2023

MACHINE LEARNING METHODS FOR PREDICTION OF HUMAN INFECTIOUS VIRUS AND IMPUTATION OF HLA ALLELES

Xiaoqing Gao
Michigan Technological University, xgao5@mtu.edu

MACHINE LEARNING METHODS FOR PREDICTION OF HUMAN INFECTIOUS VIRUS AND IMPUTATION OF HLA ALLELES

By

Xiaoqing Gao

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Statistics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2023

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Statistics.

Department of Mathematical Sciences

Dissertation Advisor:	*Dr. Kui Zhang*
Committee Member:	*Dr. Qiuying Sha*
Committee Member:	*Dr. Hairong Wei*
Committee Member:	*Dr. Xiao Zhang*
Department Chair:	*Dr. Jiguang Sun*

Contents

List of Figures

List of Tables

Author Contribution Statement

This dissertation is submitted for the degree of Doctor of Philosophy at Michigan Technological University. The research achievements described in this dissertation were conducted under the supervision of Professor Kui Zhang in the Department of Mathematical Sciences at Michigan Technological University from September 2019 to June 2023.

The study presented in Chapter 1 is a collaborative work with Professor Hairong Wei. Professor Kui Zhang conceived and supervised the entire research. Professor Hairong Wei confirmed the design of this project. Xiaoqing Gao formulated the procedures, implemented the program with R, performed the evaluations, and drafted the manuscript.

The study presented in Chapter 2 is conceived and designed by Xiaoqing Gao. Professor Kui Zhang collected the datasets and supervised the research. Xiaoqing Gao implemented the program with Python, evaluated and compared the performance of the proposed method to other popular methods, and drafted the manuscript.

The study presented in Chapter 3 is devised and supervised by Professor Kui Zhang. Xiaoqing Gao developed the method, implemented the program with R and Python, conducted the simulations, and drafted the manuscript.

Those three studies are under revising. Hopefully they would be published in the near future.

Acknowledgements

It took 30 years for this dissertation to come to you, from I was a kid to I have 2 kids, from my “A, B, C” s in elementary school to complicated proof in graduate school, from my beautiful hometown, Tianjin in China to another splendid town, Houghton in USA. Along the way, what can achieve me is not flowers and applause but difficulties and sorrows. Four years doctoral career is too long, so long that every paper I read, and every line of programming would make me toss and turn. Four years doctoral career is too short, so short that a few pictures and sentences could entirely present the whole study and life. As I retrospect my four years from 2019 to 2023, only sincere and deep gratitude is buried in my heart.

First of all, I would like to thank my advisor Dr. Kui Zhang for his guidance in my master and Ph.D program. I had taken three courses that were taught by him. His solid theoretical foundation and erudite statistical knowledge always influenced me greatly when I leant to do research on myself. I had taken his three courses, and it was enjoyable sitting in his class. I also appreciate his timely feedback when I meet some research questions. Without his guidance, I couldn’t have finished this four-year journey with abundant experience.

Next, I would express my appreciation to my dear committee members, Professor Kui Zhang, Professor Qiuying Sha, Professor Hairong Wei and Professor Xiao Zhang. They have witnessed my growth in Statistics since they are my committee members for both a master and Ph.D degree. I appreciate their valuable time for proofreading my dissertation and bringing treasured suggestions. It is my great pleasure to invite four amazing professors as the last guardians in my doctoral career.

Thirdly, I would like to thank all the instructors in the Math department. I learnt plenty of statistical knowledge from these splendid lectures by the excellent teachers. I was impressed by the solid foundation and humorous style of the instructors.

Fourthly, I would like to declare my appreciation to all staff and graduate students in the Department of Mathematical Sciences at Michigan Technological University. Thanks to the timely and efficient service from our staff, all graduate students could enjoy our studies and lives in our university. Moreover, I would like to thank all graduate students whose passions and knowledge positively drive my program to be pleasant and fruitful.

Last but not least, I would express my sincere appreciation to my supportive family, my dear husband Yang Yang and two cute sons, Aaron and Adam. Continuing the Ph.D program as well as taking care of two little ones is unbelievable, however, thanks to my husband that made this come true. I deeply appreciate him for doing his best to do housework and take care of the children so that I can obtain spare time to study and work on my project. For my two young sons, although they couldn't help me with anything, they gave me endless motivation and courage to finish the program. Also, I especially appreciate my mother for her huge assistant during the Covid pandemic. I will forever love every member in my family no matter what happens.

At this moment, I am so grateful in my heart and can't express them all. In my seven years living and studying in MTU, I appreciate everyone who helped and mentored me. For my next journey to be a lecturer in Statistics, I hope I can carry on this kindness and appreciation to my new career and be an amazing instructor who cares and helps my students.

Abstract

This dissertation contains three Chapters. The following is a concise description of each Chapters.

In Chapter 1, we introduced the Random Forest, a machine learning method, to foresee whether a virus is capable of infecting humans. The Covid pandemic informs us the importance of predicting the ability of a zoonotic virus that can infect humans from its genomic sequence. We used the k-mer with $k = 2$ and $k = 3$ as features of a virus to predict if it can affect humans. We further employed the Boruta algorithm to select the important features, then fed those important features into the Random Forest method to train the model and make predictions. After utilizing a dataset that is independent of the training dataset in the test procedure, the results show that the accuracy of the training step is almost the same as an existing model, however, the accuracy in the testing step is substantially improved. Moreover, the time consumption of our method is much less than the existing model.

In Chapter 2, we developed a new application of Long Short-Term Memory (LSTM) deep learning method for the human leukocyte antigens (HLA) allele imputation and implemented it in a software package, called LSTM*HLA. Methods for HLA allele imputation utilize single nucleotide polymorphisms (SNPs) around HLA loci and their relationship with HLA alleles to predict HLA alleles. That is the similar fundamental scheme as Bidirectional LSTM. We organized several consecutive SNPs together as an element of inputs for each cell of the LSTM algorithm and made a final imputation for HLA alleles by averaging results from different sets of hyperparameters. We evaluated and compared the performance of our method with two commonly used methods for HLA imputation with seven real data sets: CookHLA as the representative of conventional approaches and Deep*HLA as the representative of machine learning methods. We find that our method not only performs well when the reference samples and the target samples are from the same ethnic group, but also achieves high accuracy when they are from

distinctive ethnicities. Moreover, because deep learning methods hold the nature that is less dependent on Linkage Disequilibrium, LSTM*HLA could enhance the accuracy of low-frequent HLA alleles which has great influence in the fields of clinical research and personal care.

In Chapter 3, we investigated how two factors, the sample size and the choice of reference samples, can affect the accuracy of HLA imputation since these two factors are important factors that need to be carefully considered in real studies. As our results show, greater than 50 individuals is highly recommended for a reference panel to achieve a high imputation accuracy. For the choice of reference panels, the reference panel with the same ethnicity as target samples is strongly suggested, expanding the reference panel with multiple similar ethnic groups may also improve the accuracy, however, augmenting the reference panel with unrelated ethnic groups would decrease the imputation accuracy.

1 Chapter 1: Using Virus Genomic Sequence to Predict Human Infectious Virus

Abstract

Covid-19 pandemic results in crucial modification in our lives. To prevent the outburst of a new virus, it is crucial to understand if a virus is capable of infecting humans. As the technology of genomic sequence detection is getting mature, it is efficient and consequential if we could analyze the virus characteristics and perform the prediction based on the given sequences. In this paper, we proposed to use the random forest (RF) and the k-mer with $k = 2$ and $k = 3$ as features to predict if a virus is a zoonotic virus and evaluate the extent of the zoonotic virus. One of the main contributions is the use of k-mer to characterize viruses. To extract the important features, we employed the Boruta algorithm. After selecting the important features, we used the random forest method to train the model and make predictions. Our results show that our model has nearly the same error rate in the validation step with an existing method (29.6% vs 29.5%), however, the accuracy is substantially improved in the testing step (79.6% vs 70.8%). Moreover, our method has much less computational time than the existing method (2 hours vs 21 days), thus allows us to quickly predict if a novel virus can infect humans when the novel virus is discovered and sequenced. In summary, we conclude that the random forest with the k-mer as $k = 2$ or $k = 3$ is a more accurate and efficient to predict if a virus can infect humans.

1.1 Introduction

Coronavirus is remodeling our lives and bringing the zoonotic virus into our attention. More and more popular human infectious diseases stem not from humans but other animals. Thus, it is critically important to accurately identify which virus can infect humans. This task is tremendously difficult, since approximately millions of kinds of viruses are detected from other animals, among which, only a tiny portion of them will infect humans [1-3]. As the non-targeted genomic sequencing provides a broad way to explore the genomic sequence for any species, it would be effective and labor-saving to evaluate the hazard of a virus that may infect humans, assuming that we could make this prognosis according to the virus genome sequence. Gauging the virus' risk contributes a crucial significance since it may provide more time for laboratories to analyze those viruses so as to take early steps to prevent rapid transmission of viruses among humans.

There are not too many statistical and computational methods developed to detect human-infecting viruses. One commonly used method is to identify if a virus can infect humans based on the genomic sequences [4, 5]. Such a method may lead to sensitive predictions according to the previous knowledge of the similar viruses [3, 6]. Evidences showed that viruses can be identified using dinucleotide, codon and amino acid biases [7]. Following this direction, the most recent method developed by Mollentze et al. [8] utilized the XGBoost method and the above-mentioned features from both viral and human genomic sequences detect which animal-infecting virus is able to infect humans. The method outperformed a method that is based on the phylogenetic relation of zoonotic viruses. However, the method proposed by Mollentze et al. [8] has several disadvantages. Firstly, there does not exist a valid method to choose the appropriate number of important features such that the model yields the highest accuracy. Secondly, the selected features are highly correlated. Thirdly, the whole procedure is very time consuming, and it may take the program up to three weeks for model building.

The objective of this project is to develop an efficient and accurate method to forecast the probability of a virus that could infect humans based on its genomic sequence. We use the k-mer (substrings of length k contained within a biological sequence) as features to characterize a virus. It is well-known that the k-mer is an essential component in many methods in bioinformatics, and the use of the k-mer yield more robust machine learning features and greater taxonomic accurate than other classifiers [9, 10]. k can be set to any positive integer, however, in our project, we set $k = 2$ and $k = 3$ for its biological reason. Therefore, there are 80 features ($4^2 + 4^3$) to characterize a virus. All of those 80 features may not be equally important, thus we used Boruta, a well-developed package in R, to select important features [11]. After the feature selection step, the random forest (RF) method was used to train the models [12]. We compared our method to the method that uses XGBoost machine learning method and 584 features extracted from the viral and human genomic sequences [8]. Moreover, we added the 2-mer and 3-mer features to the model that is suggested by [8] and checked if there is any improvement in terms of accuracy. Our results show that the k-mer combined with the random forest method using the features selected by Boruta result in a higher accuracy that the method of Mollentze et al. [8] in both training and testing step. Moreover, the computational time of our method is much less.

1.2 Material and Method

Data Set. The species-level viruses' dataset is gathered by incorporating the dataset of [13, 14, 22], following the work of Mollentze et al. [8]. The final dataset for the training consists of 861 virus species from 36 families. There are two criteria that a virus is qualified to infect human are used: (1) it is observed in human by PCR test; and (2) its ability to infect human has been confirmed by its genome sequence. Furthermore, the viruses that can infect humans can be classified into two categories. The first category mainly contains human-to-human transmission viruses (for example, the Dengue virus), while the other one

consists of zoonotic viruses (for example, SARS-CoV-2 which caused COVID-19). The dataset includes 261 human-to-human transmission viruses and 600 zoonotic viruses. The sequences of all these 861 viruses are obtained from RefSeq database (https://www.ncbi.nlm.nih.gov/refseq/). For each virus, all segments of its sequence will be included.

The testing set comes from version #35 ICTV taxonomy release. The family layer of the viruses is approved to incorporate species that infect animals. The novel virus' dataset has 758 viruses which are from 38 families (36 families are the same as the training set, plus Anelloviridae and Genomoviridae). Among that, 113 viruses are certified human-infectious viruses.

Features. The frequency of the k-mer provides a fast and easy way to understand genomic characteristics [23]. We set $k = 2$ and $k = 3$ to compute the substrings in a genomic sequence. $k = 2$ is related with dinucleotide while $k = 3$ is related to 20 types of amino acids. Since there are four kinds of DNA sequence base [A, C, G, T], a total of 80 features ($4^2 + 4^3$) were included in our model. Moreover, we removed all the other letters other than the four base letters in any virus genome sequence. For example, a sequence "TANCG" is regarded as "TACG", and "N" will be dropped from the sequence. For each 2-mer, we calculated its frequency in the corresponding genome sequence, then obtain the probability of that 2-mer by dividing the total frequency of all 2-mer features in the corresponding genome sequence. We repeated this process for the 3-mer features. If the virus contains more than one segment, then the frequency of each feature in all segments was used.

Other numerous features given in [8] were also used. Those features include amino acid biases (20), codon usage biases (62), dinucleotide biases across the entire genome (16), dinucleotide biases across coding regions only (16), dinucleotide biases spanning the bridges between codons (16), and dinucleotide biases at non-bridge positions (16). The above mentioned 146 features are viral genomic features. The third part of features are the features indicating the closeness to human RNA transcript. Each human gene is isolated to three mutually exclusive sections which are interferon stimulated genes (ISGs), non-ISG

housekeeping genes and remaining genes. Following the work given in [8], the R package "EnvStats" was used for measuring the distribution of the observed values for each genome features in all genes in a certain section. Then a similarity score for each feature of each virus was calculated by evaluating the conditional probability density function produced by the R package. Therefore, we obtained three groups of feature sets which are "similarity to ISGs", "similarity to housekeeping genes" and "similarity to remaining genes". Each group has 146 features, thus the third part yields 146*3 features.

Feature Selection. Feature selection is a crucial step in any application of machine learning methods. In this project, not all features in the dataset are significant in predicting the infectivity of a virus. Moreover, lack of feature selection may result in tremendous computational cost and accuracy degeneration [11]. To overcome these disadvantages, we use a wrapper algorithm which is called "Boruta" in R to select important features that will be used in predicting if a virus can infect humans.

The procedure for the feature selection is as follows. Firstly, the original dataset was copied and shuffled to form the "Shadow data". This step is used to add randomness of the given dataset, and Boruta is designed to compare the original data with the random data to detect important features. Then, the Random Forest classifiers were performed on both original and shadow data sets. For each run, we can get all the Z-scores for both original and shadow variables and obtain the maximum Z-score in the shadow attributes (MZSA). Here, a two-sided equality test was applied to all original attributes. The null hypothesis is the importance of each variable is equal to MZSA. We counted the times (denoted as n, usually called "hit") that the Z-scores of attributes are higher than MZSA. The expected number of counts is $E(n) = 0.5n$, and the standard deviation is $SD = \sqrt{0.25n}$. The confidence interval can be derived accordingly. If the hits are substantially larger than the expected value, the attribute is marked as significant. On the contrary, if the hits are substantially lower than the expected value, that attribute is marked as not significant. Furthermore, the importance of all original variables was calculated, and the shadow attributes were disregarded. Finally, the above procedure was repeated until all attributes have been marked or the limit of iteration that was set by the users has reached [24].

In this project, only the significant variables were involved in the training procedure. Though the default iteration number in Boruta is 100, we tried several other iteration numbers which give us a different set of important features. As a result, we found the 500 iterations provide the best accuracy in both the training and the test steps.

Training. We compared several models. The Table 1.1 shows the features to be included in each model:

Table 1.1 The brief description of models and features used.

Model	Features	The number of features
1	k-mer	80
2	Viral Genome	146
3	Viral Genome + k-mer	$146 + 80 = 226$
4	Similarity to Human Transcription	$146 \times 3 = 438$
5	Similarity to Human Transcription + k-mer	$146 \times 3 + 80 = 518$
6	Viral Genome + Similarity to Human Transcription	$146 \times 4 = 584$
7	Viral genome + Similarity to Human Transcription + k-mer	$146 \times 4 + 80 = 664$

Our goal is to evaluate the performance of those models in identifying human infectious viruses and find the one that has the highest accuracy. Among those seven models, the features given in models #2, 4 and 6 are the models used in [8]. In our project, we employed the random forest method due to the high achievement in prediction accuracy and efficiency by its random sampling and bagging scheme [21]. In the random forest procedure, the explanatory variables are the features selected by the Boruta algorithm and the response variable is the infectivity to humans of the virus. We used some default parameters, such as set 500 trees to grow and $mtry = \sqrt{p}$, where $mtry$ indicates the

number of features to be selected in each Random Forest iteration, and p is the number of important features. The cut-off value is sensitive to the prediction accuracy. As suggested in [8], the cut-off should not be 0.5 to balance the error rate between the human infectious and non-human infectious viruses. To select the cut-off, we ran several random forest procedures with the same random seed but different cut-off values, and selected the one yielding balanced error rates, namely cut-off-rf. For example, if we selected the cut-off in the random forest procedure to be (0.322,0.678), then a virus was marked as infectious if at least 32.2% of the trees conclude the virus as human infectious. To reduce randomness, we did 1,000 iterations of the random forest in the training step with the same cut-off values set above and selected the 10% with the smallest overall out of bag error. If several iterations share the same out of bag error, then all of them were selected, hence the number of top iterations selected may be more than 100. Moreover, in each iteration of the random forest classifier, the random seed was set to be the iteration number, and the votes and out of bag times for all 861 viruses were recorded. The probability that a virus may infect human is defined as follows:

$$\text{Prob. (a virus may infect human)} = \frac{\sum vote_i * oob.times_i}{\sum oob.times_i},$$

where $vote_i$ is either 1 or 0, indicating whether the ith virus infects human or not, and $oob.times_i$ gives to total number of iterations in random forest procedure that the ith virus was not selected in the training procedure.

The parameter cut-off-rf was used for the value of $vote_i$. With the above probability, we defined another cut-off, namely the cut-off-balance, such that if this probability is greater than the cut-off-balance, then we marked this virus as human-infectious. Otherwise, the virus was not considered to be able to infect humans. The parameter cut-off-balance is used to balance the true and false error rates in all the iterations. Therefore, we chose a cut-off-balance such that the difference between the true and false error rates achieves its minimum. In general, the cut-off-balance and the cut-off-rf are close.

Prediction on Novel Viruses. To evaluate all the models and methods that we built, a different data set was used. We obtained the same viruses' data that was used in the paper of Mollentze et al. [8]. All species of viruses, confirmed in the version 35 of 2019 ICTV taxonomy (https://ictv.global/taxonomy), are included in the dataset. The family's level of the viruses is recognized to be consist of species that are able to infect human, however, not in the dataset of [6, 13, 14]. Those novel viruses come from 38 families including 36 families which are the same in the training set, the other two families are Anelloviridae and Genomoviridae. The genomic sequence of each virus which is mentioned in the version of vmr_14-010520_MSL35 of the taxonomy (https://ictv.global/vmr) was fetched and calculated for all the features. The selected best 10% of the 1000 iterations with the random forest and XGBoost methods for all models are applied in the testing step. For each iteration, after the prediction through those 500 trees, the votes for not infect or infect for each virus were recorded. The probability that a novel virus may infect human is as follows:

$$\text{Prob. (a virus may infect human)} = \frac{\sum vote\ for\ infect_i}{\#of\ iteration * 500}$$

If the probability is greater than the cut-off-balance which was used in the training step, then that virus was considered to be a human-infectious virus.

To conclude how likely a virus can infect humans, we constructed a 95% confidence interval for each virus by using the votes produced in the top 10% of the 1000 random forest models. We first divided the votes by 500 to obtain the probability that a virus infects humans. A 95% two tailed confidence interval of the probability can be constructed for each virus. If 97.5% of the probabilities of a virus are greater than cut-off-balance, then human infectivity of that virus was categorized as "Very High". If the average of those probabilities is greater than the cut-off-balance, then the infectivity of that virus was identified as a "High". If the average of the probabilities is less than cut-off-balance, then the virus was treated as being at a "Medium" level. Finally, if the 97.5% of those probabilities is less than the cutoff value, the virus has a "Low" level to infect human.

1.3 Results

We compared two methods, the random forest proposed by us and the XGBoost method proposed in [8] with the k-mer features proposed by us and 584 features used in [8]. The results are shown in Table 1.2. It is clear that using the k-mer with the random forest method outperforms the current model. The overall error rate in the validation step for our method is slightly higher than the existing method (29.6% vs 29.5%), nevertheless, the novel virus prediction is far more accurate than the current one (79.6% vs 70.8%). Moreover, we also classified the risk of those predicted human infectious viruses as "High" and "Very High" following the same idea given in [8]. The "Very High" counts from our model is greater than that from the model in [8]. In addition, our method needs much less computational time than the method in [8] (1.5 hours vs 21 days). No matter which set of features considered, the random forest method with the k-mer outperforms the XGBoost method in all aspects, including the validation overall error rate, the prediction accuracy on novel viruses, and the computation time.

Table 1.2 The comparison of the random forest and the XGBoost with the k-mer features and 584 features from [8].

MD	TF	ME	NSF	VS-OER	Confusion Matrix		TTC	PNV	
								# Acc.	Co.
2+3 mer	80	RF	49	29.6%	77/261	422/600	1.5 hours	90 79.6%	V: 75
					184/261	178/600			H: 15
		XGB	80	33.3%	87/261	400/600	21 days	72 63.7%	V: 26
					174/261	200/600			H: 46
146*4	584	RF	91	29.0%	76/261	426/600	1.5 hours	90 79.6%	V: 62
					185/261	174/600			H: 28
		XGB	125	29.5%	77/261	423/600	21 days	80 70.8%	V: 36
					184/261	177/600			H: 44

Abbreviation: MD: Model, TF: Total Number of Features, ME: Method, XGB: Extreme Gradient Boost Method, RF: Random Forest, NSF: Number of Selected Features, VSOER: Validation Step Overall Error Rate, TTC: Training Time Consuming, PNV: Prediction on Novel Virus, Acc.: Accuracy (#/113), Co.: Counts, V: Very High, H: High.

We also investigated the comparison about the correlations of the features. Since the frequency of 2-mers and 3-mers were calculated from the whole virus genomic sequence, we computed the Pearson correlation coefficient between the k-mer features and two subsets of each of the four feature sets used in [8]. The two subsets are dinucleotide biases across the entire genome and codon usage biases. The four feature sets are "Viral genomic features", "Similarity to ISGs", "Similarity to housekeeping genes" and "Similarity to remaining genes". Therefore, for each feature in our model, four correlations were calculated to represent the relationship between the two variables in different models (Table A.1.1). The scatter plots with the regression line fitting the features in our model and corresponding ones in "Viral genomic features" group (Figure A.2.1 and A.2.2)

visually demonstrate the correlation between those pairs. Since two features for tge codon usage bias, ATG and TGG, were not included in the current model, we used the rest 78 features (16 for 2-mers and 62 for 3-mers) in each feature sets to do the comparison. The criteria of correlation classification were based on two conditions. Firstly, if the absolute value of the range of correlation is from 0.5 to 1, then it was considered as highly correlated, if the range is 0 to 0.5, then it was grouped as the low correlation [16]. Secondly, when we make a summary of the correlations, we started from the essential features in our model and counted the number of features which were also marked as crucial in the current model. For each essential feature in our model, it is possible to find more than one corresponding critical feature in the current model. Moreover, if at least one correlation was marked as high, we concluded that the feature is of high correlation in the two models. The summary is as follows. After the feature selection step, 48 features in our model were selected in the next random forest procedure. In those features, 15 features were also selected in the current model and 33 features are not. In those 15 features, 9 of those are considered as highly correlated, while the other 6 are barely correlated. In those 33 features, 20 features are highly correlated and 13 of those are marginally correlated. On the other side, 125 important features were used in the XGBoost procedure. 48 out of 125 are listed in this correlation comparison step. Among that, 23 features were also marked as essential and 25 are not in our model. In those 23 features, 9 features are one to one (5 highly and 4 barely correlated), 4 features are two to one (2 highly and 2 hardly correlated), 2 features are three to one, both two features (CTT, GAA) are highly correlated. Only 18.8% (9/48) of the selected important features in our model are regarded as high correlation with current model which means those two models are running by their own systems.

Furthermore, we tested the stability and optimal running times of Boruta, the method of feature selection. We used the following two sets of features for the evaluation: the k-mer features which contains only 80 features, and the 664 features that include 80 features from the k-mer and 584 features from [8]. For Boruta, a set of 100, 200, 500, 1000 and 2000 iteration numbers, which is also named as the maximum running time, were applied to check the differences in the selected important features. To inspect if the procedure is stable for each set, we ran the procedure 10 times for each maximum running time. In each run,

the Boruta algorithm would appoint its important features. For example, Figure 1.1 shows that the one-time important features selected by Boruta with setting max run = 500 in the k-mer model. We observed that 50 attributes are treated as important features, 15 are considered as unimportant ones and the remaining 15 features are tentative. The overall true and false error rates in the validation step and the number of high and very high-risk viruses in testing step were recorded (Table A.1.2). We summarized the performance of the Boruta and average the overall error rate and the prediction accuracy in the 10 runs based on each element in the set of max run time in Table 1.3. From the table we can conclude that the 500 max running time is the optimal choice in this procedure. For the k-mer, the set of 500 max running time leads to a narrow range of number of selected features (47 to 50 features are preferred), also the average of the overall error rate in the training step will not improve as the running time increases. On average, 90 out of 113 viruses are detected in the independent testing dataset, which is relatively well enough for the testing step. As we increased the maximum running times to 1000 and 2000, the accuracy of the testing step is slightly lower and higher than the set of 500 (79.8% and 80.4%). For the 146*4+80 features, the set of 500 maximum running time results in a smaller range of selected features and the training step overall error rate. In addition, the result based on 500 max running time is considerably better than that based on 100 and 200 max running time, while accuracy gain is tiny if we increased the maximum running time to 500 or more. For the computational time, it took only approximately 6 minutes for 500 maximum running time while about 23 minutes for 2000 max running time. Although the larger max running times result in more stable and narrower range of accuracy, 500 maximum running time in Boruta was suggested and used in this process to balance the training step overall error rate (OER), the testing step accuracy and the computational time. After we set the maximum running time for Boruta, we also compared the performance of three scenarios: the use of intersection or the union of important features in the 10 Boruta procedures, and the use of features selected from the random forest without the use of Boruta. From the results presented in Table 1.4, we can see that the use of the union of features from those 10 Boruta procedures would lead to a better prediction. Based on our results, the use of the intersection of features from those 10 Boruta procedures seems to be a better choice.

However, as the number of features grows, the use of intersection of features are not sufficient to explain the characteristics of viruses which makes the prediction less desirable. The features selected from the random forest without Boruta are based on the Gini index do not perform as well as the features from Boruta. Therefore, we suggest running Boruta 10 times with 500 running in each time and using the union of the features in the real studies.

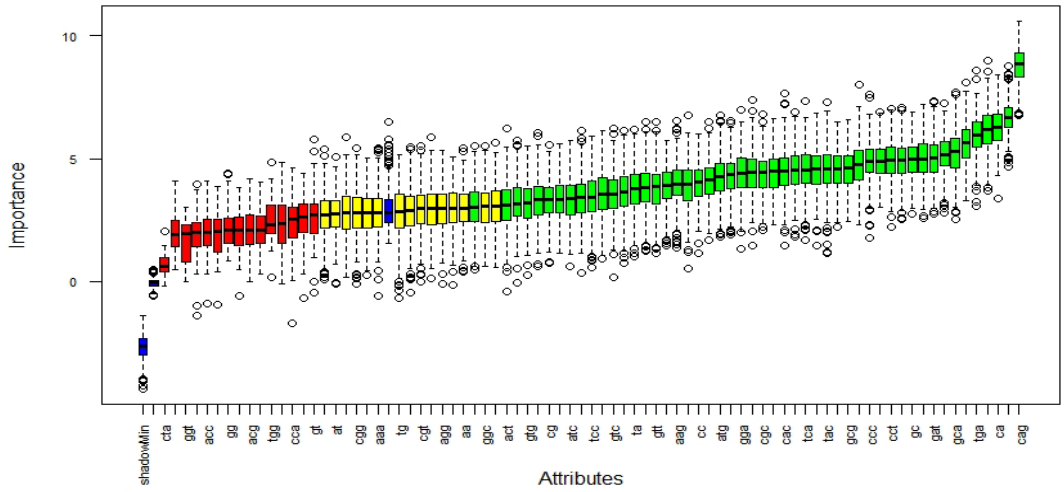


Figure 1.1 Important features selected by Boruta with setting max run is 500 in Our model.

In the figure, the three blue boxplots represent the min, mean, and max shadow features while the green, yellow, and red boxplots represent the important, tentative, and unimportant features, respectively.

Table 1.3 The comparison of different maximum running time in Boruta algorithm.

Model	Max running Time	# Feature Selected		Validation Step OER		Testing Step Acc.	
		Ave.	Range	Ave.	Range	Ave.	Range
80	100	38.1	(35, 42)	0.291	(0.287, 0.295)	79.0%	(76.1%, 80.5%)
	200	43.9	(40, 46)	0.293	(0.287, 0.300)	79.8%	(77.0%, 81.4%)
	500	48.6	(47, 49)	0.295	(0.292, 0.302)	80.0%	(79.6%, 81.4%)
	1000	52.7	(50, 55)	0.295	(0.290, 0.297)	79.8%	(78.8%, 80.5%)
	2000	54.8	(54, 55)	0.295	(0.290, 0.296)	80.4%	(79.6%, 80.5%)
146*4+80	100	25	(20, 30)	0.291	(0.283, 0.302)	61.9%	(54.0%, 69.9%)
	200	40.6	(35, 44)	0.289	(0.275, 0.301)	68.7%	(61.9%, 73.5%)
	500	50.4	(42, 57)	0.287	(0.275, 0.295)	75.1%	(66.4%, 80.5%)
	1000	53	(46, 62)	0.286	(0.275, 0.304)	75.4%	(69.0%, 81.4%)
	2000	54.6	(48, 62)	0.288	(0.278, 0.297)	76.3%	(69.9%, 81.4%)

Abbr.: OER: Overall Error Rate, Ave.: Average, Acc.: Accuracy (Total Number of High and Very High Risk/113). Blue shading represents the optimal choice when Max run is 500.

Table 1.4 The comparison of different sets of selected features in Boruta and the random forest

Model	Select Features From	# Feature Selected	Validation Step OER			Testing Step Acc.		
			OER	TER	FER	#H	#VH	Acc.
80	Intersection	47	0.293	0.291	0.293	15	76	80.5%
	Union	49	0.296	0.295	0.297	14	76	79.6%
	RF Default	80	0.298	0.299	0.297	18	70	77.8%
146*4+80	Intersection	20	0.310	0.310	0.310	18	57	66.4%
	Union	115	0.290	0.291	0.290	16	75	80.5%
	RF Default	664	0.304	0.303	0.305	16	74	79.6%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113). Blue shading represents the optimal choice when union the features of each Boruta iteration.

Finally, we investigated if the use of more features would result in better prediction. Mollentze, et al. have evaluated three models [8]: the model that contains the viral genomic features (146) only, the model that consists of the similarity to human RNA transcript (146*3) features, and the model that combines the features in the first and second models (146*4). After adding the k-mer features (80) to each of three models, we constructed 7 models and evaluated their performance. The results are shown in Table 1.5. For the first group (146 vs 146+80) of models, the outcome of the prediction step is significantly better than the other two groups. However, the overall error rate in the validation step is the worst which may lead to excluding this model at the beginning. Since these 146 and 80 features are directly generated from the virus itself, adding those 80 features would not help increase the testing set accuracy. On the contrary, it will decrease the accuracy (from 87.6% to 85.8%) due to the over-fitting problem. For the second group (146*3 vs 146*3+80) of

models, adding 80 features which are coming from viral contribute to a lower error rate in validation and higher accuracy in prediction. For the third group (146*4 vs 146*4+80), the performance is moderate in both validation and test steps compared to the other two groups. Adding the k-mer features do not decrease the overall error rate in validation and only slightly improve the accuracy of novel viruses (79.6% to 80.5%). Nonetheless, in each group, the additional 80 k-mer features would lead to a greater number of very high-level zoonotic viruses which means compared to viral genomic features in existing models, the use of the k-mer features can help us to detect human infectious viruses more effectively.

1.4 Discussion

As more and more genomic-based viruses are detected, the analytical method for priority ranking for those viruses is still an open problem [8]. Although a number of methods have been developed in the field of ranking preference of the comparably well-characterized viruses according to the scope of the familiar risk elements [13, 17, 18], it is still a barrier that tons of viruses which are not identified with the detailed depiction needed to be applied by that pattern. The genome sequence of viruses has been proved to hold a great extent of the infection ability of viruses [8]. We proposed the use of the random forest and the k-mer features to predict of a virus is a zoonotic virus and the risk level of a zoonotic virus. Our results show that the proposed method provides an efficient and accurate way to predict if a virus can infect humans.

The k-mer is widely used in studying genomic sequences and in a large number of applications in the field of bioinformatics [19]. The k-mer method is studied as extraction and reading the whole genomic sequence with k-length. The “sliding window” scan form is to erase the effect of adopting any position as a starting mark [20]. The value of k can be any integer, however, in this project, we determine $2 \leq k \leq 3$, a narrow reading length resulting from the significance of structure for dinucleotide and amino acid. A set of $k \geq 4$

may lead to very high computational cost. Handling 80 features to any extent for genomic sequencing makes the analysis straightforward and powerful.

Table 1.5 The comparison of seven different models.

Model	Total Features	# Of Selected Features	Validation OER	Confusion Matrix		Prediction on Novel Virus	
						#(Acc.)	Co.
80	80	49	29.6%	77/261	422/600	90 (79.6%)	V: 75
				184/261	178/600		H: 15
146	146	61	31.0%	81/261	414/600	99 (87.6%)	V: 82
				180/261	186/600		H: 17
146+80	226	74	31.0%	81/261	414/600	97 (85.8%)	V: 85
				180/261	186/600		H: 12
146*3	438	81	29.9%	78/261	421/600	80 (70.8%)	V: 58
				183/261	179/600		H: 22
146*3+80	518	94	27.5%	71/261	434/600	84 (74.3%)	V: 61
				190/261	166/600		H: 23
146*4	584	91	29.0%	76/261	426/600	90 (79.6%)	V: 62
				185/261	174/600		H: 28
146*4+80	664	115	29.0%	76/261	426/600	91 (80.5%)	V: 75
				185/261	174/600		H: 16

Abbr.: OER: Overall Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113), Co.: Count. Blue Shading represents the number of correctly predicted viruses.

After the genomic sequence has been extracted, we proposed a model that is suitable for complicated biological dataset. The random forest method preserves the advantages of decision trees, moreover, reaching optimal consequences in consideration of random selected subsets of features, bagging sampling and the scheme of majority voting [21]. The prerequisite of the random forest is to select essential features in the high-dimensional bioinformatics dataset to evade the overfitting problem, produce robust model performance, and reduce the computation time. The random forest uses the Gini importance to choose the important features. However, the use of features from the random forest does not perform as well as the use of features from Boruta (Table 1.4).

Another thing that needs to be noticed is that the features ATG and TGG were removed in the model given in [8]. However, the feature ATG is selected as important features in our model. ATG is one of the starting codons and whether this feature should be kept or erased is worth discussing. We improved the results upon [8] in two aspects, slightly in the training step while significantly in the testing step. Nevertheless, there is a drawback of the proposed method. The feature selection algorithm, Boruta, is not quite stable. The use of 80 features from the 2-mer and 3-mer is not problematic when the max run is set to 500. As more features are added to the model, the optimal max run time is still needed to be determined. For example, a model with 146*4+80 features, the use of the intersection of those selected features in the 10 runs should result in a better prediction accuracy. However, the intersection and union of the features in the 10 runs contain 20 and 115 features, respectively. Therefore, it is hardly consistent in every run of Boruta with 500 max running time. Moreover, the random forest and Boruta with 80 features from the 2-mer and 3-mer and 146 viral genomic features from [8] are two kinds of methods to characterize viruses. The use of k-mer features improves the prediction in the validation step while has about 8% decrease accuracy in the testing step (Table 1.5). This indicates that these 80 features from the 2-mer and 3-mer may not be sufficient to characterize the virus. For the future work, as suggested in the current model, 146*4 is the most valuable model, we would work on applying k-mer method to the features that is related to human RNA transcription, then combining with 80 viral features that may anticipate an even more accurate result in the field of prediction zoonotic viruses. Furthermore, we only set $k = 2$

and $k = 3$ in this project, k-mer method can set k to be any positive numbers. Increasing the setting of k may turn into another aspect to further increase the prediction accuracy.

1.5 Reference

1. Carroll, D., et al., *The Global Virome Project.* Science, 2018. **359**(6378): p. 872-874.
2. Holmes EC, R.A., Andersen KG. , *Pandemics: spend on surveillance, not prediction.* Nature, 2018(558): p. 180-182.
3. Wille M, G.J., Holmes EC., *How accurately can we assess zoonotic risk?* PLOS Biology, 2021. **19**(e3001135).
4. Zhang Z, C.Z., Tan Z, Lu C, Jiang T, Zhang G, et al., *Rapid identification of human-infecting viruses.* Transbound Emerg Dis., 2019. **66**: p. 2517-22.
5. Bartoszewicz, J.M., A. Seidel, and B.Y. Renard, *Interpretable detection of novel human viruses from genome sequencing data.* NAR Genom Bioinform, 2021. **3**(1): p. lqab004.
6. Mollentze, N. and D.G. Streicker, *Viral zoonotic risk is homogenous among taxonomic orders of mammalian and avian reservoir hosts.* Proc Natl Acad Sci U S A, 2020. **117**(17): p. 9423-9430.
7. Babayan, S.A., R.J. Orton, and D.G. Streicker, *Predicting reservoir hosts and arthropod vectors from evolutionary signatures in RNA virus genomes.* Science, 2018. **362**(6414): p. 577-+.
8. Mollentze, N., S.A. Babayan, and D.G. Streicker, *Identifying and prioritizing potential human-infecting viruses from their genome sequences.* Plos Biology, 2021. **19**(9).
9. Ul Alam, M.N. and U.F. Chowdhury, *Short k-mer abundance profiles yield robust machine learning features and accurate classifiers for RNA viruses.* Plos One, 2020. **15**(9).

10. Ainsworth, D., et al., *k-SLAM: accurate and ultra-fast taxonomic classification and gene identification for large metagenomic data sets.* Nucleic Acids Research, 2017. **45**(4): p. 1649-1656.
11. Kursa, M.B. and W.R. Rudnicki, *Feature Selection with the Boruta Package.* Journal of Statistical Software, 2010. **36**(11): p. 1-13.
12. Biau, G., Scornet, E., *A random forest guided tour.* TEST, 2016. **25**: p. 197-227.
13. Olival, K.J., et al., *Host and viral traits predict zoonotic spillover from mammals (vol 546, pg 646, 2017).* Nature, 2017. **548**(7669).
14. Woolhouse, M.E.J. and L. Brierley, *Epidemiological characteristics of human-infective RNA viruses.* Scientific Data, 2018. **5**.
15. Chor, B., et al., *Genomic DNA k-mer spectra: models and modalities.* Genome Biology, 2009. **10**(10).
16. Asuero, A.G., A. Sayago, and A.G. Gonzalez, *The correlation coefficient: An overview.* Critical Reviews in Analytical Chemistry, 2006. **36**(1): p. 41-59.
17. Pulliam, J.R.C. and J. Dushoff, *Ability to Replicate in the Cytoplasm Predicts Zoonotic Transmission of Livestock Viruses.* Journal of Infectious Diseases, 2009. **199**(4): p. 565-568.
18. Grange, Z.L., et al., *Ranking the risk of animal-to-human spillover for newly discovered viruses (vol 118, e2002324118, 2021).* Proceedings of the National Academy of Sciences of the United States of America, 2021. **118**(39).
19. Manekar, S.C. and S.R. Sathe, *A benchmark study of k-mer counting methods for high-throughput sequencing.* Gigascience, 2018. **7**(12).
20. Sievers, A., et al., *K-mer Content, Correlation, and Position Analysis of Genome DNA Sequences for the Identification of Function and Evolutionary Features.* Genes, 2017. **8**(4).
21. Qi, Y., *Random Forest for Bioinformatics*, in *Ensemble Machine Learning: Methods and Applications*, C. Zhang and Y. Ma, Editors. 2012, Springer US: Boston, MA. p. 307-323.
22. Mollentze, N. and D.G. Streicker, *Viral zoonotic risk is homogenous among taxonomic orders of mammalian and avian reservoir hosts.* Proceedings of the

National Academy of Sciences of the United States of America, 2020. **117**(17): p. 9423-9430.

23. Ghandi, M., et al., *Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features.* PLoS Computational Biology, 2014. **10**(7): p. e1003711.

24. Kursa, M.B., A. Jankowski, and W.R. Rudnicki, *Boruta – A System for Feature Selection.* Fundamenta Informaticae, 2010. **101**: p. 271-285.

2 Chapter 2: LSTM*HLA – HLA Allele Imputation with Long Short-Term Memory Machine Learning Method

Abstract

The current human leukocyte antigens (HLA) allele imputation methods utilize intergenic SNPs around the HLA loci to predict HLA alleles and enables the fine mapping of HLA alleles for immune phenotypes with a limited cost. We developed a deep learning method for the HLA allele imputation, LSTM*HLA, which employs Bi-directional Long Short-Term Memory (LSTM) machine learning method as its engine to impute HLA alleles. We grouped several consecutive SNPs together and used them as the input of the cells in the LSTM algorithm and made a final imputation for HLA alleles by averaging results from different sets of hyperparameters. We compared the proposed method with two commonly used methods for the HLA allele imputation using seven real datasets from different ethnic groups. Our results show that our approach outperforms (the first rank in 28 out of 40 pairs) than other two methods in matched ethnic panels and unmatched ethnic panels. Furthermore, this proposed approach enhances the imputation accuracy of low-frequent alleles which attains far-reaching influence in the areas of personal care and clinical research.

2.1 Introduction

Human Leukocyte Antigen (HLA) is a gene complex located on chromosome 6 that plays a crucial role in the human immune system by presenting antigens to T cells [1]. A prerequisite for comprehending the manner that major histocompatibility complex (MHC) involved in human immune diseases is to find associated HLA alleles [2]. However, the HLA system is highly polymorphic, with over 25,000 known HLA alleles, making it challenging to genotype and analyze [3]. Moreover, to have enough power to find HLA alleles that are associated with human immune diseases, it is generally necessary to require that hundreds even thousands of individuals are genotyped at HLA loci which is prohibitively expensive for many studies. Alternatively, a time and cost effective approach is to impute HLA alleles based on the SNPs that are around the HLA loci and the linage disequilibrium (LD) between SNPs and HLA alleles [4-6]. These SNP-based methods for the HLA allele imputation are accessible on the board of Genome-Wide Association Studies (GWAS) [7]. Alongside its labor-saving and no surcharge, the imputation techniques are widely used and becoming increasingly popular to investigate the association between HLA alleles and human immune diseases [8-15].

A number of methods for the HLA allele imputation have been developed and they are mainly divided into two categories. One group of methods includes traditional approaches based on the expectation-maximization (EM) algorithm or the Hidden Markov Model (HMM), such as Beagle, HIBAG [6], SNP2HLA, and CookHLA. CookHLA utilized Beagle v5 as its engine and creatively introduced two strategies, locally embedding markers and adaptively learning genetic map, to enhance the prediction accuracy [16]. CookHLA has proved that it accomplishes a greater accuracy than HIBAG and SNP2HLA with powerful improvement in uncommon alleles [17]. The other group of methods is based on modernized machine learning approaches. The representative method is Deep*HLA, where a deep learning method, Convolutional Neural Networks (CNN) was applied. Although all the current approaches may achieve high accuracy, it is undeniable that those methods have some limitations. When predicting 4-digits HLA alleles, the average failure rate of those techniques is from 3% to 6%. It is even worse when anticipating the low-frequencies alleles, since their LD between SNPs and HLA alleles becomes much weakened, and the reference panel may not capture abundant information

for those alleles [16]. Another limitation is that prediction accuracy highly depends on the quality of reference panels and the available methods do not perform well when the ethnicities of the reference sample and the target samples are not well-matched. It is desirable to choose a refence panel such that the ethnicity of the reference samples and the ethnicity of the target samples are well matched [16]. However, in practice the ethnicities of the reference sample and the target samples may not be well-matched even though both the reference samples and the target samples are from the same/similar ethnic groups. Moreover, each approach has its own vulnerability. For example, CookHLA is quite time consuming [17] while Deep*HLA does not perform well when the reference samples and the target samples are from slightly distinctive ethnic groups.

Here, we proposed a deep learning method, LSTM*HLA (**L**ong **S**hort-**T**erm **M**emory for the **HLA** allele imputation) to impute HLA alleles. Compared with other existing methods, LSTM*HLA achieves higher imputation accuracy by several strategies. Firstly, we transferred Bidirectional Long Short-Term Memory (LSTM) architecture to better capture the relationship between HLA alleles and SNPs surrounding them. This LSTM method is primarily designed for the purpose of sequence prediction and well-used in the area of Natural Language Processing (NLP) [18]. Secondly, before feeding into LSTM, we partitioned the SNPs around HLA positions into different groups so as to drive the entire process efficiently. Such treatment can effectively reduce the noise in the data. Thirdly, we repeated the procedure with different sets of hyperparameters, and averaged the dosage imputed by each set of hyperparameters. We applied our method to several real data sets and compared its performances with two existing methods for the HLA allele imputation: CookHLA and Deep*HLA. The results show that LSTM*HLA outperforms CookHLA and Deep*HLA when the reference samples and the target samples are from different or similar ethnic groups. For example, when the reference samples are from Southern Asian (SAS) and the target samples are from African (AFR) thus the reference samples and the target samples are from the different ethnic groups here, the accuracy of our method is 50.8% while CookHLA, obtains the accuracy of 49%, LSTM*HLA increase accuracy by 1.8%. As another example, when the reference samples are from SAS and the target samples are from EAS thus the reference samples and the target samples are from the

similar ethnic groups here, the accuracies of our method and second place (CookHLA) are 78.0% and 74.8%, respectively. LSTM*HLA has increased accuracy by 3.2%. More importantly, our method does a superior work when imputing low-frequent HLA alleles which play important roles in studying the association of HLA alleles and immune system phenotypes.

Finally, it is worth noting that the traditional HLA imputation approaches which utilize the hidden Markov model hold a robust relationship with LD. A strong LD may result in an optimal imputation, while a weak LD poses challenges in the HLA allele imputation [17]. Nonetheless, deep learning methods perfectly evade this feature. Deep learning models could evaluate sophisticated LD architecture between HLA alleles and their surrounding SNPs by the essence of neural networks [17, 30]. LSTM as a deep learning method, and it does not completely count on LD to impute alleles.

2.2 Material and Method

Data Sets

1000 Genomes Data. The datasets were downloaded from the website of 1000 Genomes Project (https://www.internationalgenome.org/). The 1000G project [19] contains samples from five ethnic groups: 661 African (AFR) individuals, 347 Admixed American (AMR) individuals, 504 East Asian (EAS) individuals, 503 European (EUR) individuals, and 489 South Asian (SAS) individuals. In the MHC region (chr6:28-35 Mb), 225,194 SNPs were genotyped with the next sequencing technology. Four-digit alleles at five HLA loci, HLA-A, HLA-C, HLA-B, HLA-DRB1, and HLA-DQB1 were obtained. HLA genotype information was extract from the Next-Generation Sequencing (NGS) dataset using PolyPheMe, a software of in-silico typing [20]. 5,539 SNPs that match the Illumina Infinium genotyping chip (Immunochip) data were finally used in our work, as the

Immunochip data were designed both to perform deep replication of major autoimmune and inflammatory diseases, and fine mapping of established GWAS significant loci.

Korean Data. The dataset contains 413 individuals and was downloaded from the website that is detailed in [21]. In the MHC region (chr6:25-35 Mb), 5,858 SNPs were genotyped using the HumanOmniExpress platform from Illumina. Four-digit HLA allele are available at 9 HLA loci, including HLA-A, HLA-C, HLA-B, HLA-DRB5, HLA-DRB4, HLA-DRB3, HLA-DRB1, HLA-DQB1, and HLA-DPB1. HLA genotype information was extracted by applying Roche 454 sequencing from the Institute for Immunology and Infectious Diseases in Australia. The American Society for Histocompatibility and Immunogenetics also accredited the algorithms of calling for those HLA alleles [21].

Pan-Asian Data. The dataset was downloaded from the website of SNP2HLA (https://software.broadinstitute.org/mpg/snp2hla/) which was described in details in [5, 22, 23]. It contains 530 samples consisting of 91 Singapore Chinese individuals, 111 Chinese individuals, 119 Indian individuals, 120 Malaysian individuals, and 89 Japanese and Han Chinese individuals from the HapMap Project Phase II data. In the MHC region (chr6:25-35 Mb), 6,173 SNPs were genotyped using Affymetrix Genome-Wide Human SNP Array 6.0 and Illumina HumanHap 1M. Four-digit alleles are available at eight HLA loci: HLA-A, HLA-C, HLA-B, HLA-DRB1, HLA-DQA1, HLA-DQB1, HLA-DPA1, and HLA-DPB1. HLA genotype information was extracted by an approach of Sequence-Based Typing (SBT) along with a sequence analysis based on taxonomy[22, 23].

Data pre-processing. The original data files are in bgl.phased format which is used by the Beagle software package and contain the coded HLA alleles: each allele at a HLA loci is coded either as “P” or “A”. Since there are two alleles at each HLA loci, there should be exactly two “P” for alleles at an HLA locus. We removed the individuals who have either zero “P” or more than two “P” at any of HLA loci. The number of samples retained in each panel is as follows. 654 AFR individuals, 341 AMR individuals 498 EAS samples, 497 EUR samples, and 482 SAS samples were retained. For the KOR data sets, 400 samples

were retained. For the Pan-Asian data, 486 samples were retained. SNPs within 500kb downstream and upstream of an HLA locus were used for the HLA allele imputation.

Brief introduction of single cells in LSTM*HLA. Long Short-Term Memory (LSTM) machine learning method proposed by Hochreiter et al. [24, 25] exhibit high accuracy for long sequences predictions due to its capacities to maintain essential message over time [26]. The architecture of a single cell in LSTM*HLA (Figure 2.1a.), includes a cell state (c) which is long-term memory over the full DNA sequence data, a hidden state (h) which represents a short-term vision or output from preceding cell, the input SNPs vector x_t consists 1 and 0. The non-linear reliance, which is recognized as gates manipulates the materials to be retained or removed. The non-linear reliance is the combination of activation functions which are sigmoid (σ) and tangent hyperbolic $(tanh)$ along with operations such as element-wise multiplication $(\times)$ and concatenation $(+)$. In addition, c_t and c_{t-1} represent the cell state from the current and previous cell, respectively. h_t and h_{t-1} are the hidden state from the current and previous cell, respectively. f_t (Formula (2)) is the forget gate which is used to establish the information to be preserved from the cell state. i_t and g_t (Formula (1) and (3)) are the first and second layers of input gate which is to determine the new message that can be saved in cell state. o_t (Formula (4)) is the output gate which integrate with new cell state (c_t, Formula (5)) to generate advanced hidden state (h_t, Formula (6)) that move on to the input of next cell. W and b in the formulas represent weight and biases [26, 27], respectively.

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (1)$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (2)$$

$$g_t = tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (3)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

$$\sigma(\boldsymbol{z})_i = \frac{e^{z_i}}{\Sigma_{j=1}^{K} e^{z_j}} \quad (7)$$

$$Se(L) = \frac{\sum_{i=1}^{n}(D_i(A1_{i,L}) + D_i(A2_{i,L}))}{2n} \quad (8)$$

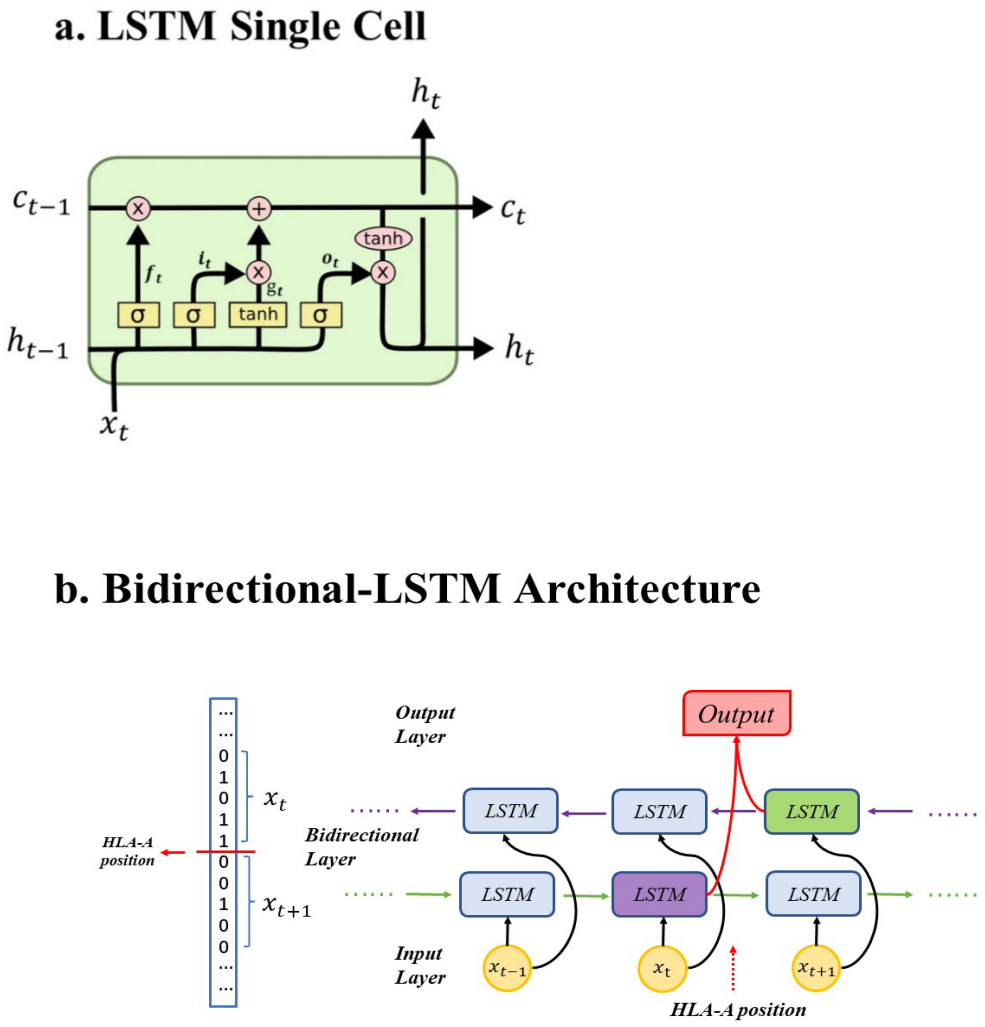


Figure 2.1 Overview of LSTM*HLA method.

a. LSTM Single Cell Architecture. **b.** Bidirectional-LSTM Architecture.

Bidirectional LSTM (Bi-LSTM) structure. To improve the performance of the LSTM, the bidirectional LSTM (Bi-LSTM) is developed. The Bi-LSTM is more powerful when dealing with contextual sequence because the output of any given cell counts on the information on both sides of that cell [28, 29]. The architecture of Bi-LSTM (Figure 2.1.b.) possesses a bidirectional layer, a forward layer and a backward layer. Every unit in the layer captures the information from the previous and the forthcoming sequence. In this research, we developed two models based on the Bi-LSTM structure for the HLA allele imputation. The difference between these two models is based on what set(s) of hyperparameters are used. The first mode is called the one-fit-all model and is named LSTM*OFA, where a single and optimal set of hyperparameters is used for the HLA allele imputation. The optimal set of hyperparameters is selected based on the cross-validation or a pair of the reference and target panels. Such an approach is widely used in the deep learning field. For the second approach, namely the LSTM*HLA model, several sets of hyperparameters are applied to determine the final imputation results. For those two models we proposed, we utilized the phased haplotypes at the SNPs and the HLA loci. If we use one SNP at a time as a cell in LSTM(x_t), we may suffer seriously elevated error rate due to the noise in the dataset. In addition, the computational time may be a concern since the back-propagation process spans over a large number of SNPs. To overcome those shortcomings, we grouped the SNPs around each HLA locus and used the groups of SNPs as the input in a LSTM cell. For example, if there are 112 SNPs in the downstream and 143 SNPs in the upstream of HLA locus, we discarded the 2 SNPs in the downstream and 3 last SNPs in the upstream and use each 10 consecutive SNPs as a group for the input (x_t), resulting 11 and 14 groups in the downstream and the upstream, respectively.

The general procedure in the training step is illustrated in Figure 2.1.b for the HLA-A locus. In this example, we considered 5 SNPs per group and the LSTM cell that is in purple. The input part in the forward layer contains two components, the vector of alleles at 5 SNP loci (<0, 1, 0, 1, 1>) (x_t) and the output of the previous cell's hidden state in the forward layer (h_{t-1}). We pre-select a sequence of the hidden dimensions, the size of the hidden state, and the learning rates, the pace at which an algorithm learns. After going through the input, the forget and output gates by one group of hyperparameters of hidden dimension

and learning rate, the outcome of h_t is a sequence of numbers for HLA-A possible alleles. Meanwhile, the HLA-A alleles are not only related to the SNPs in the upstream, but also the SNPs in the downstream. Thus, the HLA-A alleles are also imputed through the LSTM cell in green in backward layer. The input components in the backward layer contains the alleles at the 5 SNPs in the downstream (<0, 0, 1, 0, 0>) (x_{t+1}) and the output of the previous cell in the backward layer (h'_{t+2}). Again, the output of h'_{t+1} is another sequence of numbers for the HLA-A possible alleles. Then the Softmax function (Formula (7) where vector $\boldsymbol{z}$ contains K possible alleles) is used to obtain the probability of each allele at HLA-A. The allele with highest probability would be imputed as the HLA-A allele for that haplotype. All the W (Weights) and b (biases) will be updated by backpropagation so as to learn the optimal set of hyperparameters in the LSTM cell. The procedure is repeated $2N$ times as an epoch where N is the number of reference individuals. The process is terminated if the training accuracy does not increase within 20 epochs. The two HLA-A alleles imputed with the highest probability from two haplotypes of that individual are used to form the corresponding genotype. The following general settings or hyperparameters were used: 1 hidden layer, 300 epochs, learning rate of 0.01 at start for the Adam optimizer, and the categorical cross entropy loss function.

One-fit-all model. This model applies an optimal group of hyperparameters from one pair of reference and target panels to other reference-target pairs with the same reference panel. The candidate sets of hyperparameters are the hidden dimension, from 10 to 100 with the step size 10; the learning rates, from 0.01 to 0.1 with the step size 0.01 together with 0.15, 0.20, 0.25, 0.30; the number of SNPs per group, from 10 to 150 with step the size 10, and 200, 250. For a given pair of the refence and target panels, the following sequential procedure was used to determine the optimal set of hyperparameters. We first found the optimal number of SNPs per group by setting a certain number of hidden dimensions and learning rate, say 30 and 0.05. After employing Bi-LSTM approach 17 times (because the choices for the number of SNPs per group is 17), we elected the best number of SNPs per group with the highest accuracy of average imputation over all the possible HLA alleles for target panel. Secondly, we chose the learning rate parameter by the fixed number of SNPs per group and the fixed hidden dimension, also applying Bi-

LSTM approach 14 times (since there are 14 possible candidates for this parameter). Following the same criterion as the first parameter, we selected the learning rate that has the highest imputation accuracy. Lastly, we chose the last parameter, the hidden dimension, with a similar procedure.

LSTM*HLA model. Instead of using a single set of hyperparameters to impute HLA alleles, we used multiple sets of hyperparameters and combined imputation results. Let $p_1^{(l)}, p_2^{(l)}, \cdots, p_K^{(l)}$ ($l = 1,2, \cdots, L$) be the probabilities for K alleles at an HLA locus obtained from the lth set of hyperparameters. The final probability of the kth ($k = 1,2, \cdots, K$) allele is just the average of the L probabilities ($\frac{2}{L}\sum_{l=2}^{L} p_k^{(l)}$) and the allele with the highest final probability is considered as the imputed allele. Specifically, the hidden dimension (H) is set as 30, 50, or 100, learning rate (L) is set as 0.05, 0.10, or 0.2. The number of SNPs per group (S) is set as 20, 50, 80, or 120. This results in 36 sets of hyperparameters.

2.3 Results

Comparison of imputation accuracies. We compared the imputation accuracies for our method, LSTM*HLA and two existing methods, CookHLA and Deep*HLA. CookHLA employs the Beagle version 5 [34] which is based on the hidden Markov model. It has been shown that CookHLA outperforms SNP2HLA [5], a method based on the earlier version of Beagle and HIBAG-fit [6] and HIBAG-prefit [6], a method for HLA allele imputation based on the EM algorithm and the attribute bagging. Deep*HLA is a modern deep learning method for the HLA allele imputation that employs the convolution neural network (CNN) as its engine. We used the same datasets for those three methods for a fair comparison. For each HLA locus/allele, the accuracy is defined as the proportion of correctly imputed alleles over the total number of alleles.

Firstly, we investigated the performance when the reference and target panels are from the same ethnic group. The reference-target pair of the SAS and Pan-Asian datasets was used in this analysis. We calculated the accuracies at five common HLA loci,HLA-A, HLA-C, HLA-B, HLA-DRB1, and HLA-DQB1. CookHLA is not able to provide any results when the Pan-Asian dataset or the KOR dataset was chosen as a reference panel. The results when the SAS dataset was used as the reference and the Pan-Asian dataset was used as the target panel are shown in Figure 2.2.a. CookHLA is not able to provide any results when the Pan-Asian dataset or the KOR dataset was chosen as a reference panel so only the results from LSTM*HLA and Deep*HLA are shown in Figure 2.2.a. Figure 2.2.a shows that LSTM*HLA has a much higher overall accuracy of 78.7% whereas the overall accuracy for Deep*HLA is only 69.5%. Compared toDeep*HLA, the error rate for LSTM*HLA is reduced 9.2% (from 30.5% to 21.3%). The highest error rate occurs at HLA-DQB1 and HLA-HLA-DRB1. For HLA-DQB1, LSTM*HLA has an accuracy of 75.6% and with reduces 15.6% error rate, in contrast to the accuracy of 60.0% for Deep*HLA. For HLA-HLA-DRB1, the accuracies are 78.0% and 64.1% for LSTM*HLA and Deep*HLA, respectively so the error rate for LSTM*HLA is reduced 13.9%. Those two improvements are very promising as those two loci play a crucial role in plenty of autoimmune and heritable phenotypes, and HLA-DRB1 is normally the most challenging one to predict correctly [10, 14, 15, 31, 32]. When the Pan-Asian dataset was used as the reference panel and the dataset was used the SAS as the target panel, the outcomes in Figure 2.2.b show that the overall accuracy for LSTM*HLA is higher than it for Deep*HLA (86.1% vs 83.9%) (Figure 2.2.b.). This is mainly because Deep*HLA has already made reliable imputation, so any significant improvements can hardly be achieved. The essential contribution to the improvement in terms of accuracy in this case is the HLA-A with the accuracy of 91.7% (error rate 8.3%) for LSTM*HLA and 88.4% (error rate 11.6%) for Deep*HLA. Compared to Deep*HLA, LSTM*HLA performs outstandingly in all five HLA loci for those two pairs.

Secondly, we investigated the performance of LSTM*HLA, CookHLA, and Deep*HLA when the reference and target panels are from the different ethnic groups. The SAS (South Asian, N=489) datasets was still used while a new dataset, the AFR (African, N=661)

dataset was used. Those two panels are from quite disparate ethnic groups. Five HLA, loci HLA-A, HLA-C, HLA-B, HLA-DRB1, and HLA-DQB1 were used in this comparison. When the AFR and SAS datasets were chosen as the reference and target panels, respectively, the average accuracy for LSTM*HLA is slightly higher than CookHLA and much higher than Deep*HLA (73.0% vs 74.7% vs 57.3%) (Figure 2.2.c.). Correspondingly, compared to CookHLA, the error rate for LSTM*HLA reduces from 27% to 25.7% with a reduction of 1.3%. However, compared to Deep*HLA, the error rate for LSTM*HLA reduces from 42.7% to 25.7% with a reduction of 17%. Compared to CookHLA, LSTM*HLA yields higher accuracies in four out of five HLA loci (HLA-C, -B, -DRB1, -DQB1) with the accuracy increased by 1.2%, 2.1%, 3.2% and 1.4%, respectively. LSTM*HLA only has a slightly lower accuracy CookHLA (80.6% versus 81.2%) for one locus (HLA-A). When the SAS dataset was used as the reference panel and the AFR data set was used as the target panel, the imputation accuracy became much lower as we expected. The overall accuracies were only 50.8% for LSTM*HLA, 36.8% for Deep*HLA, and 49.0% for CookHLA, respectively. Same as the last comparison, CookHLA has the lowest accuracy for all five HLA loci, especially for HLA-DQB1, while LSTM*HLA and CookHLA have much higher accuracies. The imputation accuracies for HLA-DQ1 for LSTM*HLA, CookHLA, and Deep*HLA are 72.6%, 64.0%, and 44.0%, respectively. In addition, LSTM*HLA has the highest accuracy for three out of five loci (HLA-C, -DRB1, -DQB1), and has the lower accuracy for the other two HLA loci (HLA-A and HLA-B).

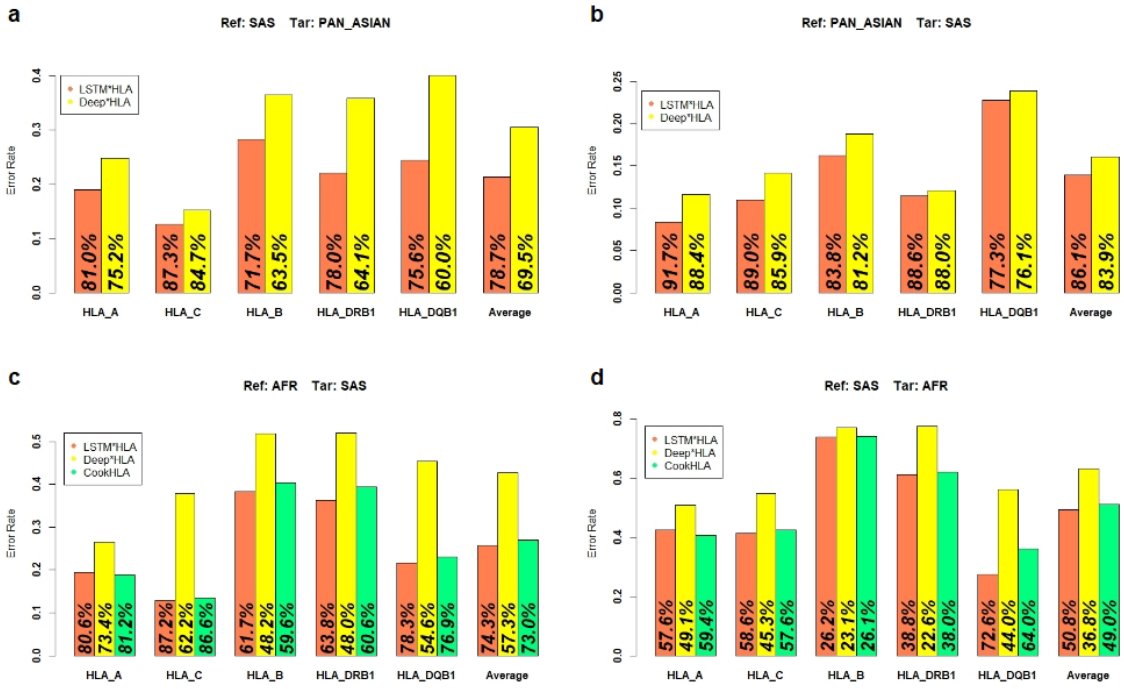


Figure 2.2 Comparison of imputation accuracies.

a. Imputation accuracy of HLA for using 1000G SAS as reference and Pan-Asian as target. **b.** Imputation accuracy of HLA for using Pan-Asian as reference and 1000G SAS as target. **c.** Imputation accuracy of HLA for using 1000G AFR as reference and 1000G SAS as target. **d.** Imputation accuracy of HLA for using 1000G SAS as reference and 1000G AFR as target. Population size: 1000G SAS (N=489), Pan-Asian (N=530), 1000G AFR (N=661). The bold percentage in each bar represents the accuracy of each category.

Comparison of overall imputation accuracy with matched and unmatched reference panels. A sizable reference panel with the same ethnicity as the target panel is quintessential for the HLA allele imputation [33]. This considerable panel is presently available in the European population as well as East Asian population [34, 35]. Other large panels of ethnicities have not been assembled yet. Two schemes for investigation of HLA genes in those panels were either constructing a reference group that is similar to the target group or working with a random reference that the ethnic group could be various from the original group. Both proposals carried their own disadvantages for the HLA allele

imputation. The difference of the ethnicities between the reference and target panels could greatly induce the imputation accuracy. Thus, it is important to investigate how these methods perform when different reference and target panels are used.

We collected 7 datasets to compare the overall accuracy for the HLA allele imputation and to see if our new model still outperforms the existing methods. The five datasets from 1000 Genome project are from African (AFR with N=661), Admixed American (AMR with N=347), East Asian (EAS with N=504), European (EUR with N=503) and South Asian (SAS with N=489). The other two datasets are the Korean panel (KOR with N=413) and the an-Asian panel (N=530) which are from the east and south of Asia, respectively and include some overlapping samples. Therefore, those two pairs, the EAS data as the reference panel and the Pan-Asian dataset as the target panel, and the Pan-Asian dataset as the reference panel and the EAS dataset as the target panel, were excluded from the comparison. This leads to totally 40 reference-target pairs for the comparison ($7 \times 6 - 2 = 40$). For every reference-target pair, we employed CookHLA, Deep*HLA, along with our One-fit-all LSTM (LSTM*OFA) model and LSTM*HLA model. LSTM*OFA model was to randomly select a pair (marked as * in Table 1) to choose the optimal set of hyperparameters with the highest overall accuracy and then applied to the set of hyperparameters chosen to other reference-target pairs with the same reference panel. For CookHLA, we could not successfully obtain the results when either the KOR dataset or the Pan-Asian data was used as reference panel.

Table 2.1 Pairwise comparison with seven reference panels for four models.

		Reference Panel							Method
		EUR	KOR	EAS	PA	SAS	AFR	AMR	
	Population	EU	EA	EA	EA, SA	SA	AF	AA	
	Sample Size	503	413	504	530	489	661	347	
Target Panel	EUR		74.6%	75.7%	71.1%	82.0%	89.7%	94.6%	LSTM*HLA
			68.1%	71.9%	65.2%	81.4%*	85.9%	92.7%*	LSTM*OFA
			65.9%	61.6%	48.6%	70.1%	81.5%	87.9%	Deep*HLA
			--	78.1%	--	84.4%	89.4%	94.9%	CookHLA
	KOR	65.2%		88.6%	84.1%	73.8%	56.4%	62.9%	LSTM*HLA
		63.0%		82.8%	84.4%*	69.5%	53.4%	59.7%	LSTM*OFA
		61.6%		91.3%	86.3%	65.8%	52.7%	64.8%	DeepHLA
		63.8%		89.7%	--	63.8%	55.1%	61.9%	CookHLA
	EAS	66.3%	88.3%			78.0%	52.1%	62.9%	LSTM*HLA
		64.3%	86.5%			74.2%	46.5%	57.6%	LSTM*OFA
		48.6%	84.4%			55.5%	38.9%	49.0%	DeepHLA
		63.4%	--			74.8%	53.3%	62.7%	CookHLA
	Pan-Asian	58.6%	78.4%			78.7%	49.0%	56.0%	LSTM*HLA
		54.6%	76.8%*			74.0%	45.2%	52.1%	LSTM*OFA
		49.2%	71.9%			69.5%	44.2%	49.3%	DeepHLA
		57.6%	--			70.4%	48.7%	53.1%	CookHLA
	SAS	78.8%	77.8%	86.7%	86.1%		74.3%	77.5%	LSTM*HLA
		79.6%*	74.4%	84.7%*	84.5%		66.4%	74.1%	LSTM*OFA
		65.0%	71.7%	79.7%	83.9%		57.3%	67.7%	DeepHLA
		76.3%	--	85.0%	--		73.0%	78.1%	CookHLA
	AFR	69.7%	39.9%	45.1%	34.4%	50.8%		82.0%	LSTM*HLA
		66.6%	35.4%	43.5%	38.2%	47.1%		77.0%	LSTM*OFA
		52.8%	34.1%	33.3%	35.0%	36.8%		69.1%	DeepHLA
		70.6%	--	44.0%	--	49.0%		84.3%	CookHLA
	AMR	81.5%	65.3%	66.7%	63.6%	69.4%	78.8%		LSTM*HLA
		77.9%	61.6%	64.3%	59.1%	67.4%	76.0%*		LSTM*OFA
		72.8%	58.7%	58.7%	55.5%	48.3%	68.8%		DeepHLA
		80.9%	--	68.2%	--	68.9%	78.3%		CookHLA

Abbreviation: EU European, EA East Asian, SA South Asian, AF African, AA Admixed American, PA Pan-Asian, LSTM*OFA in Method is LSTM One-Fit-All model. Asterisks (*) in LSTM*OFA model represents we apply the same group of parameters which maximize the overall accuracy to other pairs with the identical reference panel. Yellow shading represents the method with best accuracy in that pair. Orange shading represents the 6 matched pairs.

Table 2.1 shows the detailed results for the comparison of four methods. LSTM*HLA approach holds the highest accuracy in 28 out of 40 pairs, while CookHLA obtains the best

accuracy in 8 pairs, Deep*HLA receives 3 pairs winning, and LSTM*OFA only wins one pair. The performance of LSTM*OFA is typically normal, neither excellent nor unacceptable since it ranked second or third in 30 out of 33 pairs (excluding the 7 pairs that were utilized for experiments). The overall average accuracies for those 40 pairs are 76.0% for LSTM*HLA, 71.7% for LSTM*OFA, 69.4% for Deep*HLA and 74.5% for CookHLA, respectively. Overall, the performance of LSTM*HLA is marginally better than CookHLA and LSTM*OFA models but much better than Deep*HLA. Among 40 reference-target pairs, the reference and target panels have the same or quite similar ethnicity in 6 pairs (matched pairs) while have the quite different ethnicities in 34 pairs (unmatched paires). For the 6 matched pairs (2 for KOR vs EAS pairs, 2 for KOR vs Pan-Asian pairs and 2 for Pan-Asian vs SAS pairs), the average imputation accuracies are 84.0%, 82.0%, 81.2%, and 80.0% for LSTM*HLA, LSTM*OFA, Deep*HLA and CookHLA, respectively. However, this may generate biases for CookHLA and LSTM*OFA because CookHLA package merely successfully predict in two pairs (EAS as reference vs KOR as target and SAS as reference vs Pan-Asian as target) and after excluding two pairs of experiments, only four pairs are left for LSTM*OFA. Even so, we could easily distinguish that the accuracies for different methods are consistently stable in matched pairs. The two methods that are unbiased for contrasting are LSTM*HLA and Deep*HLA. Although LSTM*HLA reduces the error rate by 2.8% from 18.8% in Deep*HLA to 16.0% in LSTM*HLA, we could not draw the conclusion that one method outperforms the other methods due to the small sample size and fluctuation. Nonetheless, for those 34 unmatched pairs, the average imputation accuracy is 68.0% for LSTM*HLA and 57.7% for Deep*HLA. Due to the unsuccessful imputation with CookHLA when the reference panel is KOR and Pan-Asian, the accuracy of prediction is 68.9% based on 27 unmatched pairs while that of LSTM*OFA is 61.5% which was established on 29 pairs. As expected, the accuracies of unmatched pairs are much lower than the matched pairs since the same ethnic groups between reference and target panel are generally necessary for the HLA allele imputation [16, 30]. CookHLA has a slightly higher accuracy than LSTM*HLA (68.9% vs 68.0%) but both CookHLA and LSTM*HLA have much higher accuracies than the other two methods, LSTM*OFA and Deep*HLA. Therefore, LSTM*HLA, a machine learning method, and

CookHLA, a conventional approach has similar accuracy and outperform other existing models for the HLA allele imputation.

Imputation accuracies on low-frequency alleles. The research for low-frequent alleles carries profound and lasting impacts on clinical and genetical studies [8, 36]. Each single HLA locus may harbor thousands of possible alleles by IPD-IMGT/HLA dataset [37]. Some alleles with low frequencies are confirmed to be associated with certain types of diseases or adverse drug reactions. For example, HLA-DRB1*01:03 with a frequency of 0.6% is related to Ulcerative colitis [8], HLA-B*15:02 with a frequency of 0.3% is related to the response of carbamazepine [38], HLA-C*12:02 with a frequency of 1.1% is related to late-onset psoriasis [39], HLA-DRB1*08:01 with a frequency of 2.3% is related to fundamental biliary cirrhosis [40], HLA-B*57:01 with a frequency of 1.7% is related to the response of abacavir [41], HLA-B*58:01 with a frequency of 2.1% is related to the response of allopurinol [42]. All the above frequencies are evaluated by the T1DGC panel. Therefore, the study of HLA alleles with low frequencies is beneficial and necessary.

We classified the possible alleles for every HLA gene into six bins based on their frequencies. The six bins were two low frequency bins (0.1% ~ 0.5%, 0.5% ~ 1%) and four common alleles bins (1% ~ 5%, 5% ~ 10%, 10% ~ 20% and ≥ 20%). We employed accuracy and sensitivity to compare the performance for different methods. Sensitivity is defined as Formula (8) where D_i represents the imputed dosage of any allele for an individual i, $A1_{i,L}$ and $A2_{i,L}$ represent the correct alleles at locus L for individual i and n represents the number of individual samples. A pair of AFR as reference and AMR as target is appointed to explore the performance of accuracy and sensitivity for LSTM*HLA, Deep*HLA, and CookHLA.

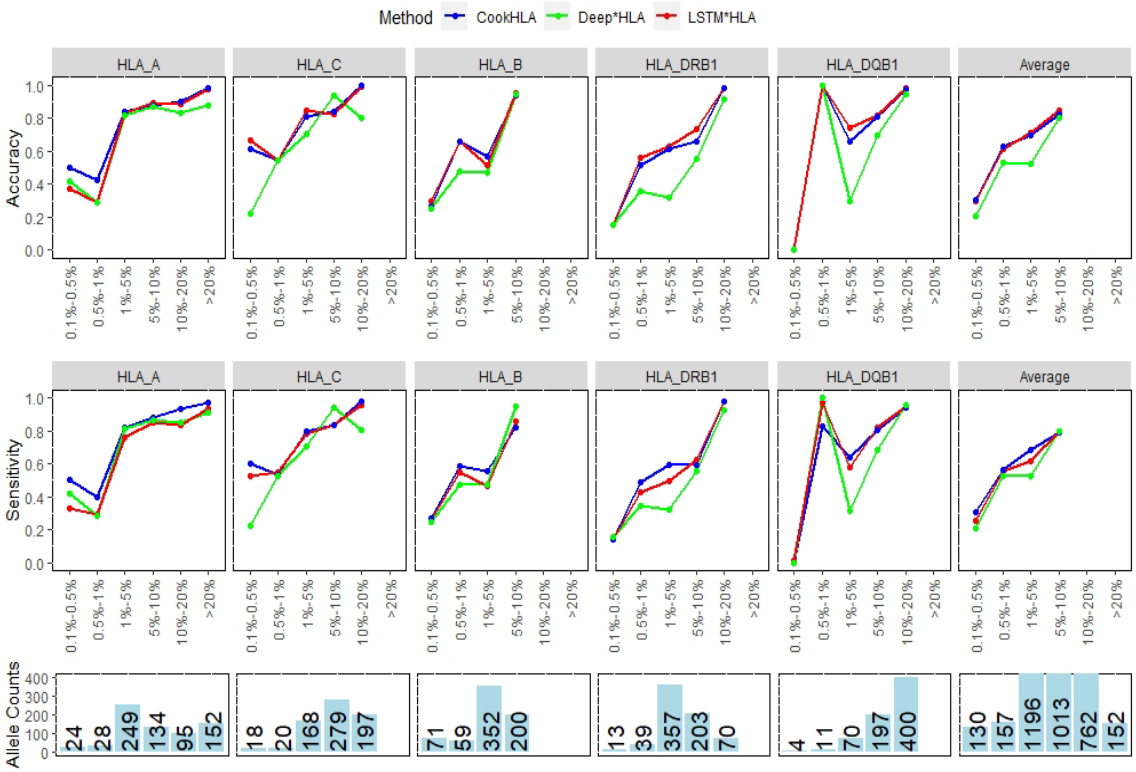


Figure 2.3 Imputation accuracies and sensitivities for six allele frequencies bin.

The pair of AFR (reference, N=661) and AMR (target, N=347) is utilized in this comparison. Accuracy is the number of corrected imputed alleles over the total number of alleles. Sensitivity is defined by Method Formula (8).

Table 2.2 Accuracy and Sensitivity of low-frequent alleles (≤ 1%) for each HLA gene.

Method		HLA-A	HLA-C	HLA-B	HLA-DRB1	HLA-DQB1	Average
CookHLA	**Acc.**	46.2%	57.9%	44.6%	42.3%	73.3%	52.9%
	Sen.	44.6%	56.7%	41.5%	40.2%	60.6%	48.7%
Deep*HLA	**Acc.**	34.6%	39.5%	35.4%	30.8%	73.3%	42.7%
	Sen.	34.7%	38.3%	35.0%	29.7%	73.2%	42.2%
LSTM*HLA	**Acc.**	32.7%	60.5%	46.2%	46.2%	73.3%	51.8%
	Sen.	30.9%	53.7%	38.9%	35.6%	71.4%	46.1%

Abbr.: Acc. Accuracy, Sen. Sensitivity. Note: The percentage in each cell is combination of the first two bins (frequency 0.1% ~ 0.5% and 0.5% ~ 1%).

In Figure 2.3, our method LSTM*HLA and CookHLA perform similarly and outperform Deep*HLA. For the common HLA alleles (frequency＞1%), LSTM*HLA has the highest accuracy in three HLA loci, HLA-C, HLA-DRB1, and HLA-DQB1. This is encouraging since the imputation methods generally do not perform well for HLA-DRB1 and HLA-DQB1. CookHLA achieves a slightly higher accuracy in HLA-A and HLA-B. For the low-frequent alleles (frequency ≤ 1%), Table 2.2 shows that our approach, LSTM*HLA obtains a higher accuracy in HLA-C, HLA-B and HLA-DRB1 with an error reduction of 2.6%, 1.6% and 3.9%, respectively, compared to CookHLA. LSTM*HLA and CookHLA have the same accuracy of 73.3% in HLA-DQB1 but CookHLA achieves a much better accuracy of 46.2% in HLA-A compared to 32.7% for LSTM*HLA. This results in that the average accuracy for low-frequent HLA alleles of LSTM*HLA is less than that of CookHLA (51.8% vs 52.9%). In terms of sensitivity, CookHLA outperforms LSTM*HLA and Deep*HLA in all five HLA loci. For low-frequent HLA alleles, CookHLA still achieves a higher sensitivity in four out of five loci, HLA-A, HLA-C, HLA-B and HLA-DRB1, especially for HLA-A. CookHLA has the sensitivity of 44.6% with decreasing the error

rate of almost 10% with Deep*HLA and 13.5% with LSTM*HLA for HLA-A. Deep*HLA has the best sensitivity for low-frequent alleles at HLA-DQB1 with a sensitivity of 73.2%. However, CookHLA does not perform well at this locus with a sensitivity of 60.6%. In average, CookHLA ranks first and has an average sensitivity of 48.7%, compared to LSTM*HLA with an average sensitivity of 46.1% and Deep*HLA with an average sensitivity of 42.2%. It is reasonable that the accuracy of LSTM*HLA outperforms CookHLA in terms of sensitivity since the randomly selected candidates for each parameter in LSTM*HLA would take the unexpected dosage for the possible alleles into account. When making the final consensus call, the average dosage from LSTM*HLA is less than the one with appropriate parameter settings.

Computational time for different models. We recorded the computational time for three methods, CookHLA, Deep*HLA and our LSTM*HLA on a server with Intel i5 3.2Ghz CPU with one core and 8 GB memory. Moreover, to discover the relationship between imputation accuracy and the number of parameters sets along with their computation costs, we measured two pairs for this attempt, one is the EAS (N=504) dataset as the reference panel and the SAS (N=489) dataset as the target panel.

Table 2.3 shows the comparison of computational time. CookHLA with Beagle v5 engine is efficient in the training procedure which is to learn adaptive genetic map by Hidden Markov Model with around 24 mins for those two conditions. However, the deep learning methods, Deep*HLA and LSTM*HLA have a much less computational cost in the imputation procedure. LSTM*HLA consumes 1.5 hours (blue shading) for the pair of EAS and SAS compared to less than 30 minutes for CookHLA and Deep*HLA. Moreover, we investigated the relation between the number of parameter settings, accuracy, and computational time. The number of SNPs per group significantly affects the imputation results, thus we prepared two sets of hyperparameters at first, and only one for the other two parameters. For the pair with couple hundreds of sample size in reference (EAS and SAS), LSTM*HLA takes only one and a half minutes in the training and imputation procedures with the accuracy of 85.1% which is higher than CookHLA with accuracy of 85.0%. As we increased the number of sets of hyperparameters, the CPU time increases

dramatically (from 1 minute to 4.8 hours), whereas the accuracy boosts slowly (from 85.1% to 86.8%). For another pair, the reference T1DGC panel has a much larger sample size (N =5225) than the other data sets, CookHLA consumes 48.5 minutes for the whole procedure while deep learning methods spend more than 5 hours on the training processing. As we further increased the number of sets of hyperparameters by increasing the number of each parameter, the CPU time in the training step for LSTM*HLA grows dramatically (from 31.6 minutes to 9.7 hours). Nonetheless, the accuracy first increases from 87.8% to 90.9% then decreases to 90.7%. Although the accuracy does not shrink greatly, this observation shows that the hyperparameters are critical to the performance of LSTM*HLA method and it directly impacts the imputation accuracy and computation time. Adding the set of number of sets of hyperparameters does not necessarily increase the imputation accuracy. It is noticed that the choice of hyperparameters is directly associated with the reference sample size. To balance the computation time and the imputation accuracy, we suggest using the sets hyperparameters that include the low, medium, and high values for learning rate and the hidden dimension and several numbers for the number of SNPs per group.

Table 2.3 Computational time for different methods and distinctive parameter settings for LSTM*HLA.

Method	Step	EAS ref (N=504) SAS tar (N=489) # of Overlap SNPs = 5067	T1DGC ref (N=5225) EUR tar (N=503) # of overlap SNPs= 3279
CookHLA (Beagle v5)	**AGM**	**Time:** 23.8 mins	**Time:** 24.4 mins
	Imp.	**Time:** 21.7 mins	**Time:** 24.1 mins
Deep*HLA	**MT**	**Time:** 28 mins	**Time:** 5.5 hours
	Imp.	**Time:** 51 secs	**Time:** 35 secs
LSTM*HLA	**MT**	L = {0.05}; S = {80, 120}; H = {30} **Time:** 1.4 min	L = {0.01}; S = {150, 200}; H= {80} **Time:** 31.6 mins
	Imp.	**Time:** 23 sec **Acc.:** 85.1%	**Time:** 19 secs **Acc.:** 87.8%
LSTM*HLA	**MT**	L = {0.05, 0.1}; S = {50, 80, 120}; H = {30, 100} **Time:** 13.4 mins	L = {0.01, 0.02}; S = {150, 180, 200}; H = {50, 100} **Time:** 2.7 hours
	Imp.	**Time:** 31 secs **Acc.:** 86.1%	**Time:** 30 secs **Acc.:** 89.6%
LSTM*HLA	**MT**	L = {0.05, 0.1, 0.2}; S = {20, 50, 80, 120}; H = {30, 50, 100} **Time:** 1.5 hours	L = {0.01, 0.02}; S = {150, 180, 190, 200}; H = {50, 80, 100} **Time:** 5.2 hours
	Imp.	**Time:** 55 secs **Acc.:** 86.7%	**Time:** 42 secs **Acc.:** 90.9%
LSTM*HLA	**MT**	L = {0.05, 0.07, 0.1, 0.15, 0.2}; S = {20, 50, 80, 100, 120, 150, 200}; H = {10, 30, 50, 80, 100} **Time:** 4.8 hours	L = {0.01, 0.02, 0.03} S = {150, 160, 180,190,200} H = {50, 80, 100} **Time:** 9.7 hours
	Imp.	**Time:** 2.8 mins **Acc.:** 86.8%	**Time:** 1.1 mins **Acc.:** 90.7%

Abbr.: AGM Adaptive Genetic Map, Imp. Imputation, MT Model Training, Acc. Accuracy. L Learning Rate, S Number of SNPs per group, H Hidden Dimension. The blue shading is the parameter setting we used in our model.

2.4 Discussion

We developed an advanced deep learning method, LSTM*HLA, based on the Long Short-Term Memory Machine Learning Method for the HLA imputation. To assess the performance of this methodology, we conducted comparisons with other popular approaches with seven real datasets. The results show that our method improves the imputation accuracy when the reference and target panels are from the same/different ethnic groups, especially for low-frequent alleles. Our approach has the following three highlights. We first employed the Bi-directional Long Short-Term Memory engine which is widely used in sequential predictions such as Natural Language Processing (NLP) and Bi directions is especially powerful for forecasting some specific position whose information are from previous and following materials. Secondly, we grouped the SNPs around the HLA loci and used them as the input cells in LSTM to reduce the noise in the data and to boost efficiency. For example, those SNPs would be divided into 20, 50, 80, 120 SNPs per group then fed into the LSTM. Finally, at the imputation procedure, we attained the allele for each haplotype HLA position with the maximized probability after averaging the outcomes for a set of hyperparameters since the output is sensitive to those hyperparameters. Such processing of a consensus call could greatly increase the robustness of our method.

Our method enhances the imputation accuracy in a couple of visible features. On the one hand, for the matched reference-target pairs, although those current approaches achieve high accuracies, our LSTM*HLA model has a higher accuracy. For example, for the reference-target pair of using the SAS dataset as the reference panel and the Pan-Asian dataset as the target panel, LSTM*HLA achieves the highest accuracy of 78.7% while the second highest accuracy is only 74.0% from LSTM*OFA method. On the other hand, the existing methods perform ordinarily unmatched reference-target pairs However, LSTM*HLA still improves the prediction accuracy in this situation. For example, for the pair of using the AFR dataset as the reference panel and AMR dataset as the target panel,

CookHLA already attains a fairly high accuracy of 78.3%, our method still has a higher accuracy of 78.8%. Last but not least, our method boosts the accuracy of low frequent HLA alleles. The study of low alleles plays a critical role in clinical research because many low-frequent alleles are associated with some diseases [8, 38-42]. Those low-frequent alleles are generally challenging to predict due to their weak LD and insufficient information in the reference panel. Our approach employs a deep learning algorithm to effectively apprehend the LD structure by the essence of network. For example, when the AFR dataset was used as the reference panel and the AMR dataset was used as the target panel, our LSTM*HLA achieves a higher accuracy for low-frequent (frequency $\leq$ 1%) alleles of HLA-C, HLA-B, HLA-DRB1 compared to CookHLA. Thus, this machine learning method may attain a far-reaching application in the low-frequent allele prediction.

The approaches we applied to compare were CookHLA and Deep*HLA. CookHLA is a traditional method that employs Beagle version 5 as its engine with embedding markers in exons and learning adaptive genetic maps as its features [17]. This approach accomplishes high accuracies in both matched and unmatched pairs. However, the computational time in the imputation procedure is high due to reusing reference haplotype information. Deep*HLA is a modern deep learning method that utilizes multi-task CNN algorithms as its primary scheme. Deep*HLA is outstanding when the reference and target panels are from the same/similar ethnic group. In our comparisons, Deep*HLA performs best when the KOR dataset is used as the target panel and the EAS dataset, or the Pan-Asian dataset was used as the reference panel.

Even though LSTM*HLA is an excellent approach that may obtain a far-reaching influence on the HLA allele imputation, some considerations still exist when analyzing real data. One potential concern is the quality of the phasing reference panel. The panels that were used in our projects might remain phasing errors. Nonetheless, this should not be overstated since all the methods compared rely on the phased haplotypes and the pre-phasing has been proven to be an efficient way for the SNP and HLA allele imputation. The other concern of LSTM*HLA is related to the hyperparameter setting. The imputation accuracy of LSTM*HLA heavily the hyperparameters used in the model. Therefore, we proposed to use multiple sets of hyperparameters that include the low, medium and high values for each hyperparameter and average the results from each set of hyperparameter. Out results show that such approach works well. However, the relationship between the

hyperparameters and the reference sample size remains ambiguous especially when large reference panels are used. In the future, we plan to develop a better procedure to determine the hyperparameters used in the model. Nevertheless, based on the results from our study, we expect that LSTM*HLA approach may possess a wide range of applications for the HLA imputation thus can be used in many studies.

2.5 Reference

1. Shankarkumar, U., *The Human Leukocyte Antigen (HLA) System.* International Journal of Human Genetics, 2004. **4**(2): p. 91-103.
2. Matzaraki, V., et al., *The MHC locus and genetic susceptibility to autoimmune and infectious diseases.* Genome Biology, 2017. **18**(1): p. 76.
3. Yamamoto, F., et al., *Capturing Differential Allele-Level Expression and Genotypes of All Classical HLA Loci and Haplotypes by a New Capture RNA-Seq Method.* Frontiers in Immunology, 2020. **11**.
4. Dilthey, A.T., et al., *HLA* IMP—an integrated framework for imputing classical HLA alleles from SNP genotypes.* Bioinformatics, 2011. **27**(7): p. 968-972.
5. Jia, X., et al., *Imputing amino acid polymorphisms in human leukocyte antigens.* PloS one, 2013. **8**(6): p. e64683.
6. Zheng, X., et al., *HIBAG—HLA genotype imputation with attribute bagging.* The pharmacogenomics journal, 2014. **14**(2): p. 192-200.
7. Zheng, X., *Imputation-based HLA typing with SNPs in GWAS studies.* HLA typing: methods and protocols, 2018: p. 163-176.
8. Goyette, P., et al., *High-density mapping of the MHC identifies a shared role for HLA-DRB1* 01: 03 in inflammatory bowel diseases and heterozygous advantage in ulcerative colitis.* Nature genetics, 2015. **47**(2): p. 172-179.
9. Hu, X., et al., *Additive and interaction effects at three amino acid positions in HLA-DQ and HLA-DR molecules drive type 1 diabetes risk.* Nature genetics, 2015. **47**(8): p. 898-905.

10. Raychaudhuri, S., et al., *Five amino acids in three HLA proteins explain most of the association between MHC and seropositive rheumatoid arthritis.* Nature genetics, 2012. **44**(3): p. 291-296.
11. Dunstan, S.J., et al., *Variation at HLA-DRB1 is associated with resistance to enteric fever.* Nature genetics, 2014. **46**(12): p. 1333-1336.
12. Study, I.H.C., *The major genetic determinants of HIV-1 control affect HLA class I peptide presentation.* Science, 2010. **330**(6010): p. 1551-1557.
13. Hirata, J., et al., *Genetic and phenotypic landscape of the major histocompatibilty complex region in the Japanese population.* Nature genetics, 2019. **51**(3): p. 470-480.
14. Okada, Y., et al., *Fine mapping major histocompatibility complex associations in psoriasis and its clinical subtypes.* The American Journal of Human Genetics, 2014. **95**(2): p. 162-172.
15. Kim, K., et al., *The HLA-DRβ1 amino acid positions 11–13–26 explain the majority of SLE–MHC associations.* Nature communications, 2014. **5**(1): p. 5902.
16. Cook, S., et al., *Accurate imputation of human leukocyte antigens with CookHLA.* Nature Communications, 2021. **12**(1): p. 1264.
17. Naito, T. and Y. Okada. *HLA imputation and its application to genetic and molecular fine-mapping of the MHC region in autoimmune diseases*. in *Seminars in Immunopathology*. 2022. Springer.
18. Yao, L. and Y. Guan. *An Improved LSTM Structure for Natural Language Processing*. in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*. 2018.
19. Consortium, G.P., *A global reference for human genetic variation.* Nature, 2015. **526**(7571): p. 68.
20. Abi-Rached, L., et al., *Immune diversity sheds light on missing variation in worldwide genetic diversity panels.* PloS one, 2018. **13**(10): p. e0206512.
21. Kim, K., et al., *Construction and application of a Korean reference panel for imputing classical alleles and amino acids of human leukocyte antigen genes.* PloS one, 2014. **9**(11): p. e112546.

22. Okada, Y., et al., *Risk for ACPA-positive rheumatoid arthritis is driven by shared HLA amino acid polymorphisms in Asian and European populations.* Human molecular genetics, 2014. **23**(25): p. 6916-6926.
23. Pillai, N.E., et al., *Predicting HLA alleles from high-resolution SNP data in three Southeast Asian populations.* Human molecular genetics, 2014. **23**(16): p. 4443-4451.
24. Graves, A. and A. Graves, *Long short-term memory.* Supervised sequence labelling with recurrent neural networks, 2012: p. 37-45.
25. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural computation, 1997. **9**(8): p. 1735-1780.
26. Greff, K., et al., *LSTM: A search space odyssey.* IEEE transactions on neural networks and learning systems, 2016. **28**(10): p. 2222-2232.
27. Huang, Z., W. Xu, and K. Yu, *Bidirectional LSTM-CRF models for sequence tagging.* arXiv preprint arXiv:1508.01991, 2015.
28. Ma, X. and E. Hovy, *End-to-end sequence labeling via bi-directional lstm-cnns-crf.* arXiv preprint arXiv:1603.01354, 2016.
29. Hameed, Z., B. Garcia-Zapirain, and I.O. Ruiz. *A computationally efficient BiLSTM based approach for the binary sentiment classification*. in *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 2019. IEEE.
30. Naito, T., et al., *A deep learning method for HLA imputation and trans-ethnic MHC fine-mapping of type 1 diabetes.* Nature Communications, 2021. **12**(1): p. 1639.
31. Megiorni, F. and A. Pizzuti, *HLA-DQA1 and HLA-DQB1 in Celiac disease predisposition: practical implications of the HLA molecular typing.* Journal of biomedical science, 2012. **19**: p. 1-5.
32. Lincoln, M.R., et al., *Epistasis among HLA-DRB1, HLA-DQA1, and HLA-DQB1 loci determines multiple sclerosis susceptibility.* Proceedings of the National Academy of Sciences, 2009. **106**(18): p. 7542-7547.
33. Okada, Y., et al., *Construction of a population-specific HLA imputation reference panel and its application to Graves' disease risk in Japanese.* Nature genetics, 2015.

47(7): p. 798-802.

34. Brown, W., et al., *Overview of the MHC fine mapping data*. 2009, Wiley Online Library. p. 2-7.
35. Zhou, F., et al., *Deep sequencing of the MHC region in the Chinese population contributes to studies of complex disease.* Nature genetics, 2016. **48**(7): p. 740-746.
36. Gourraud, P.-A., et al., *HLA diversity in the 1000 genomes dataset.* PloS one, 2014. **9**(7): p. e97282.
37. Robinson, J., et al., *Ipd-imgt/hla database.* Nucleic acids research, 2020. **48**(D1): p. D948-D955.
38. Tangamornsuksan, W., et al., *Relationship between the HLA-B* 1502 allele and carbamazepine-induced Stevens-Johnson syndrome and toxic epidermal necrolysis: a systematic review and meta-analysis.* JAMA dermatology, 2013. **149**(9): p. 1025-1032.
39. Mabuchi, T., et al., *HLA-C* 12: 02 is a susceptibility factor in late-onset type of psoriasis in Japanese.* The Journal of dermatology, 2014. **41**(8): p. 697-704.
40. Invernizzi, P., et al., *Classical HLA-DRB1 and DPB1 alleles account for HLA associations with primary biliary cirrhosis.* Genes & Immunity, 2012. **13**(6): p. 461-468.
41. Mallal, S., et al., *HLA-B* 5701 screening for hypersensitivity to abacavir.* New England Journal of Medicine, 2008. **358**(6): p. 568-579.
42. Hung, S.-I., et al., *HLA-B* 5801 allele as a genetic marker for severe cutaneous adverse reactions caused by allopurinol.* Proceedings of the National Academy of Sciences, 2005. **102**(11): p. 4134-4139.

3 Chapter 3: Practical Consideration of Reference Sample Size and Reference Panel Choice in HLA Allele Imputation

Abstract

Human leukocyte antigen (HLA) allele imputation provides a cost-effective strategy to use available SNPs in existing genome wide association studies (GWAs) to find HLA alleles that are associated with complex human immune diseases. The success of such a strategy depends on the imputation accuracy which is affected by several factors including the sample size and choice of reference panels. In this project, we investigated how two factors, the reference sample size and the reference panel selection affect the imputation accuracy using several datasets. Our results show that that more than 50 individuals are sufficient to achieve a high imputation accuracy when the reference and target panel are the same ethnic group. Our results also show that the matched reference and target panels are desired while the use of a reference panel that has samples from different ethnic groups does not necessarily improve the imputation accuracy.

3.1 Introduction

Human Leukocyte Antigen (HLA), located on chromosome 6, plays a crucial role in the human immune system. HLA is a complicated gene with the nature of highly polymorphic that over 25,000 known alleles exist. Such high polymorphism makes it challenging for

genotyping and association analysis since it requires hundreds even thousands of samples that are genotyped at HLA loci. At the same time, genome-wide association studies (GWAS) have been very successful in locating genetic variants that are associated with human complex diseases. In GWAS, genotypes at millions of SNPs including SNPs surrounding HLA loci are available. It has been proposed that one can use genotypes of SNPs surrounding HLA loci to impute HLA alleles then use imputed HLA alleles in association analysis of human complex diseases. The success of such a strategy depends on the availability of statistical methods and large reference panels in which genotypes at SNPs surrounding HLA loci and genotypes at HLA loci that can be used to impute HLA alleles with high accuracy. Fortunately, recent developments in statistical and computational methods for the HLA allele imputation [3, 7, 11, 12, 13, 14] and availability of large reference panels from multiple ethnic groups such as reference panels in the 1000 Genomes Project (https://www.internationalgenome.org/) have made such method become more popular. Although the procedure for the HLA allele imputation has not been extensively evaluated [1-6], it has been shown that such imputation method can improve power platforming association analysis of HLA genes [3, 7, 8].

The imputation accuracy is affected by several factors including the statistical methods used, the sample size of reference panels, and the choice of reference panels since reference panels from multiple ethnic groups are available. Many methods have been developed for the HLA allele imputation with high accuracy, such as conventional methods e.g., Beagle [9], SNP2HLA [10], HIBAG [11], CookHLA [12] and deep learning methods e.g., Deep*HLA [13, 14]. Although some have been done to compare the performance of methods developed [12-15], it is still lack of systematic evaluation how other factors such as the sample size and the choice of reference panels affect the imputation accuracy.

In this project, we investigated how two factors, the sample size and the choice of reference panels, can influence the quality of HLA allele imputation in terms of the accuracy and efficacy described in [15-18]. Those studies have shown that the imputation accuracy can be dramatically improved when the reference and target panels are from the same ethnic group [11-13]. However, internal panels that are fully matched with target

samples in terms of ethnicity can be difficult to obtain due to the high cost to genotype HLA alleles for a large number of samples. Therefore, it is highly desired to study how to efficiently use samples from available reference panels to further improve the imputation accuracy for a given set of target samples. For the factor of reference sample size, it is important to recognize whether a portion of a reference panel is sufficient to substitute a larger or the original reference panel.

The methods used in evaluations are CookHLA [12] and Deep*HLA [13], two recently developed methods for the HLA allele imputation. As the representative of conventional methods, CookHLA achieves higher imputation accuracy than SNP2HLA and HIBAG by employing Beagle version 5 as its engine, embedding local markers, and learning adaptively genetic maps [12]. Deep*HLA is based on a deep learning method, the multitask Convolutional Neural Network (CNN) that encompasses a shared segment of two conventional layers and a fully connected layer. By the nature of deep learning methods, Deep*HLA attains high accuracy, especially when the reference and target panels are from the same ethnic group. [13].

3.2 Materials and Methods

Data. As a continuation from Chapter 2, we still used the same seven real datasets: five of them are obtained from 1000 Genomes Project [1], one is a KOR dataset [2] and the other is a Pan-Asian dataset [3-5]. The brief introduction including ethnicity and number of individuals of these seven datasets is as follows: African (AFR, N = 661), Admixed American (AMR, N = 347), European (EUR, N = 503), East Asian (EAS, N = 504), South Asian (SAS, N = 489), Korean from East Asian (KOR, N = 413), Pan-Asian from East and South Asian (PA, N = 530). The HLA region is from MHC part (chr6:25-36 Mb) and the HLA loci used in this project include HLA-A, HLA-C, HLA-B, HLA-DRB1, and HLA-DQB1.

Measures of imputation quality. We used accuracy and efficacy as the measures of imputation quality. Accuracy [6-8] was calculated as the ratio of the number of correctly imputed haplotypes over the total number of haplotypes. Efficacy [7] was defined as the proportion of the number of correctly imputed haplotypes with the probability greater than or equal to some threshold over the total number of haplotypes. A higher threshold results in lower efficacy along with higher reliability. A noticeably high threshold is essential for greater imputation reliability. We use the threshold of 0.90, 0.95 and 0.99 in this study.

Reference size. We designed two experiments to investigate the size of reference panels. In the first experiment, the reference and target panels are from the same ethnic group. For example, the EAS dataset was used as the reference panel and the KOR dataset was used as the target panel. In the second experiment, the reference and target panels are from different ethnic groups. For example, the EUR dataset was used as the reference panel while the AFR dataset was used as the target panel. We obtained the overlapping SNPs for both the reference and target panels, and randomly selected small, medium, and large groups of haplotypes from the reference panel then utilized those haplotypes as reference panel to train the model and to impute the unselected samples in the original reference panel itself. For Deep*HLA, 5, 10, 15, 20, 25, 30, 40, 50, 75, and 100 haplotypes randomly selected to form a small reference panel; 150, 200, 250, 300, 400, and 500 randomly selected to a medium reference panel; and 600, 700, 800, and 900 haplotypes randomly selected to form a large reference panel, respectively. For CookHLA, due to the constrain on the minimum sample size required for the reference panel, 30, 40, 50, 60, and 80 haplotypes randomly selected to form a small reference panel; 100, 150, 200, and 300 haplotypes randomly selected to form a medium reference panel; and 400, 500, 600, and 800 haplotypes were randomly selected to form a large reference panel, respectively.

Choice of reference panels. The choice of appropriate reference panels is essential to accurate HLA allele imputation. Here we define a reference panel as a matched reference panel if the reference panel and the target panel are from the same ethnic group while we define a reference panel as an unmatched reference panel if the reference panel and the target panel are not from the same ethnic group [9]. For an extreme case of unmatched

reference panels, the reference panel and the target panel can be from two completely distinct ethnic groups. In our datasets, four datasets are from Asian (EAS, SAS, KOR, and PA), two datasets (AMR and EUR) are from Africa, and one dataset (EUR) is from European. Due to the limited number of ethnic groups available we used the KOR dataset as the target panel and the EAS dataset as the reference panel and gradually added other datasets to the reference panels till the reference panel contains all the datasets except the KOR datasets. Specifically, the following groups of reference panels were used: the Asian Panel (EAS, EAS + SAS, EAS + PA, and EAS + PA + SAS), the non-Asian panel (AMR + AFR + EUR, and the cosmopolitan panel (EAS + PA + SAS+ AMR + AFR + EUR).

3.3 Results

The impact of reference sample size. Impact of reference sample size. To exam the impact of the reference sample size, we utilized the KOR dataset as the target panel and the EAS dataset as the matched reference panel to construct 20 sets of reference panels with different sample sizes for Deep*HLA and 13 sets of reference panels for CookHLA. For each sample size, we randomly selected the corresponding number of haplotypes from the EAS dataset and repeated such process 10 times. The average accuracy over 10 reference panels with the sample size was calculated for each HLA locus and then averaged across HLA loci. As shown in Figures 3.1.a and 3.1.b , the imputation accuracies for all HLA loci increase dramatically as the number of haplotypes increases from 0 to 100, but such increases are less when the number of haplotypes is more than 100 but less than 600 and are much less when the number of haplotypes is more than 600 for both Deep*HLA and CookHLA. The results obtained by CookHLA are more compact than those of Deep*HLA indicating that CookHLA is more robust than Deep*HLA. Such observations can be clearly illustrated by Figures 3.1.c and 3.1.d. The standard deviations of imputation accuracies over 10 replicates drops fast when the number of haplotypes changes from 0 to 200 then diminish slowly after that for both Deep*HLA and CookHLA. CookHLA has a

smaller variance than Deep*HLA in general, resulting in a more stable imputation. For different HLA loci, the numbers of haplotypes required to achieve a high imputation accuracy differ. For example, more than 400 haplotypes are needed for HLA-B to achieve a satisfactory accuracy for Deep*HLA, while only 100 haplotypes are needed for HLA-DQB1 (blue curve) to achieve a similar accuracy for both Deep*HLA and CookHLA.

Impact of choice of reference panels. The selection of appropriate reference panels plays a critical role in the HLA allele imputation. We utilized the KOR dataset as the target panel and the EAS dataset as the starting reference panel then gradually expanded the reference panel to include all Asian dataset, only the non-Asian datasets (EUR + AFR + AMR), and all six datasets. As shown in Table 3.1, for accuracy, both CookHLA and Deep*HLA methods achieve their highest accuracies of 90.2% and 84.8%, respectively, when the reference panel is EAS+SAS (East and South Asian combination) The reference panels that have the second highest accuracy are different according to the methods used in the imputation. Deep*HLA has an accuracy of 84.5% with EAS+SAS+PA (all Asian combination) while CookHLA has an accuracy of 90.0% with EAS. In addition, f CookHLA performs very well for the reference panel consisting of all Asian datasets and has a slightly lower accuracy of 88.8%. As it is expected, the sue of the non-Asian panel results in the lowest accuracy: CookHLA has an accuracy of 64.9% while Deep*HLA has an accuracy of 7.10%, respectively. For the cosmopolitan panel that contains six datasets, CookHLA has an accuracy of 85.3% while Deep*HLA has an accuracy of 81.1%. This may mainly be due to the additional noise introduced by non-Asian samples. It is worth noting that the Pan-Asian datasets includes the samples from the East and South Asian populations, thus the imputation based on the reference panel that combines the EAS dataset and the Pan-Asian dataset may be expected to have the highest accuracy. However, accuracy from the EAS+PA panel is slightly lower than those from the EAS panel and the EAS+SAS panel. A possible explanation is as follows. Note that the EAS dataset and the SAS dataset are both from the 1000 Genomes Project while the Pan-Asian data is from another project. There are 2,241 SNPs used in the EAS+SAS panel while only 1,734 SNPs were used in the EAS+PA panel. The accuracy deficiency may be due to the elimination of a large number of SNPs used in the imputation. For efficacy, we utilized three thresholds,

0.90, 0.95 and 0.99, to explore how it is affected by the choice of reference panels. In general, the efficacies are not as high as accuracies and the higher the threshold, the lower the efficacies. The decay in terms of accuracy for different efficacies is not obvious for Deep*HLA. For example, the efficacies are 82.3%, 82.3%, and 81.4% for the thresholds of 0.90, 0.95, and 0.99, respectively, when the Asian reference panel was used. However, noticeable differences in terms of efficacy can been for CookHLA: the efficacies are 71.5%, 65.5% and 55.3% for the thresholds of 0.90, 0.95, and 0.99 efficacy, respectively, when the Asian reference panel was used. The distinction patterns in the efficacies of the two methods is mainly due to the underlying mechanisms in deep learning methods. The SoftMax procedure in deep learning methods would amplify the posterior probability of the allele with the highest probability. Thus, for deep learning methods, one allele would have a much higher posterior probability that is close to 1 so the final imputation is unlikely to be affected by the threshold of efficacy. For the traditional method such as CookHLA, many alleles may have similar posterior probabilities.

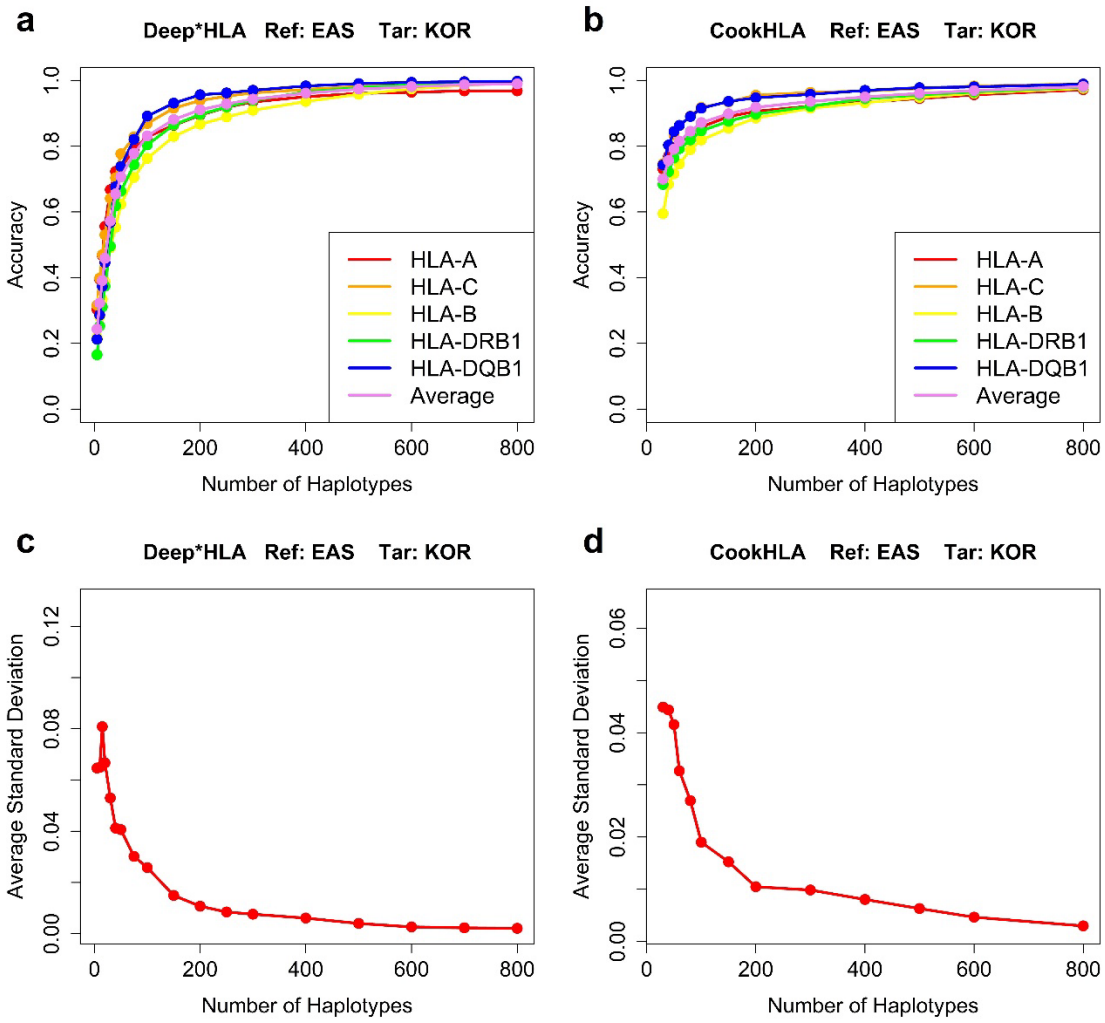


Figure 3.1 The impact of the reference sample size by using the EAS dataset as the reference panel and the KOR dataset as the target panel.

a. The performance of accuracy of Deep*HLA for each HLA locus. **b.** The performance of accuracy on CookHLA for each HLA locus. **c.** and **d.** The average standard deviation for Deep*HLA and CookHLA.

Table 3.1 The comparison of different reference panels.

Target Panel	Method	Reference Panel						
		Pop.	**EAS**	**EAS+ SAS**	**EAS+ PA**	**Asian**	**Non-Asian**	**Cos.**
		Pop. Size	498	980	906	1388	1492	2460
KOR	**Deep*HLA**	Accuracy	0.802	0.848	0.821	0.845	0.710	0.811
		Efficacy-90	0.785	0.843	0.808	0.823	0.704	0.797
		Efficacy-95	0.778	0.840	0.803	0.823	0.699	0.794
		Efficacy-99	0.768	0.836	0.794	0.814	0.693	0.782
	CookHLA	Accuracy	0.900	0.902	0.883	0.888	0.649	0.853
		Efficacy-90	0.778	0.776	0.703	0.715	0.411	0.603
		Efficacy-95	0.741	0.742	0.648	0.655	0.365	0.560
		Efficacy-99	0.638	0.650	0.546	0.553	0.294	0.458

Abbr.: Pop.: Population, PA: Pan-Asian, Asian: EAS+PA+SAS, Non-Asian: EUR+AFR+AMR, Cos.: Cosmopolitan, EAS+SAS+Pan_Asian+EUR+AFR+AMR. Efficacy-90, 95, 99: The thresholds of Efficacy are 0.90, 0.95 and 0.99 respectively.

3.4 Discussion

In this project, we investigated two factors that may influence the accuracy of HLA allele imputation using seven real datasets and two recently developed methods for the HLA allele imputation methods: Deep*HLA and CookHLA which are representative of popular machine learning approaches and conventional methods. We evaluated the imputation accuracies of different reference sample sizes when the EAS dataset was used as the reference panel and the KOR dataset was used as the target panel. Moreover, we examined the accuracy and efficacy of different reference panels when the KOR dataset was used as the target panel.

It is important to know how large a reference panel can guarantee the accurate imputation. It is not doubtful that a larger reference sample size is always better, but a certain sample size must be used in real studies due to the limited resources and the high expense for HLA gneotyping. Our results from Figure 3.1 showed that steep slopes generally appear on the curves when the number of haplotypes is less than 100, or the number of individuals is less than 50, or both Deep*HLA and CookHLA. The increases gradually slowdown from when the number of haplotypes increases from 100 to 600while there is no noticeable gains in terms of accuracy when more than 600 haplotypes are used. For a specific HLA locus, the numbers of haplotypes required to achieve a high accuracy differ. For example, for Deep*HLA, 200 reference haplotype is sufficient for imputing HLA-DQB1. However, to impute HLA-B, it may request at least 600 haplotypes. Therefore, we advocate to use a reference panel that contains at least 200 haplotypes or 100 individuals for accurate imputation of HLA alleles.

The choice of appropriate reference panels is another critical factor that needs to be carefully considered. Based on the findings from Table 3.1, a matched reference panel is essential. When the KOR dataset was used as the target panel, the accuracy from the EAS+SAS reference panel is the highest the use of the Asian panel results in high imputation accuracy too. It is expected that the use of the non-Asian reference panel has the lowest accuracy. Surprisingly, the use of the cosmopolitan panel does not improve the imputation accuracy. In terms of efficacy, as we increased the threshold from 0.90 to 0.99, the efficacy of Deep*HLA does not drop fiercely compared to CookHLA.

There are some limitations of our current study. In this study, we only utilized the KOR dataset as the target panel which lead to some biases. In future, other datasets will be used as target panels. In this study, we considered two critical factors which would affect imputation accuracies. However, some other factors, such as the window size, the target sample size, and the set of SNPs around HLA loci, would also influence the accuracy for the HLA allele imputation. Moreover, different combinations of reference panels may result in different sets of SNPs used in the imputation, leading to unexpected imputation results. In the future, we will conduct comprehensive evaluations.

3.5 Reference

1. Anderson, C.A., et al., *Evaluating the effects of imputation on the power, coverage, and cost efficiency of genome-wide SNP platforms.* The American Journal of Human Genetics, 2008. **83**(1): p. 112-119.
2. Li, Y., et al., *Genotype imputation.* Annual review of genomics and human genetics, 2009. **10**: p. 387-406.
3. Marchini, J. and B. Howie, *Genotype imputation for genome-wide association studies.* Nature Reviews Genetics, 2010. **11**(7): p. 499-511.
4. Servin, B. and M. Stephens, *Imputation-based analysis of association studies: candidate regions and quantitative traits.* PLoS genetics, 2007. **3**(7): p. e114.
5. Spencer, C.C., et al., *Designing genome-wide association studies: sample size, power, imputation, and the choice of genotyping chip.* PLoS genetics, 2009. **5**(5): p. e1000477.
6. The Wellcome Trust Case Control Consortium, *Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls.* Nature, 2007. **447**(7145): p. 661-678.
7. Hao, K., et al., *Accuracy of genome-wide imputation of untyped markers and impacts on statistical power for association studies.* BMC genetics, 2009. **10**: p. 1-10.
8. Zeggini, E., et al., *Meta-analysis of genome-wide association data and large-scale replication identifies additional susceptibility loci for type 2 diabetes.* Nature genetics, 2008. **40**(5): p. 638-645.
9. Browning, S.R. and B.L. Browning, *Rapid and Accurate Haplotype Phasing and Missing-Data Inference for Whole-Genome Association Studies By Use of Localized Haplotype Clustering.* The American Journal of Human Genetics, 2007. **81**(5): p. 1084-1097.
10. Jia, X., et al., *Imputing amino acid polymorphisms in human leukocyte antigens.*

PloS one, 2013. **8**(6): p. e64683.

11. Zheng, X., et al., *HIBAG—HLA genotype imputation with attribute bagging.* The pharmacogenomics journal, 2014. **14**(2): p. 192-200.
12. Cook, S., et al., *Accurate imputation of human leukocyte antigens with CookHLA.* Nature Communications, 2021. **12**(1): p. 1264.
13. Naito, T., et al., *A deep learning method for HLA imputation and trans-ethnic MHC fine-mapping of type 1 diabetes.* Nature communications, 2021. **12**(1): p. 1639.
14. Naito, T. and Y. Okada. *HLA imputation and its application to genetic and molecular fine-mapping of the MHC region in autoimmune diseases*. in *Seminars in Immunopathology*. 2022. Springer.
15. Zhang, B., et al., *Practical consideration of genotype imputation: sample size, window size, reference choice, and untyped rate.* Statistics and its interface, 2011. **4**(3): p. 339.
16. Huang, L., et al., *Genotype-imputation accuracy across worldwide human populations.* The American Journal of Human Genetics, 2009. **84**(2): p. 235-250.
17. Huang, L., C. Wang, and N.A. Rosenberg, *The relationship between imputation error and statistical power in genetic association studies in diverse populations.* The American Journal of Human Genetics, 2009. **85**(5): p. 692-698.
18. Pei, Y.-F., et al., *Analyses and comparison of accuracy of different genotype imputation methods.* PloS one, 2008. **3**(10): p. e3551.
19. Consortium, G.P., *A global reference for human genetic variation.* Nature, 2015. **526**(7571): p. 68.
20. Kim, K., et al., *Construction and application of a Korean reference panel for imputing classical alleles and amino acids of human leukocyte antigen genes.* PloS one, 2014. **9**(11): p. e112546.
21. Okada, Y., et al., *Risk for ACPA-positive rheumatoid arthritis is driven by shared HLA amino acid polymorphisms in Asian and European populations.* Human molecular genetics, 2014. **23**(25): p. 6916-6926.
22. Pillai, N.E., et al., *Predicting HLA alleles from high-resolution SNP data in three Southeast Asian populations.* Human molecular genetics, 2014. **23**(16): p. 4443-

4451.

23. Nothnagel, M., et al., *A comprehensive evaluation of SNP genotype imputation.* Human genetics, 2009. **125**: p. 163-171.

24. Zhao, H., R. Pfeiffer, and M.H. Gail, *Haplotype analysis in population genetics and association studies.* Pharmacogenomics, 2003. **4**(2): p. 171-178.

A Appendix Chapter 1 Supplement

A.1 Tables

Table A.1.1 Pearson correlation coefficients between 80 features in our model and four sets of features in the model given in [8].

Original Corr. / 2&3-mer	Viral	Similarity to ISGs	Similarity to Housekeeping Genes	Similarity to Remaining Genes
AA	-0.21	0.19	0.23	0.17
AC	0.57	-0.52	-0.54	-0.54
AG	0.65	0.60	0.61	0.61
AT	0.53	-0.44	-0.40	-0.44
CA	0.58	0.29	0.31	0.34
CC	-0.04	-0.01	-0.01	-0.01
CG	0.80	-0.70	-0.62	-0.71
CT	0.55	0.53	0.54	0.53
GA	0.69	0.48	0.49	0.44
GC	0.37	0.27	0.29	0.27
GG	0.02	0.04	0.02	0.03
GT	0.72	-0.66	-0.67	-0.64
TA	0.69	-0.58	-0.61	-0.61
TC	0.67	0.13	0.06	0.09
TG	0.73	0.65	0.64	0.64
TT	0.18	0.14	0.13	0.13
AAA	0.82	-0.46	-0.40	-0.38
AAC	-0.09	0.21	0.25	0.25
AAG	-0.39	0.09	0.12	0.13
AAT	0.81	-0.32	-0.10	-0.22
ACA	0.75	-0.25	-0.16	-0.29
ACC	0.78	0.57	0.40	0.51
ACG	0.89	-0.86	-0.85	-0.83
ACT	0.57	-0.12	-0.08	-0.15
AGA	0.75	-0.65	-0.72	-0.72
AGC	0.80	0.65	0.48	0.61
AGG	0.54	-0.11	-0.33	-0.19
AGT	0.78	-0.18	-0.05	-0.22
ATA	0.80	-0.77	-0.77	-0.77

ATC	-0.04	0.39	0.42	0.40
ATG	0	0	0	0
ATT	0.55	0.09	0.25	0.11
CAA	0.67	-0.61	-0.63	-0.63
CAC	0.69	0.64	0.49	0.62
CAG	0.43	0.46	0.43	0.44
CAT	0.40	-0.01	0.13	0.00
CCA	-0.19	0.11	0.11	0.15
CCC	0.86	0.37	0.16	0.39
CCG	0.81	-0.74	-0.73	-0.73
CCT	-0.10	0.03	0.01	0.03
CGA	0.40	-0.15	0.13	-0.10
CGC	0.89	-0.77	-0.79	-0.74
CGG	0.82	0.32	0.30	0.29
CGT	0.43	-0.42	-0.37	-0.41
CTA	0.55	-0.46	-0.45	-0.47
CTC	0.83	0.27	0.10	0.26
CTG	0.52	0.66	0.64	0.64
CTT	0.69	-0.55	-0.46	-0.59
GAA	0.56	-0.06	0.07	0.02
GAC	0.80	0.46	0.06	0.38
GAG	0.50	0.31	0.21	0.22
GAT	0.76	-0.29	0.10	-0.21
GCA	0.07	-0.08	-0.08	-0.09
GCC	0.84	0.59	0.46	0.61
GCG	0.85	-0.73	-0.74	-0.73
GCT	0.02	-0.27	-0.29	-0.27
GGA	0.46	-0.41	-0.39	-0.41
GGC	0.83	0.22	0.14	0.20
GGG	0.70	0.18	0.03	0.25
GGT	0.19	-0.11	-0.13	-0.10
GTA	0.71	-0.70	-0.68	-0.70
GTC	0.77	-0.01	-0.24	-0.03
GTG	0.63	0.55	0.54	0.57
GTT	0.78	-0.74	-0.74	-0.75
TAA	0.25	-0.35	-0.34	-0.33
TAC	-0.28	-0.16	-0.10	-0.14
TAG	-0.03	-0.10	-0.10	-0.11
TAT	0.81	-0.44	-0.26	-0.42
TCA	0.64	-0.52	-0.49	-0.52
TCC	0.83	0.33	0.17	0.33
TCG	0.87	-0.79	-0.81	-0.82
TCT	0.58	-0.30	-0.24	-0.35
TGA	0.01	0.09	0.10	0.10

TGC	0.46	0.27	0.16	0.26
TGG	0	0	0	0
TGT	0.56	-0.25	-0.14	-0.23
TTA	0.90	-0.85	-0.86	-0.83
TTC	-0.01	0.16	0.17	0.17
TTG	0.72	-0.50	-0.47	-0.53
TTT	0.80	-0.49	-0.36	-0.51
2mer-Total	0.47	0.03	0.03	0.02
3mer-Total	0.54	-0.12	-0.12	-0.12

Table A.1.2 The comparison of different setting of max-run in Boruta for 80 models and 146*4+80 models

Model: 80, Boruta Maxrun = 100								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	39	0.292	0.291	0.292	12	79	91	80.5%
2nd	38	0.2915	0.291	0.292	9	80	89	78.8%
3rd	38	0.293	0.291	0.293	14	72	86	76.1%
4th	40	0.287	0.287	0.287	9	78	87	77.0%
5th	36	0.295	0.295	0.295	11	80	91	80.5%
6th	42	0.293	0.291	0.293	16	74	90	79.6%
7th	35	0.287	0.287	0.287	10	79	89	78.8%
8th	39	0.295	0.295	0.295	16	74	90	79.6%
9th	36	0.292	0.291	0.292	11	80	91	80.5%
10th	38	0.288	0.287	0.288	10	79	89	78.8%
Average	38.1	0.291	0.291	0.291	11.8	77.5	89.3	79.0%
Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)								

Model: 80, Boruta Maxrun = 200								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	45	0.290	0.291	0.290	13	77	90	79.6%
2nd	44	0.296	0.295	0.297	8	82	90	79.6%
3rd	44	0.296	0.295	0.297	13	79	92	81.4%
4th	43	0.292	0.291	0.292	8	82	90	79.6%
5th	42	0.293	0.291	0.293	15	77	92	81.4%
6th	45	0.300	0.299	0.300	15	75	90	79.6%
7th	40	0.281	0.287	0.287	9	78	87	77.0%
8th	45	0.292	0.291	0.292	13	77	90	79.6%
9th	46	0.293	0.291	0.293	12	79	91	80.5%
10th	45	0.292	0.291	0.292	13	77	90	79.6%
Average	43.9	0.293	0.292	0.293	11.9	78.3	90.2	79.8%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 80, Boruta Maxrun = 500								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	48	0.292	0.291	0.292	15	75	90	79.6%
2nd	49	0.296	0.295	0.297	14	76	90	79.6%

3rd	47	0.293	0.291	0.293	15	76	91	80.5%
4th	49	0.302	0.303	0.302	15	76	91	80.5%
5th	48	0.292	0.291	0.292	15	75	90	79.6%
6th	49	0.294	0.295	0.293	16	74	90	79.6%
7th	48	0.292	0.291	0.292	15	75	90	79.6%
8th	48	0.292	0.291	0.292	15	75	90	79.6%
9th	49	0.296	0.295	0.297	14	76	90	79.6%
10th	49	0.301	0.303	0.300	15	77	92	81.4%
Average	48.6	0.295	0.295	0.295	14.9	75.5	90.4	80.0%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 80, Boruta Maxrun = 1000								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	52	0.294	0.291	0.295	12	78	90	79.6%
2nd	53	0.294	0.291	0.295	14	76	90	79.6%
3rd	50	0.292	0.291	0.292	14	75	89	78.8%
4th	51	0.297	0.299	0.297	17	73	90	79.6%
5th	54	0.290	0.291	0.290	13	77	90	79.6%
6th	55	0.295	0.295	0.295	15	76	91	80.5%
7th	53	0.297	0.299	0.297	15	75	90	79.6%
8th	53	0.295	0.295	0.295	15	76	91	80.5%

9th	53	0.295	0.295	0.295	13	78	91	80.5%
10th	53	0.296	0.295	0.297	13	77	90	79.6%
Average	52.7	0.295	0.294	0.295	14.1	76.1	90.2	79.8%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 80, Boruta Maxrun = 2000								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	55	0.295	0.295	0.295	15	76	91	80.5%
2nd	55	0.295	0.295	0.295	15	76	91	80.5%
3rd	54	0.290	0.291	0.290	13	77	90	79.6%
4th	55	0.295	0.295	0.295	15	76	91	80.5%
5th	55	0.295	0.295	0.295	15	76	91	80.5%
6th	55	0.295	0.295	0.295	15	76	91	80.5%
7th	54	0.296	0.295	0.297	15	76	91	80.5%
8th	55	0.295	0.295	0.295	15	76	91	80.5%
9th	55	0.295	0.295	0.295	15	76	91	80.5%
10th	55	0.295	0.295	0.295	15	76	91	80.5%
Average	54.8	0.295	0.295	0.295	14.8	76.1	90.9	80.4%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 146*4+80, Boruta Maxrun = 100								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	20	0.287	0.287	0.287	11	57	68	60.2%
2nd	24	0.283	0.284	0.283	12	51	63	55.8%
3rd	23	0.294	0.291	0.295	21	52	73	64.6%
4th	28	0.285	0.287	0.283	18	53	71	62.8%
5th	20	0.302	0.303	0.302	13	48	61	54.0%
6th	29	0.287	0.287	0.287	21	58	79	69.9%
7th	25	0.292	0.291	0.292	16	50	66	58.4%
8th	30	0.296	0.295	0.297	19	56	75	66.4%
9th	28	0.290	0.291	0.290	18	51	69	61.1%
10th	23	0.290	0.291	0.290	19	55	74	65.5%
Average	25	0.291	0.291	0.291	16.8	53.1	69.9	61.9%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 146*4+80, Boruta Maxrun = 200								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	42	0.290	0.287	0.292	19	61	80	70.8%
2nd	44	0.281	0.284	0.280	26	53	79	69.9%
3rd	39	0.287	0.287	0.287	16	54	70	61.9%

4th	42	0.287	0.287	0.287	28	52	80	70.8%
5th	35	0.275	0.276	0.275	21	49	70	61.9%
6th	44	0.292	0.291	0.292	21	62	83	73.5%
7th	42	0.294	0.295	0.293	22	57	79	69.9%
8th	39	0.297	0.295	0.298	21	57	78	69.0%
9th	39	0.301	0.299	0.302	24	50	74	65.5%
10th	40	0.287	0.287	0.287	24	59	83	73.5%
Average	40.6	0.289	0.289	0.289	22.2	55.4	77.6	68.7%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 146*4+80, Boruta Maxrun = 500								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	52	0.283	0.280	0.285	21	63	84	74.3%
2nd	50	0.275	0.276	0.275	26	62	88	77.9%
3rd	49	0.292	0.291	0.292	22	60	82	72.6%
4th	51	0.289	0.291	0.288	19	61	80	70.8%
5th	55	0.281	0.280	0.282	27	64	91	80.5%
6th	50	0.288	0.287	0.288	21	68	89	78.8%
7th	51	0.285	0.284	0.285	31	56	87	77.0%
8th	42	0.295	0.295	0.295	18	57	75	66.4%
9th	47	0.288	0.287	0.288	19	65	84	74.3%

10th	57	0.289	0.287	0.290	25	64	89	78.8%
Average	50.4	0.287	0.286	0.287	22.9	62	84.9	75.1%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 146*4+80, Boruta Maxrun = 1000								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	57	0.281	0.280	0.282	21	71	92	81.4%
2nd	50	0.275	0.276	0.275	26	62	88	77.9%
3rd	53	0.286	0.287	0.285	25	58	83	73.5%
4th	52	0.288	0.291	0.287	20	58	78	69.0%
5th	59	0.275	0.276	0.275	24	65	89	78.8%
6th	50	0.288	0.287	0.288	21	68	89	78.8%
7th	53	0.289	0.291	0.288	28	58	86	76.1%
8th	46	0.304	0.303	0.305	21	57	78	69.0%
9th	48	0.292	0.291	0.292	21	63	84	74.3%
10th	62	0.286	0.287	0.285	25	60	85	75.2%
Average	53	0.286	0.287	0.286	23.2	62	85.2	75.4%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

Model: 146*4+80, Boruta Maxrun = 2000								
Times	**# Feature Selected**	**Training Step**			**Testing Step**			
		OER	**TER**	**FER**	**# H**	**# VH**	**Total (Acc.)**	
1st	57	0.281	0.280	0.282	21	71	92	81.4%
2nd	52	0.278	0.276	0.278	26	61	87	77.0%
3rd	58	0.295	0.295	0.295	23	64	87	77.0%
4th	53	0.288	0.287	0.288	24	58	82	72.6%
5th	62	0.279	0.280	0.278	22	67	89	78.8%
6th	52	0.297	0.299	0.297	19	71	90	79.6%
7th	54	0.289	0.287	0.290	30	57	87	77.0%
8th	48	0.297	0.295	0.298	22	57	79	69.9%
9th	48	0.292	0.291	0.292	21	63	84	74.3%
10th	62	0.286	0.287	0.285	25	60	85	75.2%
Average	54.6	0.288	0.288	0.288	23.3	62.9	86.2	76.3%

Abbr.: OER: Overall Error Rate, TER: True Error Rate, FER: False Error Rate, #H: number of High-Risk Level, #VH: number of Very High-Risk Level, Acc.: Accuracy (Total/113)

A.2 Figures

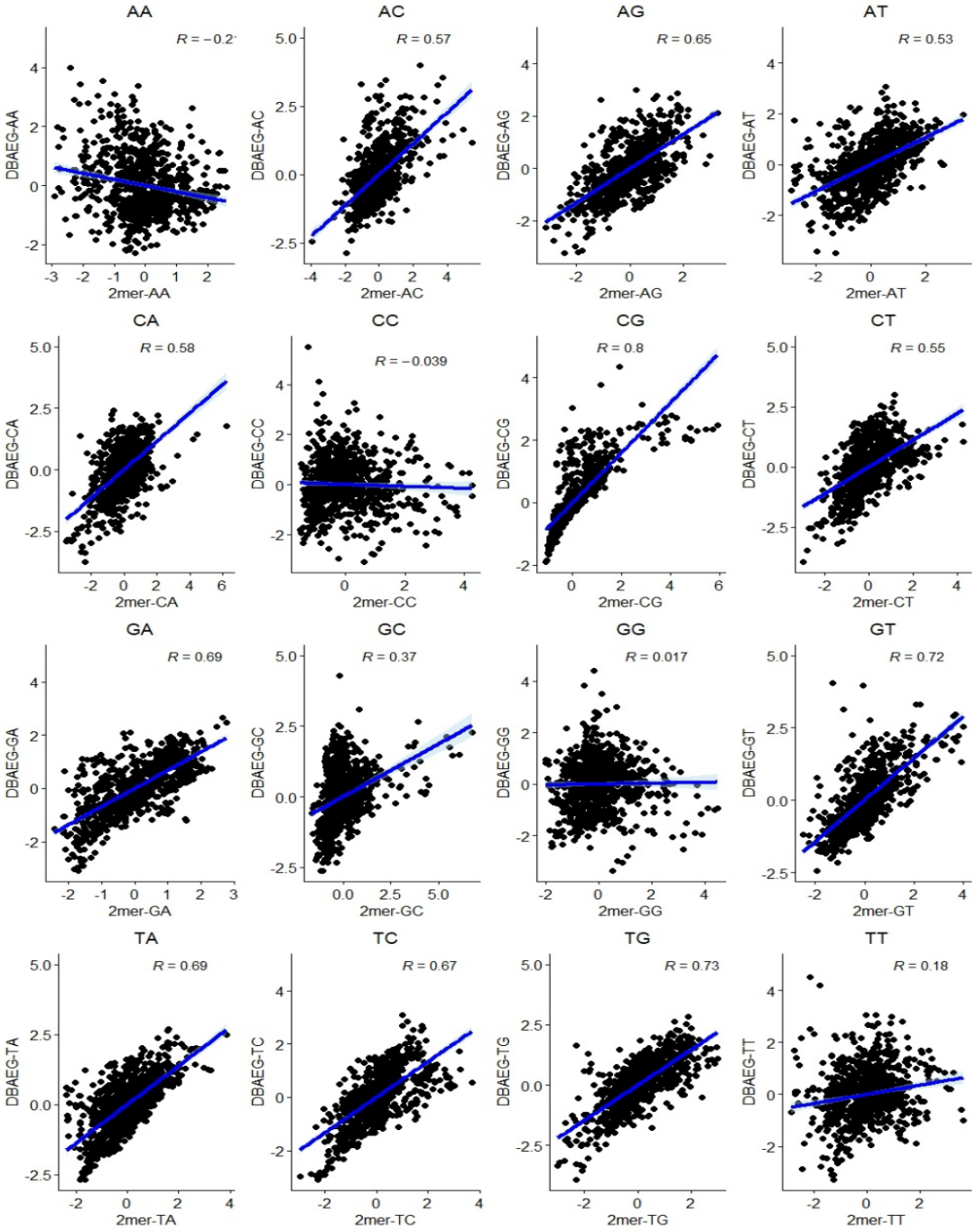
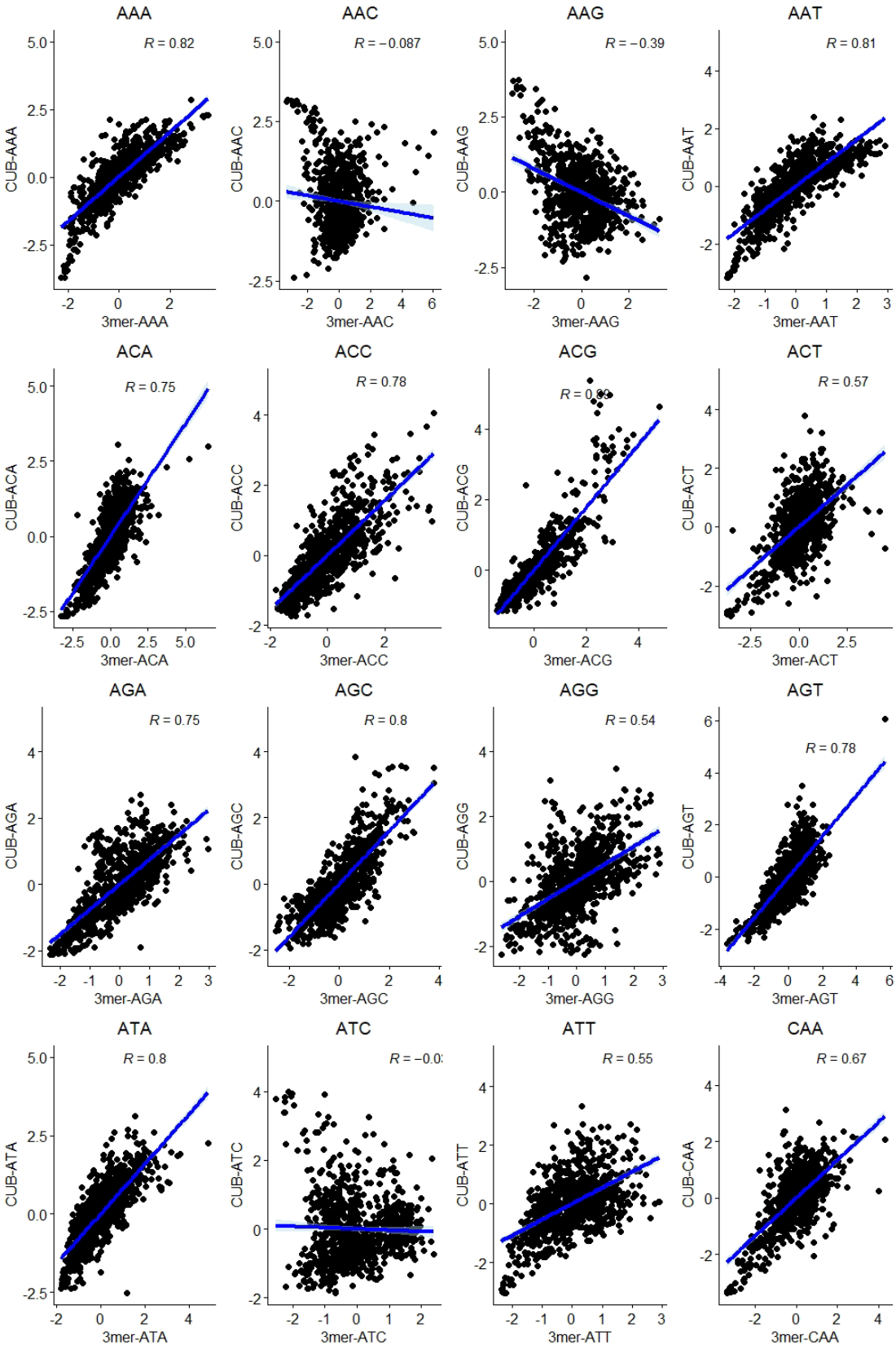
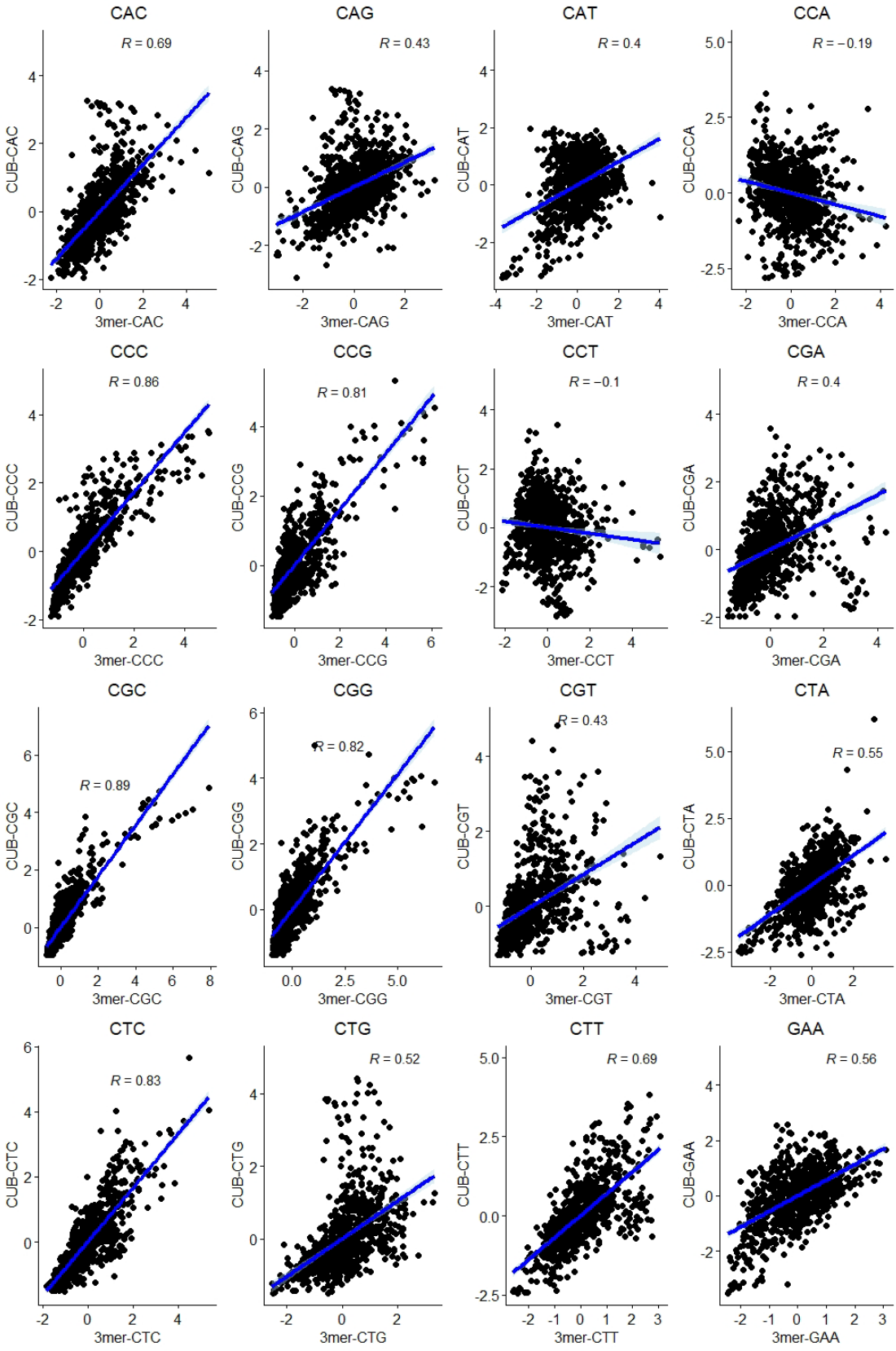
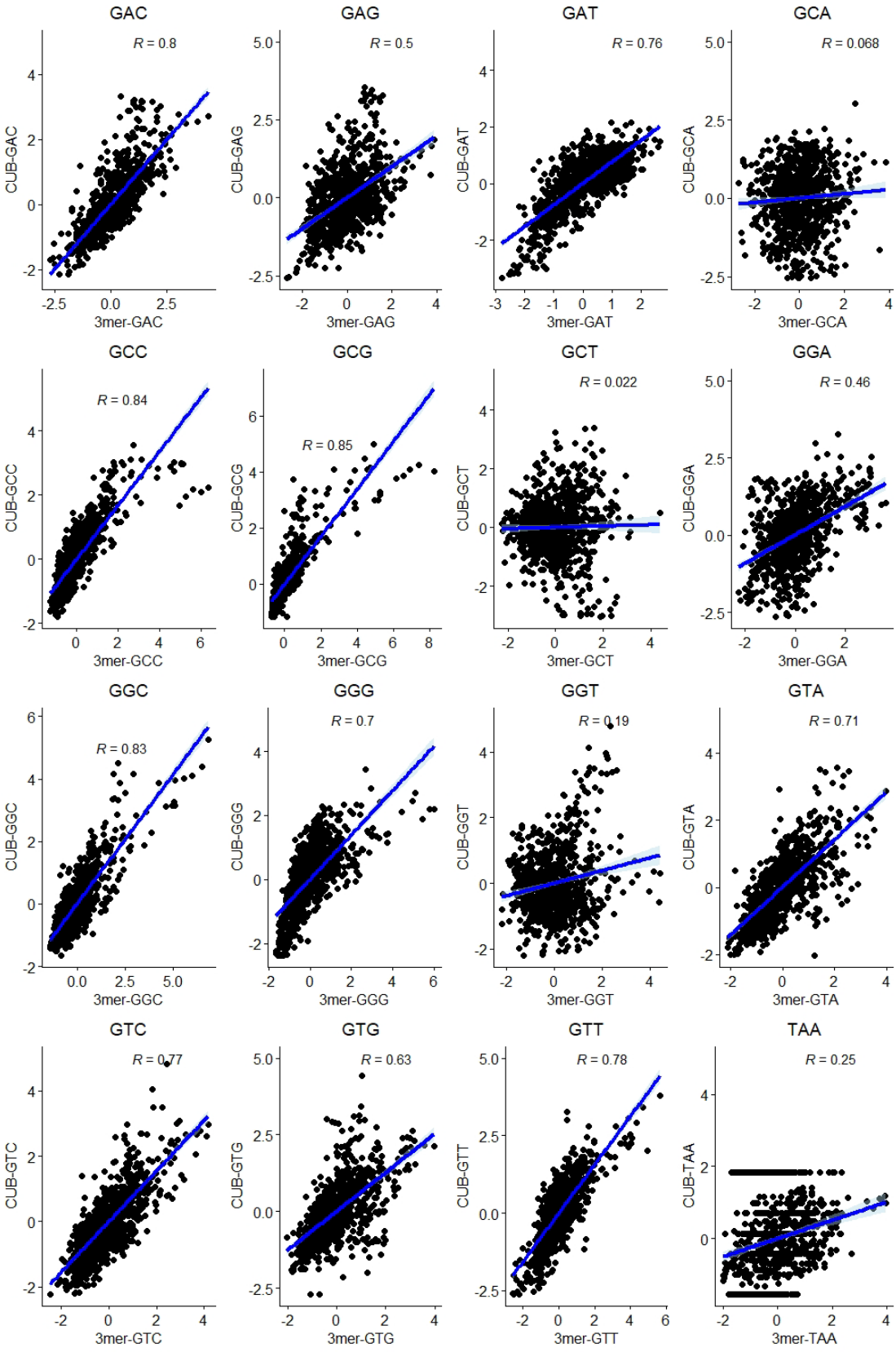


Figure A.2.1 The scatter plot and corresponding correlation coefficient between the 2-mer and dinucleotide biases across the entire genome (Abbr.: DBAEG) in current model.

The blue line represents the linear regression line, and the light blue region represents the 95% confidence interval of the regression line.


AAA
R = 0.82
CUB-AAA
3mer-AAA
AAC
R = −0.087
CUB-AAC
3mer-AAC
AAG
R = −0.39
CUB-AAG
3mer-AAG
AAT
R = 0.81
CUB-AAT
3mer-AAT
ACA
R = 0.75
CUB-ACA
3mer-ACA
ACC
R = 0.78
CUB-ACC
3mer-ACC
ACG
CUB-ACG
3mer-ACG
ACT
R = 0.57
CUB-ACT
3mer-ACT
AGA
R = 0.75
CUB-AGA
3mer-AGA
AGC
R = 0.8
CUB-AGC
3mer-AGC
AGG
R = 0.54
CUB-AGG
3mer-AGG
AGT
R = 0.78
CUB-AGT
3mer-AGT
ATA
R = 0.8
CUB-ATA
3mer-ATA
ATC
CUB-ATC
3mer-ATC
ATT
R = 0.55
CUB-ATT
3mer-ATT
CAA
R = 0.67
CUB-CAA
3mer-CAA


CAC
R = 0.69
CUB-CAC
3mer-CAC
CAG
R = 0.43
CUB-CAG
3mer-CAG
CAT
R = 0.4
CUB-CAT
3mer-CAT
CCA
R = −0.19
CUB-CCA
3mer-CCA
CCC
R = 0.86
CUB-CCC
3mer-CCC
CCG
R = 0.81
CUB-CCG
3mer-CCG
CCT
R = −0.1
CUB-CCT
3mer-CCT
CGA
R = 0.4
CUB-CGA
3mer-CGA
CGC
R = 0.89
CUB-CGC
3mer-CGC
CGG
R = 0.82
CUB-CGG
3mer-CGG
CGT
R = 0.43
CUB-CGT
3mer-CGT
CTA
R = 0.55
CUB-CTA
3mer-CTA
CTC
R = 0.83
CUB-CTC
3mer-CTC
CTG
R = 0.52
CUB-CTG
3mer-CTG
CTT
R = 0.69
CUB-CTT
3mer-CTT
GAA
R = 0.56
CUB-GAA
3mer-GAA


GAC
R = 0.8
CUB-GAC
3mer-GAC
GAG
R = 0.5
CUB-GAG
3mer-GAG
GAT
R = 0.76
CUB-GAT
3mer-GAT
GCA
R = 0.068
CUB-GCA
3mer-GCA
GCC
R = 0.84
CUB-GCC
3mer-GCC
GCG
R = 0.85
CUB-GCG
3mer-GCG
GCT
R = 0.022
CUB-GCT
3mer-GCT
GGA
R = 0.46
CUB-GGA
3mer-GGA
GGC
R = 0.83
CUB-GGC
3mer-GGC
GGG
R = 0.7
CUB-GGG
3mer-GGG
GGT
R = 0.19
CUB-GGT
3mer-GGT
GTA
R = 0.71
CUB-GTA
3mer-GTA
GTC
R = 0.77
CUB-GTC
3mer-GTC
GTG
R = 0.63
CUB-GTG
3mer-GTG
GTT
R = 0.78
CUB-GTT
3mer-GTT
TAA
R = 0.25
CUB-TAA
3mer-TAA

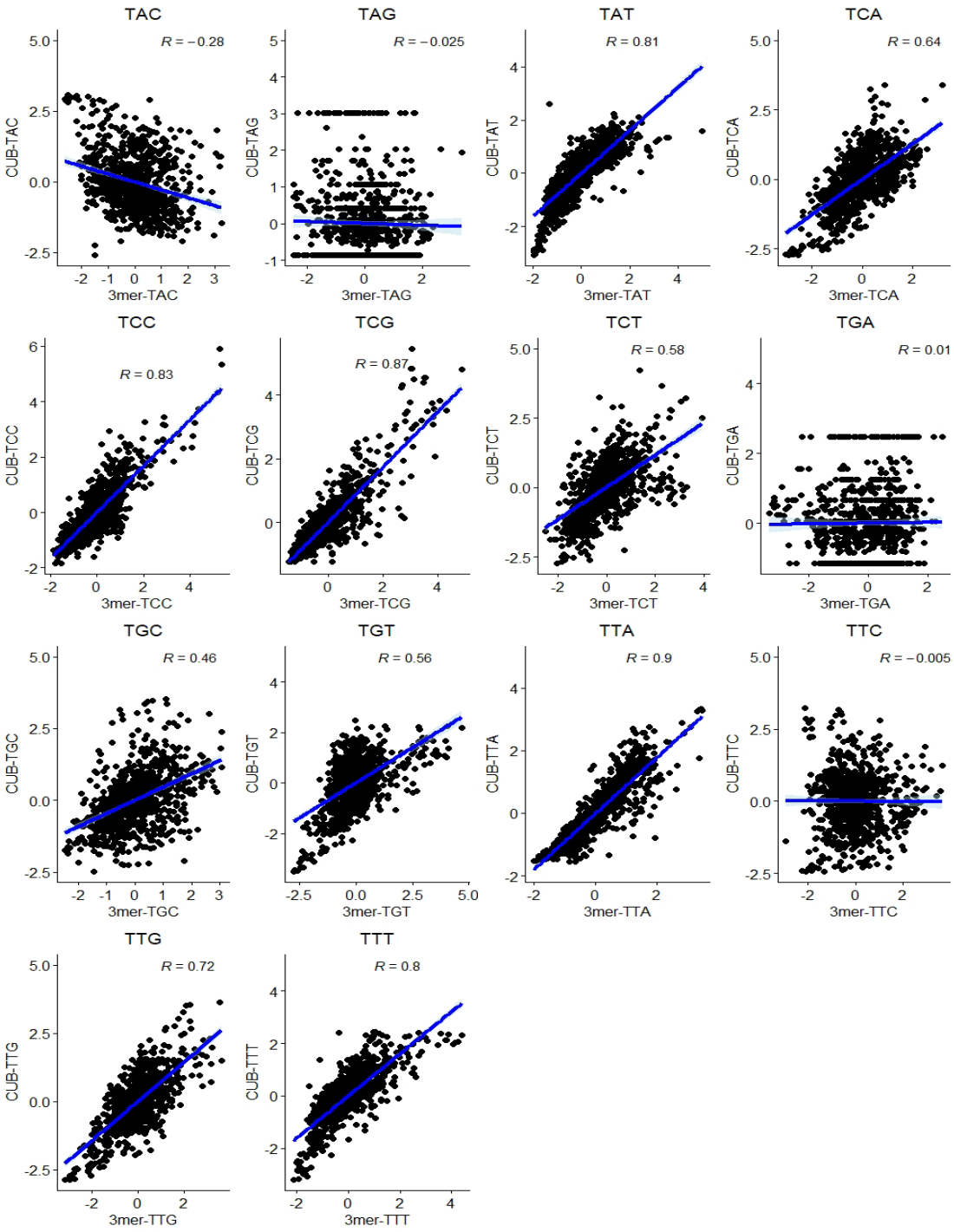


Figure A.2.2 The scatter plot and corresponding correlation coefficient between the 3-mer and codon usage biases in current model.

The blue line represents the linear regression line, and the light blue region represents the 95% confidence interval of the regression line.

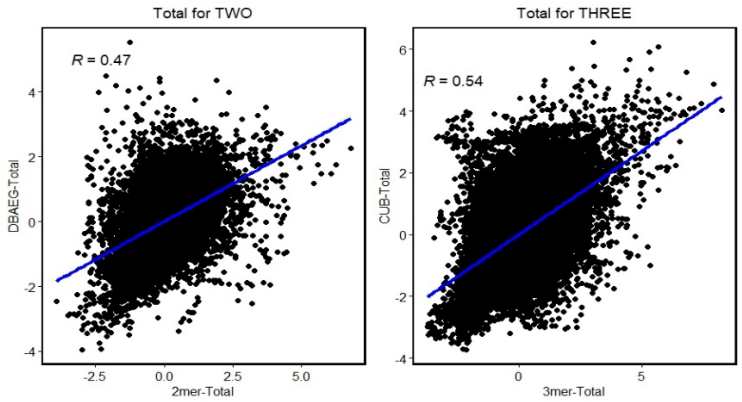


Figure A.2.3 The scatter plot and corresponding correlation coefficient for total features between the 2-mer and DBAEG (Dinucleotide Biases Across the Entire Genome) versus the 3-mer and CUB (Codon Usage Biases).

The blue line represents the linear regression line, and the light blue region represents the 95% confidence interval of the regression line.