CSC687- R

COVID-19/SARS B-cell Epitope Prediction

Riya Agrawal

James Alfano

Advisor: Prof. Aguiar-Pulido

March 29, 2022

Table of Contents

1.0 Introduction and Background

The purpose of this project is to predict, using binary classification, whether an amino acid peptide exhibits antibody-inducing activities. The amino acid peptides studied are from the COVID-19 antigen. An antigen is any substance that causes the host's immune system to produce antibodies to fight off an unrecognized substance. The region of the antigen that is recognized by the host's immune system and upon which an antibody binds is called the epitope. Antibody valence is the relative capability of an antibody to bind with an antigen. If epitope regions of an antigen can be predicted and then mapped, then this information can be used to design and develop vaccines that will prompt the immune system to produce antibodies to a specific antigen. Here, we will be looking at COVID-19 amino acid peptides datasets to predict whether it is positive or negative for antibody-inducing activities. The target value is antibody valence. Antibody valence was originally measured as "Positive-High," "Positive-Intermediate," "Positive-Low," "Positive," or "Negative." However, due to an imbalance of data in the target value, "Positive-High," "Positive-Intermediate," "Positive-Low," and "Positive" were combined into one "Positive" label. In doing so, the target value is transformed to either "Positive" or "Negative."

Epitopes can play a key role in the design of vaccines, and the prevention, diagnosis, and treatment of diseases. Rapid and accurate identification of epitope sites can further achieve those objectives. Employing computational techniques may improve the accuracy of epitope identification and may reduce the time and potentially the cost of identifying epitopes. Within machine learning, researchers have approached this problem from multiple angles. Traditional machine learning techniques have been implemented with success. In particular, support vector machines (SVMs) have shown promise in B-cell prediction. Deep learning techniques have also found success with researchers building state-of-the-art classifiers specific to epitope prediction.

New algorithms such as DeepNetBim for human leukocyte antigen (HLA)-peptide binding prediction and epitope3D for conformational epitopes prediction have shown promise.

Kaggle, the source of our data, allows the users to publish their work on a given dataset. The user-based submissions, for the dataset in question, employ a wide range of techniques, including different preprocessing techniques, and implement a variety of models. Preprocessing is a broad set of techniques used to transform the data into a format that is more conducive to analysis and general data science techniques. Some users applied little to no preprocessing on the data. While others applied techniques such as randomizing the data and upsampling on the target value. User submissions also showed a great variety of models chosen. Many different traditional machine learning techniques were applied. Most common were logistic regression, k-nearest neighbor, and tree-based models. Many users also implemented deep learning techniques. The neural networks applied, had different architecture.

The user reported success also varied. Generally, submissions ranged from 85% to 95% accuracy. However, most users reported approximately 90% accuracy for their best model. Generally, the best results were obtained from tree-based models as well as deep learning techniques.

COVID-19 has shown the need for fast, effective, and efficient vaccine development. A computationally facilitated epitope-based vaccine is part of that charge forward. This project seeks to predict whether COVID-19 amino acid peptides datasets are positive or negative for antibody inducing activities; activities occurring at epitopes. Moreover, while the scope of this project focuses on COVID-19, it is believed that the knowledge gleaned from the results may have broader application to other antigens.

2.0 Methods

2.1 Dataset

The dataset for this project has been taken from Kaggle:

https://www.kaggle.com/futurecorporation/epitope-prediction

The dataset contains three files with no missing values:

- input_bcell.csv: this is the main training data. The number of rows is 14,387 for all combinations of 14,362 peptides and 757 proteins.
- input_sars.csv: this is also the main training data. The number of rows is 520.
- input_covid.csv: this is our target data. There is no label data in columns.

There are 13 variables that we will be considering for classifying the target and the explanation for each is given below:

- parent_protein_id : parent protein ID
- protein_seq: parent protein sequence
- start_position: start position of the peptide
- end_position: end position of the peptide
- peptide_seq : peptide sequence
- chou_fasman: peptide feature, β turn
- emini: peptide feature, relative surface accessibility
- kolaskar_tongaonkar: peptide feature, antigenicity
- parker: peptide feature, hydrophobicity
- isoelectric_point: protein feature
- aromacity: protein feature
- hydrophobicity: protein feature

- stability: protein feature and bcell and sars dataset have antibody valence (target value)
- target : antibody valence (target value)

We combined the input_bcell and input_sars file for training and validation, and the input_covid file for future prediction (the target value is unknown for the input_covid dataset). The variables which give identity information are not considered for prediction. These variables are:

- parent_protein_id (parent protein ID): identifier
- protein_seq (parent protein sequence): sequence name and is unique
- start_position (start position of peptide): the unique identifier of start position
- end_position (end position of peptide): the unique identifier of the end position
- peptide_seq (peptide sequence): sequence name and is unique in nature

2.2 Experimental Data Analysis

The objective of experimental data analysis (EDA) is to better understand the data by performing statistical analysis and creating visual plots for the data. The first step was to check for missing values and to address it, if present. In this case, there were no missing values. The next step was to look at a summary/plot of the variables. This step provided multiple insights: (1) there are numeric and character values present in the data; (2) the predictors emini and stability most likely contain outliers as the max value is significantly larger than the 3rd quartile value; (3) the target value, antibody valence, is heavily skewed with 10,865 negative values and 4,042 positive values. To improve our ability to model the data, the outliers were removed and the target value was balanced with upsampling. The techniques used to accomplish these tasks are discussed below in section 2.3.

It was also identified, during the EDA, that the COVID-19 data set does not contain the target value. Thus, we will not be able to report evaluation metrics for this data set. To combat this, a small holdout set was created from the combined B-cell and SARS dataset to be able to report metrics. The holdout set uses approximately 20% of the available B-cell and SARS data. Specifically, 11,712 samples are used for training and 3,195 samples are used for testing.

2.3 Techniques Applied

Before the models could be applied, the data had to undergo preprocessing. As previously stated, the data contained outliers and an imbalanced target variable. Preprocessing was performed to address these issues. The outliers were removed by identifying the interquartile range (IQR) and removing the values which do not lie within 3.0*IQR of the first and third quartile. To combat the imbalanced target value, upsampling was applied using the "groupdata2" package. Upsampling is randomly sampling the data (with replacement) such that two classes are of equal size (the size of the minority class is made equal to the size of the majority class). These changes were allowed us to make better predictions on our data.

Four models were chosen for the classification: logistic regression; k-nearest neighbor (KNN); random forest; and gradient boosting machine (GBM).

Logistic regression serves as a baseline model. The goal of logistic regression is to predict the probability that the response variable belongs to a class. To estimate the coefficients, the model utilizes the maximum likelihood estimation (MLE). The output is then transformed using a sigmoid function, thereby resulting in the probability that a value belongs to a class.

KNN seeks to classify a point by assigning it to its nearest class (neighbor). For this project, two KNN classification models were built with different values of k: sqrt(n) and 5 (where n is the number of samples). The value of k determines the number of neighbors included in the model.

There are no statistical methods that guarantee the most favorable value of k. Generally, choosing a small value of k leads to unstable decision boundaries. A common starting value used is k = sqrt(n). Thus, we built our first model with this value. The value 5 was chosen because cross-validation results gave k = 5 with the highest accuracy.

The random forest model was applied with the optimal value of mtry. Mtry is the number of predictors sampled for each tree in the random forest. Random forest is a bagging model that works by building multiple decision trees with different predictors and data subsets for each tree. The output combines the results of multiple decision trees to obtain a single result. The model then takes the average in the case of regression and the majority vote in the case of classification. The value of mtry was determined by using a 10-fold cross-validation (CV). CV is discussed below in section 2.4.

The GBM model was also applied using a 10-fold CV to calculate an estimate of generalization error. GBM is a boosting algorithm that works by building decision trees sequentially such that the new tree built learns from the previous tree.

2.4 Evaluation

To gauge the performance of our models, accuracy, F-score, and their corresponding confusion matrices have been calculated. Accuracy is the measure of correctly identified predictions. F-score combines the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. The confusion matrix is an m by m matrix used to evaluate the performance of a classification model, where m is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.

For the logistic regression model, we employed the holdout set technique and 10-fold CV for our evaluation method. The holdout set technique works by dividing the dataset into two parts:

the training set and the testing set. The testing set, also known as the holdout set, is the unseen data that is used for the evaluation of the model. For our project, 80% of the data was used for training and 20% of the data was used for testing. 10-fold CV works by splitting the data into 10 equal parts and "holding out" one part for evaluation in each iteration. These evaluation sets are disjoint and the final error of the model is the average errors from the 10 iterations.

For the KNN models, the holdout set technique, with the same split, and 10-fold CV was used for evaluation. The optimal value of k (k = 5) was determined with 10-fold CV.

The random forest model used a combination of 10-fold CV and the holdout set technique for evaluation. 10-fold CV was used on training data to obtain the best value of mtry. The model, with the optimal value for mtry, was then tested with the holdout set.

The GBM model also used 10-fold CV and the holdout set technique. 10-fold CV was used on training data to calculate an estimate of generalization error. The optimal model was then tested with the holdout set.

3.0 Results

Our initial results, before removing the outliers and upsampling the target variable (outlined in the March 29th draft), were not optimal. While testing accuracies and F-score were relatively high, the confusion matrices showed that the models were heavily skewed; the majority of cases were predicted as negative (0). The issues of outliers and an imbalanced target variable were most likely causing the poor results. To correct this, outliers were removed and upsampling was implemented.

The new confusion matrices for each model show a more even distribution of predictions, *i.e.*, less skewed towards negative (0) predictions. Generally, the accuracies and F-scores also

improved. The random forest model, before addressing these issues, had a testing accuracy of .84 and an F-score of .89. After implementing the changes, the model had a testing accuracy of .92 and an F-score of .92. Out of all the models tested, the random forest gave us our best results.

The confusion matrices, accuracy assessments, and F-scores are set forth below. Accuracy is reported before cross-validation and after cross-validation.

Performance: Confusion Matrix

<table>
<tr><th>Model</th><th>Confusion Matrix</th></tr>
<tr><td>Logistic Regression</td><td>
<table>
<tr><td></td><td colspan="2">Actual</td></tr>
<tr><td>Pred</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1286</td><td>886</td></tr>
<tr><td>1</td><td>946</td><td>1346</td></tr>
</table>
</td></tr>
<tr><td>KNN (with optimal k, i.e, k = 5, from CV)</td><td>
<table>
<tr><td></td><td colspan="2">Actual</td></tr>
<tr><td>Pred</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1675</td><td>367</td></tr>
<tr><td>1</td><td>557</td><td>1865</td></tr>
</table>
</td></tr>
<tr><td>Random Forest</td><td>
<table>
<tr><td></td><td colspan="2">Actual</td></tr>
<tr><td>Pred</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1982</td><td>105</td></tr>
<tr><td>1</td><td>250</td><td>2127</td></tr>
</table>
</td></tr>
</table>

GBM	

	Actual	
Pred	0	1
0	1889	339
1	343	1893

Performance: Accuracy and F-score

Model	Accuracy		F-score
	Before Cross-Validation	After Cross-Validation	
Logistic Regression	0.59	0.77	0.58
KNN	0.70 (k=sqrt(n))	0.79 (k=5)	0.78
Random Forest	**0.91 (mtry=8)**	**0.92 (mtry=4)**	**0.92**
GBM	0.80	0.84	0.84

As shown in the tables above, random forest and GBM perform relatively well. The random forest model, however, outperforms the GBM model. Comparing the application and non-application of CV to the random forest model, the difference in accuracy is minimal. For all other models, applying CV resulted in a greater improvement in accuracy.

The logistic regression had the lowest F-score and one of the lowest accuracy rates for all the models tested. This model did not perform well on the data set. KNN, with k = sqrt(n) and k = 5, also did not prove to have optimal results.

4.0 Conclusion

The goal of this project was to predict, using binary classification, whether an amino acid peptide exhibits antibody inducing activities, *i.e*,. positive (1) or negative (0) for such activities. This goal was accomplished. After preprocessing our data and tuning our models, we obtained a top test accuracy of .92 by using a random forest.

This project challenged and furthered our machine learning skills and introduced us to the concepts of epitopes. By completing this project, we learned how to (1) implement, tune, and test machine learning algorithms, (2) apply and gain experience with multiple R packages, and (3) manipulate and clean large data sets. The largest non-machine learning obstacle of this project was stepping outside of our comfort zones into an unfamiliar domain, biochemistry. This obstacle was tackled through research and by a desire to learn. In terms of machine learning techniques, when issues arose, our team worked together to identify the problem, research the topic, and find the optimal solution.

In sum, the knowledge gained from this project improved our understanding and appreciation for machine learning. It made us appreciate the potential that machine learning has to solve a myriad of problems in numerous domains.

4.1 Future Work

While the scope of this project focuses on traditional machine learning techniques, a logical next step would be to apply deep learning algorithms to the problem. A comparison could then be

made between the methods implemented here and the results obtained from applying a neural network. Given the complexity and size of the epitope data, a neural network has the potential to provide a higher accuracy rate.

Another logical next step would be to apply the pre-trained models to other antigens. By assessing the models on other unseen antigens, one could get a better idea of how the models would perform if implemented in the real world. If the models prove to make successful predictions, then this may benefit researchers studying novel antigens. An accurate model might reduce the cost and time it takes to gain valuable information on new antigens.

5.0 References

COVID-19/SARS B-cell epitope prediction. (n.d.). Kaggle. https://www.kaggle.com/datasets/futurecorporation/epitope-prediction

Da Silva, Bruna Moreira, et al. (2021). Epitope3D: A machine learning method for conformational B-cell epitope prediction. *Briefings in Bioinformatics*, *23(1)*. https://doi.org/10.1093/bib/bbab423

Wang, Hsin-Wei, and Tun-Wen Pai. (2014). Machine learning-based methods for prediction of linear B-cell epitopes. *Methods in Molecular Biology*, *1184,* 217–236. https://doi.org/10.1007/978-1-4939-1115-8_12

Yang, Xiaoyun, et al. (2021). DeepNetBim: Deep learning model for predicting HLA-epitope interactions based on network analysis by harnessing binding and immunogenicity information. *BMC Bioinformatics*, *22*. https://doi.org/10.1186/s12859-021-04155-y