Comparison of West Nile Virus and Yellow Fever Virus Using Apriori Algorithm, Decision Tree, and Support Vector Machine(SVM)

Seunghwan Lee, Changyoon Lee, Donghee Kim, and Taeseon Yoon

***Abstract*—West Nile virus (WNV), initially being identified in 1937, has largely spread throughout the world in the 20th century. Posing a number of serious threats to both human and animals, WNV even became the cause of the neuro-invasive diseases. The development of the vaccine against WNV has increased in importance nowadays as it is now considered to be an endemic pathogen in various regions. In this study, we compared the nucleotide sequence of West Nile Virus and Yellow Fever virus. As the complete vaccine against Yellow Fever virus (YFV) was currently developed, comparing the sequence of both viruses would provide the essential key of developing vaccine against WNV. Decision Tree, Apriori algorithm, and Support Vector Machine were used to reanalyze the whole sequence of WNV and YFV, representing some distinguishable features of both viruses.**

***Index Terms*—West nile virus, yellow fever virus, Apriori, decision tree, support vector machine.**

I. Introduction

West Nile virus (WNV), a mosquito-borne zoonotic arbovirus belonging to the genus *Flavivirus*, was first identified in Uganda, in 1937. Initially the virus was considered a minor risk for human beings, not until a series of global outbreak and the case of infection in New York City in 1999 were reported. The virus has spread globally for decades, now being commonly found in Africa, the Middle East, Europe, North America and West Asia. In 2012, WNV caused 286 deaths in United States, especially causing heavy damage on Texas, which is the most severe record ever [1]. Since no proper vaccines or treatments are developed currently, the comparison of the virus with the Yellow Fever virus (YFV), which also belongs to the *Flavivirus*, has increased in the importance.

A. Yellow Fever Virus

Yellow fever is an acute viral disease which causes fever, nausea, headaches, loss of appetite and abdominal pain. These symptoms usually improve within 3~4 days, but in some cases, liver damage causing yellow skin, bleeding, melena, and kidney damage occurs. It is believed to have originated from Central or East Africa, and has spread to South America through trade in the 17th century. Since the 1980s, yellow fever caused a lot of infections and death and these cases have been increased continuously, with nearly most of occurring in tropical areas of Africa and South America. Yellow fever virus is an enveloped RNA virus of the genus *Flavivirus* and this is mainly transmitted through the bite of female yellow fever mosquito *Aedes aegypti* [2].

B. West Nile Virus

West Nile Virus (WNV) is a disease which causes various symptoms like fever, infections of nervous system, poliomyelitis, encephalitis, muscle pain, rash in 20% infected people (80% of infections in humans are subclinical). According to WHO, in this year, total 493 cases occurred and 15 people died in the U.S. (By Oct. 2012, 4,249 people infected and 168 people died in the U.S.) [3]. WNV was first identified in a Ugandan woman's blood in 1937, and spread to Europe, Asia. In 1999, first American outbreak began in College Point, Queens in New York City and spread to other states, which causes continuous infections in the U.S every year. The genetic material of West Nile Virus is a positive-sense, single strand of RNA. And it is an arbovirus which belongs to *Flavivirus* and mainly transmitted through the bite of female mosquito or infected birds [4].

II. Materials and Methods

A. Materials

We obtained the nucleotide sequence data of the virus from NCBI. Complete genomes of each virus were pursued in order to compare various features of them. Fasta format of NCBI reference sequence *AY646354.1* was used for West Nile virus and *NC_002031.1* for Yellow Fever virus in the experiment.

B. Methods

1) Apriori algorithm

In the field of data mining, apriori algorithm is one of the most typical algorithms applicated while learning association rules. The fundamental basis of the algorithm is to trace association rules among the data by comparing the frequency of each data. When the transaction is provided, the algorithm observes an association rule between one itemset and other itemsets. As all the other similar algorithms do, this algorithm narrows the search space and deals with part of all rules. This algorithm uses 'support', which is the percentage of transactions that contain a specific itemset. When a particular itemset has larger support than 'minimum support', which is set up moderately, it is called 'frequent itemset', and we apply 'apriori principle' here to restrict the search space. Apriori

Manuscript received December 30, 2015; revised February 23, 2016.

Seunghwan Lee, Changyoon Lee, and Donghee Kim are with Hankuk Academy of Foreign Studies, Yonginsi, Republic of Korea (e-mail: glenlee3407@hafs.hs.kr, changyoon16@hafs.kr, 1021dh@naver.com).

Taeseon Yoon was with Korea University, Seoul, Republic of Korea. He is now with Hankuk Academy of Foreign Studies, Yonginsi, Republic of Korea (email: tsyoon@hafs.hs.kr).

doi: 10.18178/ijmlc.2016.6.2.591

principle is; all the subsets of the frequent itemset are also frequent itemsets, and itemsets which contain non-frequent itemset are also non-frequent itemsets. Based on the apriori principle, the apriori alogrithm undergoes level-wise process. Initially, this algorithm calculates the support of 1-itemset (itemset composed of an item) and extracts 1-itemsets which are "frequent itemset". Next, it picks out 2-itemset that might be a frequent itemset based on apriori principle. This process is called 'join' step. Then, the algorithm calculates the support of candidate 2-itemsets and extracts frequent itemset once again. This process is called pruning step. These procedures are repeated until it cannot find frequent k-itemset [5]. In this study, we applied the algorithm three times by constituting data transactions composed of 9, 13, and 17 bases. We extracted the association rule among the bases in the same position of the transaction.

2) Decision tree algorithm

The decision tree is an algorithm which is used commonly to serve classification in data mining. The primary intention of the algorithm is to approximate discrete-valued target function. This function is represented by a decision tree. 'Entropy' is a value that exhibits the impurity of a data group.

$$\text{Entropy(S)} = -p\log p \qquad (1)$$

If a specific group had high entropy, that group would be impure, which means data are displayed disorderly. 'Information gain' is the amount of entropy decreased when the data are class distinguished based on certain criteria [6].

$$\in Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)(S_v = \{s \in S | A(s) = v\})$$

The attribute that incur higher information gain makes divided group with lower impurity, and this is when the classification between the groups is clear. In other words, the attribute with higher information gain is more dominant attribute. When expressed in a form of decision tree, the attribute with higher information gain would show up early division of the branches [7]. Again, we applied the algorithm three times by constituting data transactions composed of 9, 13, and 17 bases. The attributes set up in the algorithm is the position of the base.

3) Support vector machine(SVM)

Support Vector Machine is a powerful algorithm mainly applied for pattern recognition, regression analysis, and classification. When data sets which could be divided into two categories are provided, SVM generates non-statistical binary linear classification model that determines in which category a new data set would be included [8]. SVM consists of hyperplanes that could be used during classification or regression analysis. Since it has smaller classifier error when a hyperplane has longer distance from the closest data sample, the ultimate object of the algorithm is to find hyperplane that has the longest distance from certain classified data to the closest data sample. It selects hyperplane that has the largest margin between two classes [9]. 4 kinds of kernel functions were used in the experiment: normal, polynomial1, polynomial2, Gaussian radical basis function kernel (RBF kernel). Normal and polynomial1 kernel functions belong to linear SVM, and the rest of them belong to nonlinear SVM. Both linear and nonlinear SVM were used in the experiment in order to clearly identify whether the viruses can be classified as different virus or not.

III. Results

A. Apriori Results

Certain rules of each virus were extracted by Apriori algorithm. We analyzed particular amino acids existing on particular position of genes by splitting the complete genome of the viruses under 9, 13, 17 window. Table I to 3 below show all the rules extracted from viruses under each window.

TABLE I: Rule Extraction under 9 Window

Virus	Rule

West Nile virus	Position 1=L41 Position 2=L47
	Position3=L47 Position3=V41
	Position5=L41 Position9=A47

Yellow Fever virus	Position1=G44 Position3=G47
	Position4=G40 Position5=R42
	Position7=G47 Position8=R42
	Position9=G41

TABLE II: Rule Extraction under 13 Window

Virus	Rule

West Nile virus	Position2=L33 Position3=T31
	Position4=A28 Position5=A30
	Position5=L30 Position6=V28
	Position8=G32 Position10=L30

Yellow Fever virus	Position2=G31 Position4=G29
	Position5=G29 Position6=G30
	Position6=S30 Position7=S30
	Position8=G31 Position11=G30
	Position12=G35 Position13=G28

TABLE III: Rule Extraction under 17 Window

Virus	Rule

West Nile virus	Position1=T22 Position3=L25
	Position5=V25 Position5=A23
	Position7=L32 Position8=L23
	Position9=L25 Position10=E22
	Position11=A26 Position11=V23
	Position13=V28 Position13=G24
	Position14=G22 Position16=L27
	Position16=T25

Yellow Fever virus	Position1=R25 Position1=S21
	Position2=G28 Position2=S21
	Position3=S21 Position4=G24
	Position4=S21 Position6=G26
	Position8=E26 Position8=G21
	Position10=G21 Position10=S21
	Position11=G21 Position13=G31
	Position14=G32 Position17=G24

According to Table I, we could clearly recognize the difference of amino acid distribution between the two viruses. Leucine (L) was most frequently shown in West Nile virus. 4 rules for it were extracted out of 6 rules. Besides, the number of Glycine (G) was the largest in Yellow Fever virus, which was not shown in West Nile virus. Under 13window, more diverse amino acids were shown in both viruses comparing to the results under 9window. Rules of Threonine (T), Glycine (G), Valine (V), and Alanine (A) were found in West Nile virus, and Serine (S) was newly identified in Yellow Fever virus. Still, Leucine and Glycine occupied the largest portion in each virus. Throughout the results under 9,13,17 window, it is noticeable that Leucine (L) is the most frequent amino acid in West Nile virus, Glycine (G) in Yellow Fever virus likewise. Such results imply that those amino acids represent each virus, featuring distinctive characteristics of them. Glycine (G) and Glutamic acids (E) were the amino acids that commonly shown in both viruses, which accounts for the similar symptoms of the diseases caused by those viruses. Fig. 1 to 3 indicates the number of represented amino acids in each virus.

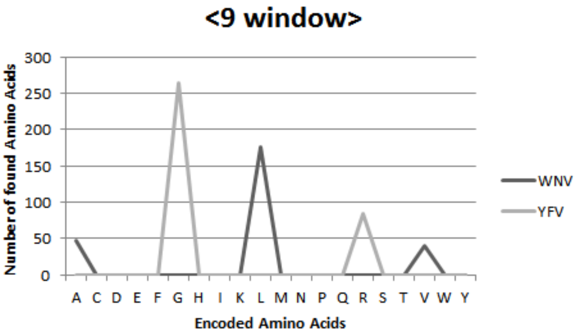

Fig. 1. Results under 9 Window.

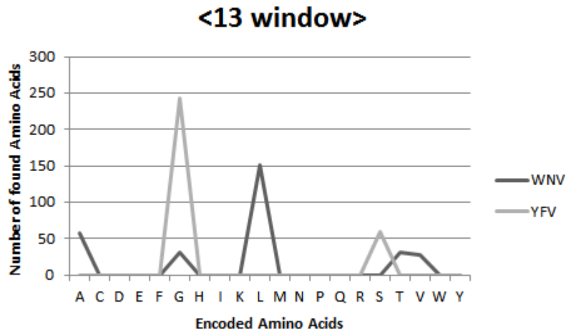

Fig. 2. Results under 13 Window.

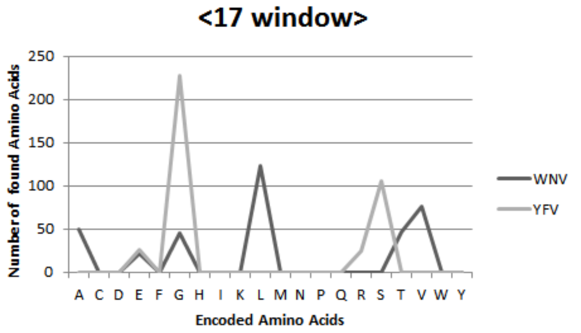

Fig. 3. Results under 17 Window.

B. Decision Tree Results

We implemented the Decision Tree algorithm with item sets which consist of 9, 11, and 13 bases. As we conducted experiment in 10 fold cross validation, overall 10 experiments were operated each. We considered the rules with accuracy lower than 0.8 negligible because we concluded that they don't reflect the characteristic of the virus accurately. Then, we extracted the repeated rules throughout the 10 folds of experiments.

TABLE IV: RULE EXTRACTION UNDER 9 WINDOW

Virus	Rule
West Nile virus	Position 4=M Position 5=M Position 9=D position 5=I Position 9=A Position 4=A
Yellow Fever virus	Position 4=G Position 9=L Position 5=G Position 5=P

TABLE V: RULE EXTRACTION UNDER 13 WINDOW

Virus	Rule
West Nile virus	Position11=Y
Yellow Fever virus	Position 1=H Position 6=H Position 12=H Position 12=C

TABLE VI: RULE EXTRACTION UNDER 17 WINDOW

Virus	Rule
West Nile virus	Position 7=Y Position 2=I Position 7=M Position 11=V Position10=T Position2=T Position 7=G Position 7=S
Yellow Fever virus	Position 7=C Position 7=Q Position 10=H Position 10=C Position 11=Q Position 7=E Position 15=L

From the amino acid rules extracted from the algorithm, we can predict both the amino acid distribution in certain part of the genes in viruses and the similarity of amino acid distribution between the two viruses. It is noticeable that throughout all the 3 experiments, there weren't any the same amino acid rule extracted between two viruses. This shows that these two viruses are clearly distinguishable, because it states that the amino acids composing the protein of the two viruses clearly show difference. Generally, the kinds of amino acid appeared as a result in decision tree were quite different from those appeared in apriori algorithm. However, it is remarkable that glycine was extracted as a rule in position 4 and 5 in Table IV. This supports that there would be a lot of genetic codes which represents glycine in certain parts of genes in Yellow fever virus. This result also accords with that of apriori algorithm. Furthermore, in certain part of gene in West Nile virus, DNA sequence would encode methionine, isoleucine, and tyrosine a lot. On the other hand, in certain part of gene in Yellow Fever virus, parts representing glycine, hystidine, and cysteine would emerge frequently. In addition, as different amino acid rules were extracted in the same position, the two viruses would have different kinds of amino acid in the certain parts of genes, which contributes to the distinction of two viruses. However, the error rate of the experiment was relatively high, which means the DNA sequence of the viruses couldn't be easily classified. This contradicted with our results and we decided to use SVM to clarify if the two viruses have certain similarity in their genes.

C. SVM Results

Even though results of Apriori and Decision tree algorithm show West Nile virus and Yellow Fever virus has quite different characteristics, they never provide us with scientifically accurate differences. SVM was used to clearly distinguish the viruses by using both linear-classification and nonlinear-classification. Same as we did in the previous experiments, we divided the whole genome of the virus under 9, 13, 17 window. 10-fold cross-validation was adopted for more credible results, which means 10 experiments were made under each window. 10 data sets made of particular number of bases were formed. For each experiment, 9 datasets were used to train the model, and only one was involved in the process of testing. Every 10 dataset can become a test data only once in whole experiments. Table VII ~ IX below indicate the accuracy values and the average of them in each experiment.

TABLE VII: Results under 9 Window Accuracy

Times	1	2	3	4	5	6	7	8	9	10	mean
Normal	.525	.488	.500	.538	.500	.488	.500	.525	.500	.513	**.5075**
Polynomial	.550	.750	.588	.588	.538	.563	.488	.463	.450	.550	**.5525**
Polynomail2	.588	.613	.488	.488	.525	.475	.525	.438	.450	.538	**.5125**
RBF	.813	.775	.863	.850	.788	.800	.713	.800	.825	.800	**.8025**

TABLE VIII: Results under 13 Window Accuracy

Times	1	2	3	4	5	6	7	8	9	10	mean
Normal	.500	.500	.500	.500	.500	.500	.500	.500	.500	.500	**.5000**
Polynomial	.517	.633	.500	.683	.517	.600	.617	.567	.633	.583	**.5850**
Polynomail2	.550	.617	.567	.483	.567	.600	.533	.450	.500	.533	**.5000**
RBF	.850	.883	.817	.850	.833	.900	.700	.817	.883	.850	**.8383**

TABLE IX: Results under 17 Window Accuracy

Times	1	2	3	4	5	6	7	8	9	10	mean
Normal	.500	.575	.525	.500	.500	.500	.475	.525	.500	.488	**.5088**
Polynomial	.525	.600	.600	.675	.725	.600	.400	.575	.525	.561	**.5786**
Polynomail2	.800	.850	.775	.900	.700	.750	.625	.775	.925	.805	**.7905**
RBF	.725	.700	.775	.950	.800	.800	.625	.775	.750	.854	**.7754**

The results derived from SVM algorithm were breathtaking. We considered the accuracy of 17 window the most essential one, because the longer DNA sequences can reveal the characteristics of virus more accurately. Applying normal function to DNA-sequence-based data transactions, we could get relatively low accuracy (about 0.5) in 30 times of experiments (10 times each for 9 window, 13 window, and 17 window). The tendency was maintained in experiment using polynomial1 function which is also a linear kernel function. The accuracy from polynomial 1 function was higher than those from normal function, but it was still relatively low (about 0.55~0.58). This implies that data sets are not classified linearly well enough. It is because the algorithm couldn't assort DNA sequence pattern of two viruses, which means they are similar to each other. Even though two viruses showed definite difference in their amino acid distribution, we concluded that the overall DNA sequences of two viruses are considerably similar to each other, which means that they are basically similar viruses. High accuracy rates were observed in experiments using polynomial2 function and Gaussian radical basis function. It was obvious result because the data transactions would be classified better with non-linear functions. Still, our focus was on the fact that the DNA-sequence-based data weren't classified linearly and two viruses were similar.

IV. Conclusion and Discussion

Initially we issued the research, because two viruses seemed to have possibility of sharing similar genetic characteristics as both of them cause endemic diseases in the similar region. We implemented the analysis using 3 algorithms; Apriori algorithm, Decision tree, and SVM. These algorithms were applied at DNA sequence divided in to 7 nucleotides, 13 nucleotides, and 17 nucleotides (called 'window').

If the same amino acid rules were extracted between two viruses as a result of apriori algorithm, it would stand for high possibility of existence of the DNA codes translated into the same amino acid in the similar sector in two viruses. However, different amino acid rules were extracted in the same position (refers to position of nucleotide in created data transaction) of two viruses from all experiments, which implies that the DNA sequences which encode the similar parts of the virus are distinguishable. In addition, while leucine was most frequently shown in West Nile virus, glycine was the largest in Yellow Fever virus. This stands for the possibility of frequent existence of DNA sequence mainly translated into leucine in WNV, and glycine in the case of YFV. This is also remarkable that these tendencies are compatible with the symptoms of viruses. While viruses with large amounts of glycine is known to cause fever-like symptoms, viruses with large amount of leucine are likely to affect the muscle of hosts; and these accord with the symptom of WNV and YFV.

The results of decision tree also supported the distinguishable difference between two viruses. As mentioned above, they represented considerable difference in DNA sequence. Furthermore, methionine, isoleucine, and tyrosine account for large portion of amino acid distribution in WNV. On the other hand, parts translated into glycine, hystidine, and cysteine would emerge frequently in YFV. It was also remarkable that glycine rules were extracted in 7 window of YFV which strengthens the credibility of the results of the apriori algorithm.

Since we judged that scientific evidence was inadequate to judge two viruses are different at all with only amino acid rules, we conducted experiment once again using SVM. Low accuracy rates were shown in experiment using normal function of SVM which means that the sequence of two viruses couldn't be classified linearly. Furthermore, the accuracy value remained low in experiments using polynomial1 function. From here we could conclude that these viruses could not be classified linearly and they have certain similarity in common. This seemed to contradict with the results of the apriori algorithm and decision tree algorithm. We interpreted these results as the fact that they are radically similar viruses though they differ in amino acid distribution.

If two viruses shared definite similarity, the complete vaccine against YFV could also be effective to WNV, and since they are basically similar, YFV's vaccine can be effective to WNV to some degree. However, according to our research these two viruses have different amino acid distribution which makes them different from each other, which implies that application of YFV's vaccine to WNV wouldn't be a wise method to be recommended. Further in-depth research should be conducted to figure out what makes two viruses so similar despite the clear difference in amino acid composition. Furthermore, vaccination against WNV, which must be different from that of YFV should be completed and our amino acid distribution data would be valuably put to use in here.

References

[1] D. Nash, F. Mostashari *et al*., "The outbreak of West Nile virus infection in the New York City area in 1999," *N. Engl. J. Med.,* vol. 344, no. 24, pp. 1807–1814, June 2001.

[2] M. A. Tolle, "Mosquito-borne diseases," *Curr Probl Pediatr Adolesc Health Care*, vol. 39, no. 4, pp. 97–140, April 2009.

[3] K. O. Murray, D. Ruktanonchai, D. Hesalroad, E. Fonken, and M. S. Nolan, "West Nile virus, Texas, USA, 2012," *Emerging Infectious Diseases*, vol. 19, no. 11, pp. 1836–1838, November 2013.

[4] R. S. Lanciotti, G. D. Ebel *et al*., "Complete genome sequences and phylogenetic analysis of West Nile virus strains isolated from the United States, Europe, and the Middle East," *Virology*, vol. 298, no. 1, pp. 96–105, June 2002.

[5] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. the 20th International Conference on Very Large Data Bases*, Santiago, Chile, September 1994, pp. 487-499.

[6] J. R. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, no. 3, p. 221, 1987.

[7] S.-H. Cha and C. C. Tappert, "A genetic algorithm for constructing compact binary decision trees," *Journal of Pattern Recognition Research*, vol. 4, no. 1, pp. 1–13, 2009.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, p. 273, 1995.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. he Fifth Annual Workshop on Computational Learning Theory*, 1992, p. 144.

Seunghwan Lee was born in Seoul, Korea, in 1999. He is currently with Hankuk Academy of Foreign Studies majoring in natural science. He is interested in bioinformatics, especially analyzing genomes of virus with machine learning algorithms.

Changyoon Lee has been with Hankuk Academy of Foreign Studies majoring in natural science since 2014. His current interests include biology and computer science.

Donghee Kim was born in Suwon, Republic of Korea, in October 21, 1998. He is currently a student in natural science major of Hankuk Academy of Foreign Studies, Republic of Korea. He is deeply interested in ethology and bioinformatics.

Taeseon Yoon was born in Seoul, Korea in 1972. He received his Ph.D. degree in computer education from the Korea University, Seoul, Korea, in 2003. From 1998 to 2003, he was with EJB analyst and SCJP. From 2003 to 2004, he joined the Department of Computer Education, University of Korea, as a lecturer and Ansan University, as an adjunct professor. Since December 2004, he has been with the Hankuk Academy of Foreign Studies, where he was a computer science and statistics teacher. He was the recipient of the Best Teacher Award of the Science Conference, Gyeonggi-do, Korea, 2013.