Capreomycin Resistance Prediction in Two Species of *Mycobacterium* Using a Stacked Ensemble Method

Abu Sayed Chowdhury[1], Ehdieh Khaledian[1], and Shira L. Broschat[1,2,3]

[1]School of Electrical Engineering and Computer Science,

[2]Paul G. Allen School for Global Animal Health,

[3]Department of Veterinary Microbiology and Pathology,

Washington State University,

Pullman, Washington, United States

E-mail: {abu.chowdhury, ehdieh.khaledian, shira}@wsu.edu

Corresponding author: Abu Sayed Chowdhury

Email: abu.chowdhury@wsu.edu

Phone number: +1 509-715-9463

Address: School of Electrical Engineering and Computer Science, Washington State University,

P.O. Box 642752, Pullman, Washington 99164-2752, United States

Abstract

Aims: Predicting bacterial resistance provides valuable information that can assist in clinical decisions. With recent advances in whole genome sequencing technology, the detection of antibiotic resistance (AR) proteins directly from genomic data is becoming feasible. AR genes/proteins can be identified using best-hit methods that work by comparing candidate sequences with known AR genes in public databases. However, these approaches may fail to detect resistance genes with sequences that differ significantly from known sequences. Our goal is to develop a machine learning technique to accurately predict capreomycin resistance in *Mycobacteria* with low false discovery rates.

Methods and Results: We present a stacked ensemble learning model as an alternative to traditional DNA sequence alignment-based methods using optimal features generated from the physicochemical, evolutionary, and secondary structure properties of protein sequences. We train logistic regression, C5.0, and support vector machine (SVM) algorithms as our base classifiers, and our stacked ensemble predictors combine the results from the base classifiers to achieve higher accuracy. Compared with our most accurate base classifier (SVM), our most accurate stacked ensemble predictor increases training accuracy by 2.43%. Our stacked ensemble predictors achieve test accuracy up to 81.25%.

Conclusions: We developed a stacked ensemble model to predict capreomycin resistance for *Mycobacteria* with an accuracy greater than 80% using protein sequences with sequence similarity ranging between 10% and 70%. This performance cannot be achieved with best-hit methods due to differences in sequence similarity.

Significance and Impact of the Study: Today an estimated one-half million cases of multidrug-

resistant (MDR) and extensively-drug resistant (XDR) tuberculosis (TB) occur annually worldwide at a great cost. Because capreomycin is a second-line drug used to treat drug resistant TB, the ability to use a machine learning approach to classify capreomycin-resistant TB in a timely manner is crucial for the successful treatment of MDR or XDR TB.

Keywords: Capreomycin resistance; antibiotic resistance; tuberculosis; physicochemical features; secondary structure features; feature selection; ensemble learning; machine learning

Introduction

The introduction of antibiotics has saved millions of lives over the past decades. However, their widespread use has resulted in the rapid emergence of antibiotic resistance (AR) (Aminov 2009, Dijkstra et al. 2018). AR impacts the safety and efficacy of treatment options that require the use of antibiotics and leads to the risk of fatal outcomes. Therefore, various strategies such as reduction in antibiotic usage or alternative second-line antibiotics have been suggested. One of the major AR bacteria that is a global health threat is *Mycobacterium tuberculosis* (MTB). MTB bacteria cause the life-threatening, airborne infectious disease, tuberculosis (TB). TB has infected almost one-third of the world population. It usually attacks lungs, but it can affect the brain, kidneys, and spine as well. It can be multidrug resistant (MDR) due to resistance to first-line TB antibiotics such as isoniazid and rifampin, and it also can be extensively drug resistant (XDR) to second-line antibiotics such as capreomycin, amikacin, and kanamycin (Gygli et al. 2017). A species known as *M. yongonense* TKK-01-0059 (originally *M.* sp. TKK-0100059) ,which causes pulmonary infections in humans, has been found to be resistant to capreomycin. This species has related strains *M. yongonense* 05-1390(T), Asan 36912, and Asan 36527 which may also be resistant to capreomycin, but their genomes are not available for inspection (Mnyambwa et al.

2018).

AR detection in *Mycobacterium* spp. can provide valuable information for improving clinical decisions in the treatment of the infectious diseases they cause, such as identifying drugs with the potential for success or eliminating a drug that is ineffective due to antibiotic resistance. As such, AR prediction has become a fundamental challenge for improving health care. Traditional *in vitro* culture-based tests for AR phenotype detection take many days to provide susceptibility test results (Chen et al. 2018), and phenotyping using the gold-standard proportion method can take up to two months (Niehaus et al. 2014). A molecular approach is an alternative to conventional culture-based detection; however, it often fails to detect rare gene variants and provides limited biomarkers for AR detection. Whole genome sequencing is currently a feasible approach for the detection of target mutations, and sequence alignment techniques such as best-hit approaches can be applied to detect AR genes using sequence identity in existing online databases (Kleinheinz et al. 2014, Davis et al. 2016, Yang et al. 2016). These methods work well for finding known and highly conserved AR sequences and have low probability of producing false positives, *i.e.*, predicting non-AR genes as AR genes (Forsberg et al. 2014). However, they may produce high false negative rates when they fail to identify AR genes that have lower sequence identity with known AR genes (Xavier et al. 2016, Yang et al. 2016, Mcarthur et al. 2017). Machine learning can be a useful alternative computational framework for identifying AR phenotypes accurately using features, i.e., characteristics of known AR protein sequences, found in the genomic data regardless of sequence similarity (Niehaus et al. 2014, Santerre et al. 2016). Machine learning models require a large number of genomes with AR metadata to make a strong prediction. Databases such as the Pathosystems Resource Integration Center (PATRIC) (Wattam

et al. 2013, Davis et al. 2016, Wattam et al. 2016) and the Antibiotic Resistance Genes Database (ARDB) (Lio and Pop 2008) currently provide genomes with AR metadata. In some recent studies (Her and Wu 2018; Kavvas et al. 2018; Moradigaravand et al. 2018), machine learning tecnhiques were applied to find resistance to varieties of antibiotics from whole genome sequences using databases of known AR genes. These studies first constructed a pan-genome with identifying core and accessory gene clusters, then computed features such as presence-absence patterns and the population structures of gene clusters, and finally applied a machine learning algorithm to identity putative AR genes using the extracted features. However, these approaches used a small number of genetic features for AR gene prediction.

In this work, we developed a machine learning approach to detect capreomycin resistance in two species of *Mycobacterium* which may be extended to other species. We collected protein sequences for both antibiotic resistance and antibiotic susceptibility with sequence similarity ranging between 10% and 70% from the two databases mentioned above. We then extracted potential features including physicochemical, evolutionary, and structural properties and evaluated these features to find the optimal ones for prediction. The advantage of using protein sequences rather than gene sequences is that a greater number of features are discernible. We applied a stacked ensemble learning approach to predict capreomycin resistance. To our knowledge, this is the first time stacked ensemble learning has been used for this task. Our stacked ensemble predictor trained a learning algorithm to combine the results from logistic regression, C5.0, and SVM algorithms. Finally, we divided our data into training and test sets to validate the performance of our model. We also compared the performance of our classifier with BLASTp results.

Materials and Methods

In this section, we discuss the details of data collection, feature generation from protein sequences, and feature evaluation to find an optimal feature set.

Data Collection

To collect capreomycin resistance and susceptibility protein sequences, we searched the PATRIC (Davis et al. 2016) database and ARDB (Lio and Pop 2008). The Pathosystems Resource Integration Center (PATRIC) is a resource center designed to store and integrate a variety of data types, e.g., genomic sequences, three-dimensional protein structures, and sequence typing data. The Antibiotic Resistance Genes Database (ARDB) is a manually curated database that provides a centralized collection of information on antibiotic resistance. We retrieved 3,366 resistance and 5,411 susceptibility protein sequences. The vast majority of these were duplicate sequences. We applied CD-HIT (Li and Godzik 2006, Fu et al. 2012) to remove sequences with similarity greater than or equal to 70% from both sets of protein sequences. Our final result was 44 resistance and 36 susceptibility sequences (Supplementary Tables: Table S1, and Table S2). We used this dataset to train and test our classification model.

Feature Generation

We performed a thorough literature search to determine potential protein features to consider for use with our machine learning model. A number of researchers have suggested techniques for extracting features from proteins based on composition, physical and chemical characteristics, and secondary structure (Ding and Dubchak 2001, Cai et al. 2003, Zhang et al.

2011, Liu et al. 2013, Wei et al. 2015, Li et al. 2016, Eslami et al. 2018). In this work, we first computed the amino acid composition for each sequence. Because there are 20 different amino acids, this gives 20 features, each representing a fraction of the amino acids within a protein sequence of length n. Next, we applied the composition, transition, and distribution (CTD) model to obtain global physicochemical information from the protein sequences (Dubchak et al. 1995, Dubchak et al. 1999). Amino acids are grouped into three classes using hydrophobicity properties. These classes are polar, neutral, and hydrophobic. Arginine, lysine, glutamic acid, aspartic acid, glutamine, and asparagine belong to the polar class. The neutral group contains glycine, alanine, serine, threonine, proline, histidine, and tyrosine, and the remaining amino acids cysteine, leucine, valine, isoleucine, methionine, phenylalanine, and tryptophan are in the hydrophobic group. Composition C represents the number of amino acids of each class in a protein sequence divided by the total number of amino acids. Transition T gives the percent frequency of one amino acid class followed by amino acids of another class. The distribution D measures the fraction of the whole sequence where the first, 25%, 50%, 75%, and 100% of the amino acids of a class are located.

The CTD model considers eight physicochemical amino acid properties, namely, hydrophobicity, normalized van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure, and solvent accessibility. Because there are three classes for amino acids, we obtain three features for composition, three features for transition, and $3 \times 5 = 15$ features for distribution for each of the eight physicochemical amino acid properties. Thus, in total, the CTD model gives $(3 + 3 + 15) \times 8 = 168$ features for each protein sequence.

We also considered the position-specific scoring matrix (PSSM) to obtain evolutionary-

relevant features from protein sequences. To accomplish this, we generated the PSSM from protein sequences using PSI-BLAST (Altschul et al. 1997) and then calculated transition scores between neighboring amino acids to generate a 400D feature vector for each sequence. Let $M = (m_{i,j}) \in \mathbb{R}^{n\times 20}$ be the PSSM matrix where $m_{i,j}$ is the entry in the ith row and jth column of M. The transition $t_{x,y}$ between neighboring amino acids x and y can be computed using Eq. (1).

$$t_{x,y} = \sum_{k=1}^{n-1} m_{k,x} m_{k+1,y} \quad (1)$$

Finally, we generated features from the secondary structure of a protein sequence which has proven to be very efficient in protein fold recognition. We applied PSIPRED (Jones 1999) which identifies the secondary structure of a protein sequence based on PSI-BLAST. The output of PSIPRED is a secondary structure sequence composed of $H(\alpha$-helix), $E(\beta$-strand), $C(\gamma$-coil), and an $n \times 3$ structure probability matrix where n is the number of amino acids in a protein sequence. Suppose $SPM = (p_{i,j}) \in \mathbb{R}^{n\times 3}$ is the structure probability matrix where each entry in the matrix represents the probability of an amino acid to be in one of the states H, E, or C. Then we extracted features from the secondary structure sequence and SPM as described in existing works (Kurgan and Homaeian 2006, Kurgan et al. 2008, Liu and Jia 2010, Zhang et al. 2011, Wei et al. 2015). We generated location-oriented features CMV_H, CMV_E, and CMV_C by measuring the spatial arrangements of the three states—H, E, and C—using Eqs. (2), (3), and (4), respectively.

$$CMV_H = \frac{\sum_{j=1}^{T_H} I_{H_j}}{n(n-1)} \quad (2)$$

$$CMV_E = \frac{\sum_{j=1}^{T_E} I_{E_j}}{n(n-1)} \quad (3)$$

$$CMV_C = \frac{\sum_{j=1}^{T_C} I_{C_j}}{n(n-1)} \quad (4)$$

$$Max_H = \frac{\max\{n_H\}}{n} \quad (5)$$

$$Max_E = \frac{\max\{n_E\}}{n} \quad (6)$$

where I_{H_j}, I_{E_j}, and I_{C_j} are the position indexes, and T_H, T_E, and T_C are the total number of states for H, E, and C, respectively. We computed the normalized maximum spatial consecutive E and H states in the secondary structure sequence using Eqs. (5) and (6) where Max_H and Max_E are the maximum lengths of consecutive H and E states in the secondary structure sequence, respectively. We also computed the segmented sequences by ignoring the coil segments in the secondary structure. For example, let $S = CCCEEEEEHHCCEECCCCHHHHCCEEE$ be the secondary structure sequence. If we ignore the coil segments C and combine consecutive H's and E's as H' and E', then we obtain $S' = E'H'E'H'E'$. As α and β are usually separated and interspersed in α/β proteins and $\alpha+\beta$ proteins, respectively (Zhang et al. 2011, Wei et al. 2015), we measured the frequency of $E'H'E'$ (denoted as $f_{E'H'E'}$) using Eq. (7) where $T_{E'H'E'}$ is the total number of occurrences of $E'H'E'$ in S', and the length of S' is $L_{S'}$.

$$f_{E'H'E'} = \frac{T_{E'H'E'}}{L_{S'}-2} \quad (7)$$

After extracting features from the secondary structure sequence, we retrieved features from the structure probability matrix SPM. We obtained three global information features from the SPM using Eqs. (8), (9), and (10).

$$F_{g_1} = \frac{1}{n}\sum_{j=1}^{n} p_{j,1} \quad (8)$$

$$F_{g_2} = \frac{1}{n}\sum_{j=1}^{n} p_{j,2} \quad (9)$$

$$F_{g_3} = \frac{1}{n}\sum_{j=1}^{n} p_{j,3} \quad (10)$$

Here, $p_{j,1}$, $p_{j,2}$, and $p_{j,3}$ are the jth probability values of the SPM for H, E, and C, respectively. We also captured local information features by subdividing the SPM into smaller δ matrices each with $\lfloor\frac{n}{\delta}\rfloor$ rows and three columns. We chose $\delta = 8$ and computed 3D features for a particular sub-matrix q as shown in Eq. (11). Note that $F_{local_q}(1)$, $F_{local_q}(2)$, and $F_{local_q}(3)$ are computed using the same formulas given in Eqs. (8), (9), and (10), respectively. In this way, we found a total of 24 features with local information.

$$F_q = \left\{F_{local_q}(1), F_{local_q}(2), F_{local_q}(3)\right\}, 1 \le q \le \delta \quad (11)$$

In summary, we obtained a 20D feature vector based on amino acid composition, a 168D feature vector of physicochemical properties using a CTD model, a 400D feature vector using evolutionary information from the PSSM, a 6D feature vector having location oriented features of secondary structure, and a 27D feature vector having global and local information from the structure probability matrix (SPM) of a protein sequence. We merged all of these features to obtain a 621D high-dimensional feature vector as summarized in Table 1. In the next section, we explain how we removed redundant and noisy information to obtain our final feature set.

Dimensionality Reduction

To reduce the dimension of our features, we first calculated Pearson's correlation coefficient between features, that is, we measured the linear correlation between two feature vectors u and v. Pearson's correlation coefficient $\rho_{u,v}$ is calculated using

$$\rho_{u,v} = \frac{E[(u-\mu_u)(v-\mu_v)]}{\sigma_u\sigma_v} \quad (12)$$

where E is the expectation, μ_u and μ_v are the mean values, and σ_u and σ_v are the standard deviations of u and v, respectively. The $\rho_{u,v}$ value can be $[-1, +1]$ where the signs '$+$' and '$-$' indicate positive and negative correlations between features. We considered the absolute value of $|\rho_{u,v}|$ to estimate the correlation between features. When the value of $|\rho_{u,v}|$ is large, then two features are highly correlated, and we can consider either of the features for classification purposes. We set a threshold value of 0.95, keeping one feature and eliminating the other. In this way, we removed redundant information in the feature set and reduced the number of features from 621 to 392.

Next we performed a standard t-test using our dataset of resistant and susceptible protein sequences together with our feature set. The t-test is used to check whether two populations are significantly different based on a statistical measurement. In the t-test, the null hypothesis is that there is no difference between the two populations (resistance and susceptibility protein sequences), and we measure the p-values for all the features to see whether we can reject the null hypothesis. A low p-value for a feature means that the feature is important for predicting capreomycin resistance proteins, and we can reject the null hypothesis. To further reduce the dimension of the feature set, we set a threshold p-value of 0.05, eliminating features with $p > 0.05$. Figure 1 depicts the p-values for selected features both in linear and logarithmic scales. We found that the top selected features were from the distribution measurements of the CTD model which clearly indicates the importance of this feature group in our prediction model.

We utilized the R *stats* package (version 3.4.3) to measure correlation and to perform t-tests for all the features. Following the filtering of features through this t-test step, our feature vector was further reduced from 392D features to 336D. We considered this 336D feature vector as our

optimal feature set. In the following section, we describe how we used our optimal features to develop a classification model for recognizing capreomycin resistance proteins.

Stacked Ensemble Model

For our machine learning model, we applied stacking (also called stacked generalization) to increase accuracy. We measured the accuracy of a classifier using Eq. (13) where FP, FN, TP, and TN are false positive, false negative, true positive, and true negative, respectively.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (13)$$

Stacking is an efficient ensemble machine learning technique that takes the predictions of different machine learning algorithms in the first layer as inputs to a second layer (Wolpert 1992, Breiman 1996, Van der Laan et al. 2007, Chowdhury et al. 2015). It combines the predictions of the first-layer models together using another classifier as depicted in Figure 2. In the stacked ensemble model, the second layer classifier (also called a meta-classifier or meta-learner) is trained using the predictions of the first-layer models (also called base classifiers) to make an optimal combination of the predictions.

As shown in Figure 2, we used three different models as base classifiers in our first layer: logistic regression via a generalized linear model (GLM), a C5.0 model, and a support vector machine (SVM). The GLM is a statistical linear model which generalizes multiple linear regression for more than one dependent variable (Madsen and Thyregod 2010). C5.0 is an algorithm for creating decision trees, which supports boosting (Golmah 2014). An SVM is a supervised learning model formally characterized by a separating hyperplane that divides binary data (Scholkopf 2001). These three classifiers can accommodate multiple variables, including mixtures of categorical and continuous variables. They accept the same dataset as input and provide

individual prediction outputs PG, PC, and PS, respectively. The meta-classifier in the second layer delivers the final prediction PM based on the predictions of the base classifiers PG, PC, and PS. We tested several different meta-classifiers: a generalized linear model (GLM), linear discriminate analysis (LDA), and a random forest (RF). LDA is a classification method that searches for a linear combination of variables that best divides two classes. It is mathematically robust and often produces models with accuracy as high as more complex classifiers (Balakrishnama and Ganapathiraju, 1998). An RF is an ensemble of decision trees. It uses a bagging method to combine the ensemble learning methods in order to obtain a more accurate and stable prediction and to improve the overall result (Breiman 2001).

Note that in order to combine the results of the base classifiers, the pairwise correlations between their predicted outputs should be low. If they are low (< 0.75), the stacked ensemble model is likely to have better accuracy than a base classifier alone. If the correlations are not low, the base models will provide the same or very similar predictions making the ensemble of classifiers worthless.

To build the stacked ensemble model, we used six R packages: *mlbench* (version 2.1-1), *caret* (version 6.0-79), *caretEnsemble* (version 2.0.0), *C50* (version 0.1.2), *plyr* (version 1.8.4), and *randomForest* (version 4.6-14). In the next section, we provide test results and show that the stacked ensemble technique is superior in predicting capreomycin resistance in *Mycobacterium* spp. rather than applying our machine learning algorithms individually.

Results

For classification, we divided the dataset into training and test sets. We considered stratified sampling where 80% of the dataset is used as the training set and the remaining 20% is used as

the test set. First we trained our base classifiers—the generalized linear model (GLM), C5.0 algorithm, and SVM—using the training set and then fed the predictions obtained from this step to the meta-classifier. Next we applied this stacked model to predict the resistance and susceptibility of our test dataset. We tuned the C5.0 and SVM models based on our training dataset to find the best parameters and selected *trials* = 1, *model* = rules, *winnow* = true for the C5.0 model, and the radial basis kernel function in the SVM with C and σ parameter values of 1 and 0.002, respectively. We considered 10 repeats of 10-fold cross validation to train the model using 100 resamples for the training dataset.

Figure 3 depicts the accuracy statistics for all simulation runs for the three base classifiers. The box-and-whisker plots in Figure 3 show the accuracy statistics for the training dataset where each box represents the median (indicated by a solid vertical line) and lower and upper quartiles, and each whisker gives the lowest and highest observations. Note that a small circle beyond the left whisker in Figure 3 indicates an outlier that is far from the lowest value. By calculating the average training accuracy values, we found that the SVM achieved the most accurate result (78.95%), much more accurate than those of the GLM and C5.0, 50.25% and 54.20%, respectively. Based on the training accuracy, the SVM is identified as a strong base classifier for our dataset. As stated earlier, we can combine the predictions of all three models to increase the classification accuracy if pairwise predictions of the classifiers have low correlation. The histograms in Figure 4 represent the accuracies of all simulation runs for each classifier, and the scatter plots show pairwise comparisons of the accuracies obtained from the three classifiers. The fitted line in each scatter plot gives the positive or negative correlation of all accuracies (indicated by small circles) between any two classifiers.

283 It is clear from the scatter plots in Figure 4 that the predicted outputs of the base classifiers
284 have low correlation. The correlation between the GLM and C5.0 results is 0.094 (positive
285 correlation). The GLM and SVM results have a correlation of 0.046 (negative correlation), and the
286 C5.0 and SVM results have a correlation value of 0.024 (negative correlation). As all the
287 correlations are much less than 0.75, combining the predictions using a meta-classifier is likely to
288 increase the classification accuracy.

289 The training and test accuracies obtained using the stacked ensemble model with the three
290 different meta-classifiers are listed in Table 2. We tuned the RF meta-classifier and selected an
291 *mtry* parameter value of 2.

292 The meta-classifiers provide significant improvement in training accuracy. In particular, when
293 we combine the predictions of the base classifiers using the GLM meta-classifier, the training
294 accuracy is improved to 81.38%, an increase in training accuracy of approximately 2.43% from
295 the best base model alone. We applied our stacked ensemble model to make predictions on our
296 test dataset and achieved test accuracies up to 81.25%. The confusion matrices for the test
297 dataset for the meta-classifiers are given in supplementary tables (Table S3-Table S5).

298 We also compared our stacked ensemble approach with the performance of BLASTp using
299 default parameter settings (https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins). The results
300 depicted in Figure 5 are the number of matched AR protein sequences as a function of percent
301 identity threshold in test data using the AR samples from the training data. For example, if we
302 consider percent identity ≥50% for a test protein sequence to be matched (true positive) with a
303 training AR sequence, we obtain five of nine true positives. Importantly, in order to achieve the
304 same number of true positives as for our GLM meta-classifer, the percent identity threshold of

BLASTp must be 33%, but the "cost" comes in the form of increasing false positives. For example, when we set the percent identity threshold to 33%, all non-AR test sequences are incorrectly identified as AR sequences, resulting in a high false positive rate. Thus, there is a big tradeoff with the threshold setting when using a best-hit method such as BLASTp.

Discussion

Prediction of capreomycin resistance in *Mycobacterium* spp. is a challenging but important problem. In this paper we introduced the use of machine learning with protein sequences which allows a rich feature set to be used to predict capreomycin resistance in *Mycobacterium* spp. We based our classifier on a stacked ensemble machine learning technique which uses two layers of classifiers to achieve higher accuracy. We gathered 44 unique resistance and 36 unique susceptibility protein sequences and filtered 621 original features, removing redundant and less informative features, to obtain a set of 336 features. We considered amino acid composition, physicochemical properties, evolutionary-relevant properties, and the secondary structure of proteins to generate features.

To validate the effectiveness of our approach, we built a stacked ensemble classifier using a generalized linear model (GLM), a C5.0 model, and an SVM as base classifiers and GLM, linear discriminant analysis, and a random forest as individual meta-classifiers. Our repeated ten-fold cross validation results showed that stacking can increase the training accuracy up to 2.43% compared to the best base classifier and can achieve test accuracy up to 81.25%. The identity threshold must be quite low to obtain similar classification results with BLASTp, and this leads to a marked increase in false positives, i.e., sequences identified as AR that are not. Although we considered only capreomycin resistance, our approach could potentially be applied to identify

other antibiotic resistance such as amikacin and kanamycin resistance in *Mycobacterium* spp. as well as antibiotic resistance in other bacteria. The stacked ensemble model with the GLM meta-classifier gives superior performance for our training and test datasets. While the accuracy of our classifier is reasonable, i.e., greater than 80%, given the availability of more sequences to use for training, the accuracy would be higher. In the future, when a greater number of AR sequences are available, we will retrain our algorithm.

Conflict of Interest

We declare that no conflict of interest exists.

Acknowledgements

We wish to thank the reviewers for their careful reading of the manuscript and for their thoughtful comments and suggestions. Their contributions have resulted in a stronger publication.

References

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J., 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res, 25(17), pp.3389-3402.

Aminov, R.I., 2009. The role of antibiotics and antibiotic resistance in nature. Environ Microbiol, 11(12), pp.2970-2988.

Balakrishnama, S., and Ganapathiraju, A., 1998. Linear discriminant analysis-a brief tutorial. Institute for Signal and information Processing, 18, 1-8.

Breiman, L., 1996. Stacked regressions. Machine learning, 24(1), pp.49-64.

Breiman, L. 2001. Random forests. Machine learning, 45(1), 5-32.

Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X. and Chen, Y.Z., 2003. SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res, 31(13), pp.3692-3697.

Chen, M.L., Doddi, A., Royer, J., Freschi, L., Schito, M., Ezewudo, M., Kohane, I.S., Beam, A. and Farhat, M., 2018. Deep learning predicts tuberculosis drug resistance status from genome

sequencing data. bioRxiv, p.275628.

Chowdhury, A.S., Alam, M.M. and Zhang, Y., 2015, August. A biomarker ensemble ranking framework for prioritizing depression candidate genes. In 2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) (pp. 1-6). IEEE.

Davis, J.J., Boisvert, S., Brettin, T., Kenyon, R.W., Mao, C., Olson, R., Overbeek, R., Santerre, J., Shukla, M., Wattam, A.R. and Will, R., 2016. Antimicrobial resistance prediction in PATRIC and RAST. Sci Rep, 6, p.27930.

Dijkstra, J.A., van der Laan, T., Akkerman, O.W., Bolhuis, M.S., de Lange, W.C.M., Kosterink, J.G., van der Werf, T.S., Alffenaar, J.W.C. and van Soolingen, D., 2018. In vitro susceptibility of Mycobacterium tuberculosis to amikacin, kanamycin, and capreomycin. Antimicrob Agents Chemother, 62(3), pp.e01724-17.

Ding, C.H. and Dubchak, I., 2001. Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics, 17(4), pp.349-358.

Dubchak, I., Muchnik, I., Holbrook, S.R. and Kim, S.H., 1995. Prediction of protein folding class using global description of amino acid sequence. Proceedings of the National Academy of Sciences, 92(19), pp.8700-8704.

Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I. and Kim, S.H., 1999. Recognition of a protein fold in the context of the SCOP classification. Proteins: Structure, Function, and Bioinformatics, 35(4), pp.401-407.

Eslami Manoochehri, H., Kadiyala, S. S., Birjandtalab, J., and Nourani, M., 2018. Feature Selection to Predict Compound's Effect on Aging. Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (pp. 419-427). ACM.

Forsberg, K.J., Patel, S., Gibson, M.K., Lauber, C.L., Knight, R., Fierer, N. and Dantas, G., 2014. Bacterial phylogeny structures soil resistomes across habitats. Nature, 509(7502), p.612.

Fu, L., Niu, B., Zhu, Z., Wu, S. and Li, W., 2012. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics, 28(23), pp.3150-3152.

Golmah, V., 2014. An efficient hybrid intrusion detection system based on C5. 0 and SVM. International Journal of Database Theory and Application, 7(2), 59-70.

Gygli, S.M., Borrell, S., Trauner, A. and Gagneux, S., 2017. Antimicrobial resistance in Mycobacterium tuberculosis: mechanistic and evolutionary perspectives. FEMS Microbiol Rev, 41(3), pp.354-373.

Her, H.L. and Wu, Y.W., 2018. A pan-genome-based machine learning approach for predicting antimicrobial resistance activities of the Escherichia coli strains. Bioinformatics, 34(13), pp.i89-

i95.

Jones, D.T., 1999. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol, 292(2), pp.195-202.

Kavvas, E.S., Catoiu, E., Mih, N., Yurkovich, J.T., Seif, Y., Dillon, N., Heckmann, D., Anand, A., Yang, L., Nizet, V. and Monk, J.M., 2018. Machine learning and structural analysis of Mycobacterium tuberculosis pan-genome identifies genetic signatures of antibiotic resistance. Nature communications, 9(1), p.4306.

Kleinheinz, K.A., Joensen, K.G. and Larsen, M.V., 2014. Applying the ResFinder and VirulenceFinder web-services for easy identification of acquired antibiotic resistance and E. coli virulence genes in bacteriophage and prophage nucleotide sequences. Bacteriophage, 4(2), p.e27943.

Kurgan, L., Cios, K. and Chen, K., 2008. SCPRED: accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. BMC bioinformatics, 9(1), p.226.

Kurgan, L.A. and Homaeian, L., 2006. Prediction of structural classes for protein sequences and domains—impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy. Pattern Recognition, 39(12), pp.2323-2343.

Li, W. and Godzik, A., 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics, 22(13), pp.1658-1659.

Li, Y.H., Xu, J.Y., Tao, L., Li, X.F., Li, S., Zeng, X., Chen, S.Y., Zhang, P., Qin, C., Zhang, C. and Chen, Z., 2016. SVM-Prot 2016: a web-server for machine learning prediction of protein functional families from sequence irrespective of similarity. PloS one, 11(8), p.e0155290.

Liu, B. and Pop, M., 2008. ARDB—antibiotic resistance genes database. Nucleic Acids Res, 37(suppl_1), pp.D443-D447.

Liu, B., Zhang, D., Xu, R., Xu, J., Wang, X., Chen, Q., Dong, Q. and Chou, K.C., 2013. Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. Bioinformatics, 30(4), pp.472-479.

Liu, T. and Jia, C., 2010. A high-accuracy protein structural class prediction algorithm using predicted secondary structural information. J Theor Biol, 267(3), pp.272-275.

Madsen, H., and Thyregod, P., 2010. Introduction to general and generalized linear models. CRC Press.

McArthur, A.G. and Tsang, K.K., 2017. Antimicrobial resistance surveillance in the genomic age. Annals of the New York Academy of Sciences, 1388(1), pp.78-91.

Mnyambwa, N.P., Kim, D.J., Ngadaya, E., Chun, J., Ha, S.M., Petrucka, P., Addo, K.K., Kazwala, R.R. and Mfinanga, S.G., 2018. Genome sequence of Mycobacterium yongonense RT 955-2015 isolate from a patient misdiagnosed with multidrug-resistant tuberculosis: First clinical detection in Tanzania. Int J Infect Dis, 71, pp.82-88.

Moradigaravand, D., Palm, M., Farewell, A., Mustonen, V., Warringer, J. and Parts, L., 2018. Prediction of antibiotic resistance in Escherichia coli from large-scale pan-genome data. PLoS computational biology, 14(12), p.e1006258.

Niehaus, K.E., Walker, T.M., Crook, D.W., Peto, T.E. and Clifton, D.A., 2014, June. Machine learning for the prediction of antibacterial susceptibility in Mycobacterium tuberculosis. In IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI) (pp. 618-621). IEEE.

Santerre, J.W., Davis, J.J., Xia, F. and Stevens, R., 2016. Machine learning for antimicrobial resistance. arXiv preprint arXiv:1607.01224.

Scholkopf, B., and Smola, A. J., 2001. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Van der Laan, M.J., Polley, E.C. and Hubbard, A.E., 2007. Super learner. Stat Appl Genet Mol Biol, 6(1).

Wattam, A.R., Abraham, D., Dalay, O., Disz, T.L., Driscoll, T., Gabbard, J.L., Gillespie, J.J., Gough, R., Hix, D., Kenyon, R. and Machi, D., 2013. PATRIC, the bacterial bioinformatics database and analysis resource. Nucleic Acids Res, 42(D1), pp.D581-D591.

Wattam, A.R., Davis, J.J., Assaf, R., Boisvert, S., Brettin, T., Bun, C., Conrad, N., Dietrich, E.M., Disz, T., Gabbard, J.L. and Gerdes, S., 2016. Improvements to PATRIC, the all-bacterial bioinformatics database and analysis resource center. Nucleic Acids Res, 45(D1), pp.D535-D542.

Wei, L., Liao, M., Gao, X. and Zou, Q., 2015. Enhanced protein fold prediction method through a novel feature extraction technique. IEEE Trans Nanobioscience, 14(6), pp.649-659.

Wolpert, D.H., 1992. Stacked generalization. Neural Netw, 5(2), pp.241-259.

Xavier, B.B., Das, A.J., Cochrane, G., De Ganck, S., Kumar-Singh, S., Aarestrup, F.M., Goossens, H. and Malhotra-Kumar, S., 2016. Consolidating and exploring antibiotic resistance gene data resources. Journal of clinical microbiology, 54(4), pp.851-859.

Yang, Y., Jiang, X., Chai, B., Ma, L., Li, B., Zhang, A., Cole, J.R., Tiedje, J.M. and Zhang, T., 2016. ARGs-OAP: online analysis pipeline for antibiotic resistance genes detection from metagenomic data using an integrated structured ARG-database. Bioinformatics, 32(15), pp.2346-2351.

Zhang, S., Ding, S. and Wang, T., 2011. High-accuracy prediction of protein structural class for low-similarity sequences based on predicted secondary structure. Biochimie, 93(4), pp.710-714.

List of figure legends

Figure 1. Illustrating changes of *p*-values with selected features, (a) *p*-values vs. features and (b) -$\log_{10}$(*p*-value) vs. features.

Figure 2. Stacked ensemble model of generalized linear model (GLM), C5.0, and SVM algorithms using a meta-classifier. The meta-classifier can be a GLM, linear discriminant analysis (LDA), or a random forest classifier (RF).

Figure 3. Whisker plots illustrating the training accuracy of the three base classifiers.

Figure 4. Correlations between predictions of base classifiers. Low correlations indicate that an ensemble classifier can be used.

Figure 5. Identification of AR sequences in test data using BLASTp as a function of percent identity for AR samples from training data. In order to identify all AR sequences, an identity threshold of 33% must be used, and this leads to a high rate of false positives.

List of supplementary information

Table S1: List of 44 Resistance Sequences

Table S2: List of 36 Susceptibility Sequences

Table S3: Confusion matrix for GLM meta-classifier

Table S4: Confusion matrix for LDA meta-classifier

Table S5: Confusion matrix for RF meta-classifier

Tables

Table 3. List of 621 Protein Features.

Feature	Feature dimension
Amino acid composition	20D
Hydropathy	21D
Normalized van der Waals volume	21D
Polarity	21D
Polarizability	21D
Charge	21D
Secondary structure	21D
Solvent accessibility	21D
Surface tension	21D
PSSM-based bigrams	400D
Structure-sequence	6D
Structure-probability	27D

Table 4. Performance Comparison of Stacked Ensemble Models.

Meta Classifier	Training Accuracy	Test Accuracy
GLM	81.38%	81.25%
LDA	80.30%	75.0%
RF	80.94%	81.25%

Figures

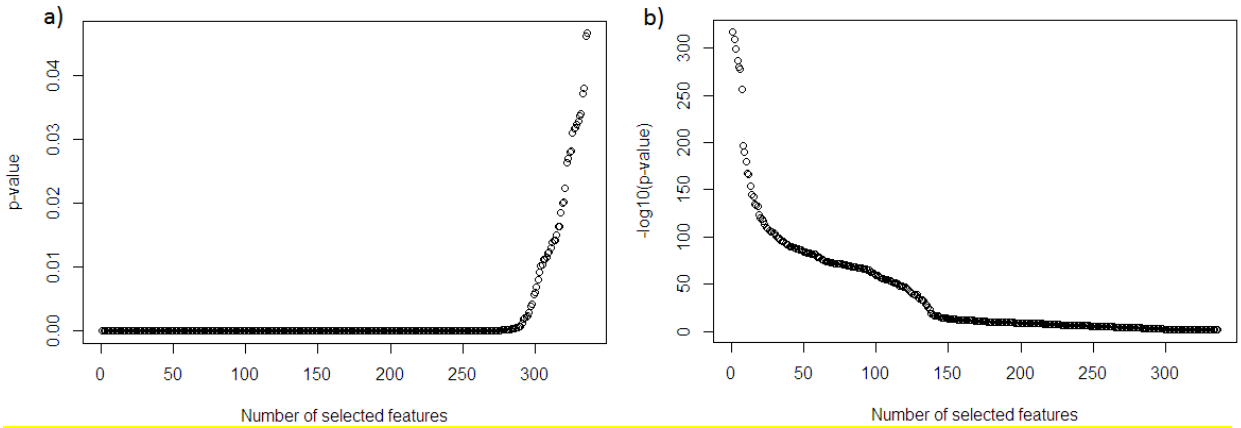

Figure 1. Illustrating changes of p-values with selected features: (a) p-values vs. features and (b) $-\log_{10}$(p-value) vs. features

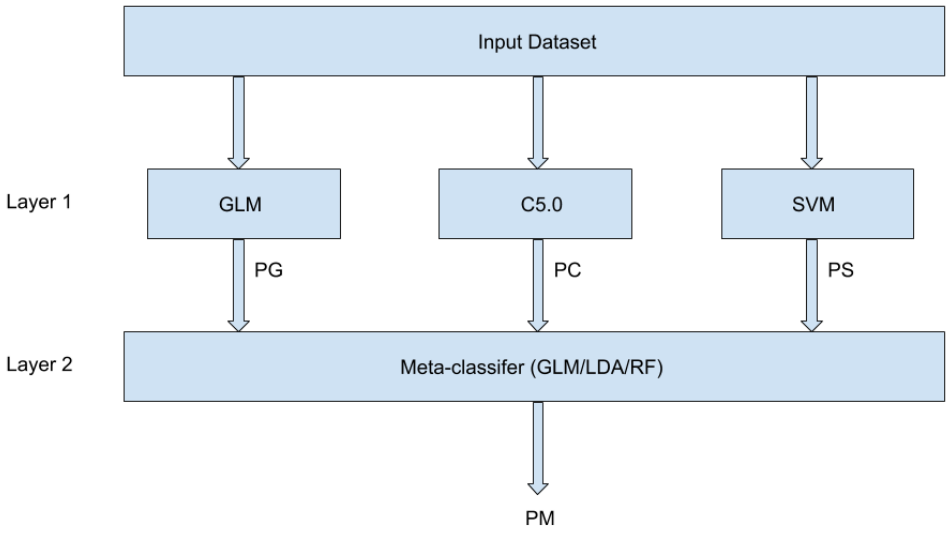

Figure 2. Stacked ensemble model of generalized linear model (GLM), C5.0, and SVM algorithms using a meta-classifier. The meta-classifier can be a GLM, linear discriminant analysis (LDA), or a random forest classifier (RF).

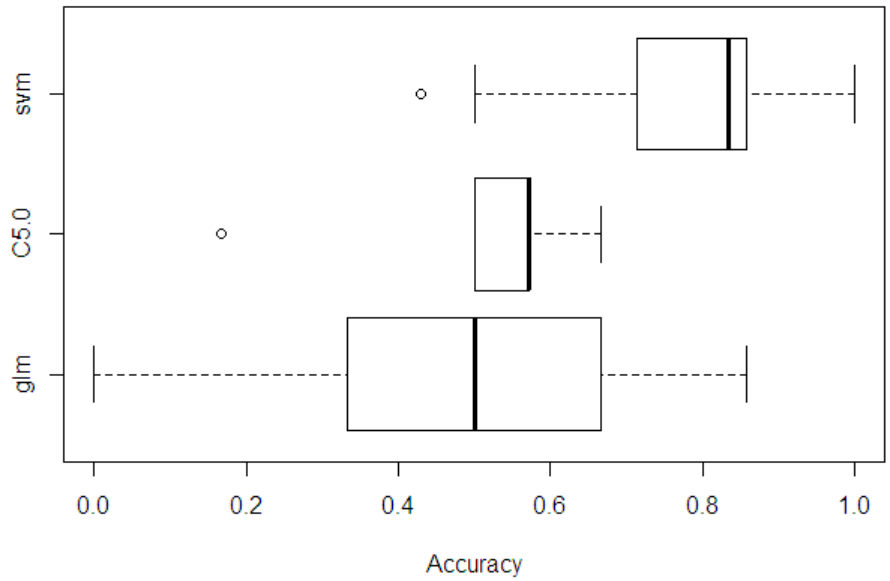

Figure 3. Whisker plots illustrating the training accuracy of the three base classifiers.

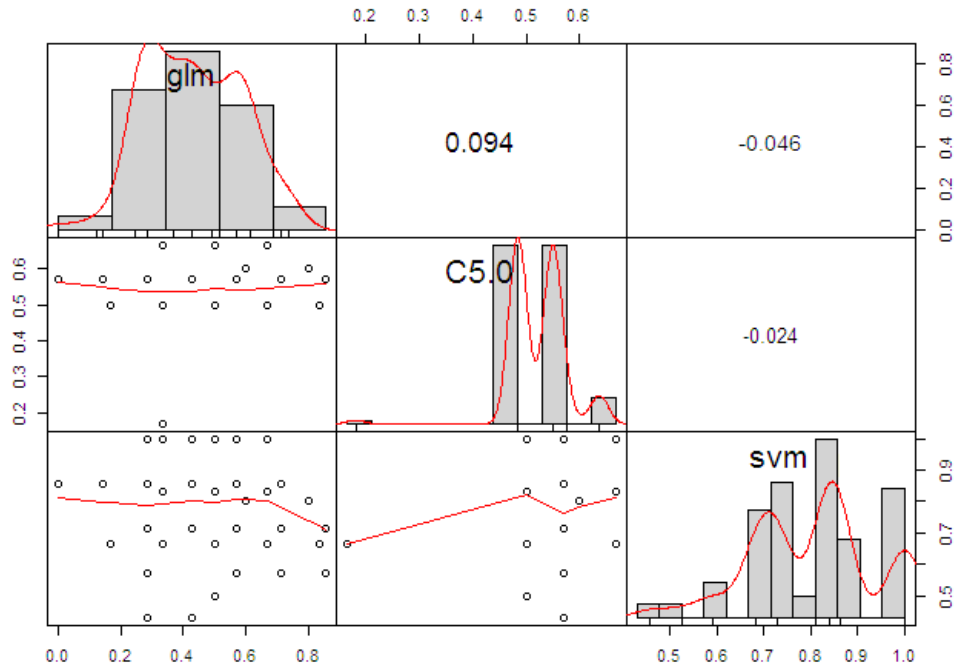

Figure 4. Correlations between predictions of base classifiers. Low correlations indicate that an ensemble classifier can be used.

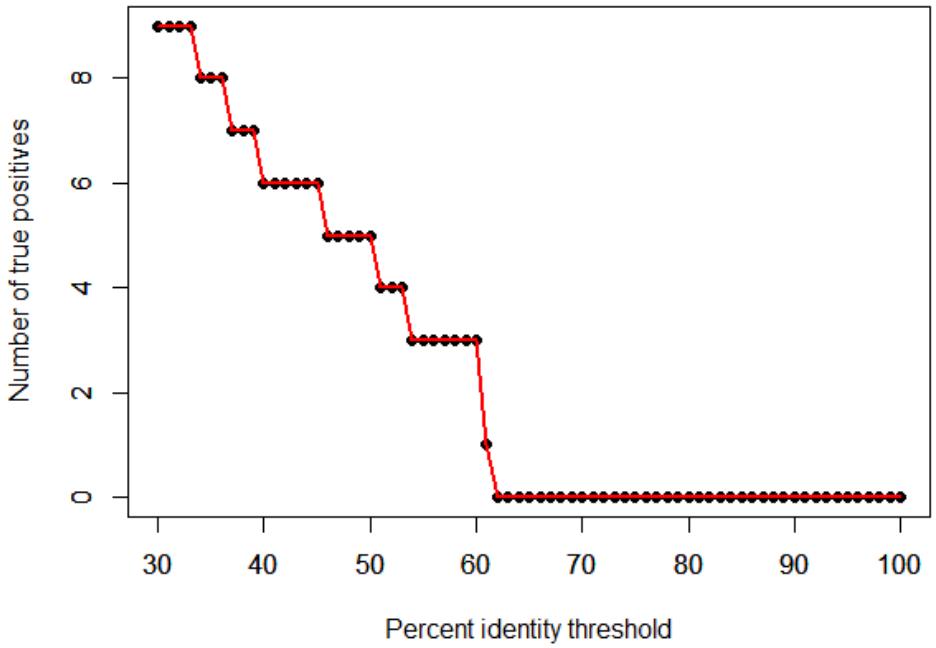

Figure 5. Identification of AR sequences in test data using BLASTp as a function of percent identity for AR samples from training data. In order to identify all AR sequences, an identity threshold of 33% must be used, and this leads to a high rate of false positives.