Improving Classification Performance Using Genetic Programming to Evolve String Kernels

Ruba Sultan[1], Hashem Tamimi[1,2], and Yaqoub Ashhab[2]
[1]College of IT and Computer Engineering, Palestine Polytechnic University, Palestine
[2]Palestine-Korea Biotechnology Center, Palestine Polytechnic University, Palestine

Abstract: *The objective of this work is to present a novel evolutionary-based approach that can create and optimize powerful string kernels using Genetic Programming. The proposed model creates and optimizes a superior kernel, which is expressed as a combination of string kernels, their parameters, and corresponding weights. As a proof of concept to demonstrate the feasibility of the presented approach, classification performance of the newly evolved kernel versus a group of conventional single string kernels was evaluated using a challenging classification problem from biology domain known as theclassification of binder and non-binder peptides to Major Histocompatibility Complex Class II. Using 4794 strings containing 3346 binder and 1448 non-binder peptides, the present approach achieved Area Under Curve=0.80, while the 11 tested conventional string kernels have Area Under Curve ranging from 0.59 to 0.75. This significant improvement of the optimized evolved kernel over all other tested string kernels demonstrates the validity of this approach for enhancing Support Vector Machine classification. The presented approach is not exclusive for biological strings. It can be applied to solve pattern recognition problems for other types of strings as well as natural language processing.*

Keywords: *Support vector machine, string kernels, genetic programming, pattern recognition.*

Received October 31, 2015; accepted June 1, 2016

1. Introduction

Support Vector Machine (SVM) is a kernel-based supervised learning technique that has been successfully applied to solve various types of classification, clustering, and regression problems [2, 9]. Using kernel function, SVM can efficiently extract and map the hidden relations among a set of labeled training data.

The basic objective of SVM-based classification is to use kernel functions in order to transform overlapping classes of ambiguous data to a high-dimensional feature space, where the data classes become more separable [21]. Constructing a feature space of a valid kernel should follow the Mercer's Theorem that conserves the Gram and kernel matrices positive semi-definite [15]. Also, kernels must satisfy a number of closure properties that enable constructing more complicated kernels from simple kernels. Specific mathematical operations can be applied to aset of kernels to produce a new valid kernel [21].

There are different types of kernels that can be used to develop SVM including linear, string, polynomial and Gaussian kernel [11, 15]. String kernels are widely used for processing natural text and biological sequences. They have been successfully used to solve many pattern recognition problems in biological sequences, including Deoxyribonucleic Acid (DNA), Ribonucleic Acid (RNA) and protein [3, 16].

String kernels introduce embedding two strings in a high dimensional space in such a way that their relative distance in that space reflects their similarity. Then, the inner product between the embedded sequences can be computed easily [15]. Most string kernels involve counting the substrings that the two main strings have in common. In biological applications, finding sequence similarity between two sequences is a fundamental approach to infer functional similarities between these sequences [13].

The most important decision in developing an SVM is the selection of the optimal kernel function among a collection of different kernels. The development of new kernels and the optimization of their parameter is still a big challenge in machine learning field, where novel solutions are greatly needed.

In general, two kinds of approaches are followed to optimize the use of different kernels for SVM: the first approach is to examine the available kernels one by one in order to find a good kernel for the problem of interest and this is a cumbersome method for kernel optimization. The second approach is to combine different numerical kernels using an optimization technique such as Genetic Programming (GP), Multiple Kernel Learning (MKL), and ensemble methods that can determine the weights for each of the tested kernels [1, 5, 6, 8, 10, 19].

There has been considerable effort to solve the problem of optimizing numerical kernels. However, less attention has been paid to investigate the use of evolutionary-based optimization techniques in improving the efficiency of string kernels. The aim of the present work is to explore and evaluate genetic

programming as an evolutionary-based approach to solve the problem of searching and evolving optimal string kernel.

In order to test the performance of the newly optimized string kernels selection approach, we sought to test it on a rather challenging string classification problem. We decided to use a well-known challenging problem from the bioinformatics domain known as the classification of binding and non-binding peptides to the Major Histocompatibility Complex class II (MHC-II) molecules [22]. MHC-II molecules are the key players of the immune system that can recognize specific sequence patterns in bacterial proteins. Proteins of bacteria are usually chopped down by immune system into short strings (peptides). These peptides are divided into two classes: those that are able to bind to MHC-II molecules and hence they can stimulate immune response whereas the second class that are unable to bind to MHC-II molecules cannot stimulate immune system [12]. Classification of binder and non-binder peptides to MHC-II is a well-known challenging classification problem that has been addressed by many machine learning methods [12].

2. Research Methodology

2.1. Optimization Process

The optimization process of the GP starts with a collection of potential kernels. Then, creates new potential solutions through the use of the genetic operators: crossover and mutation. These potential solutions are selected on the basis of their quality as solutions to the problem using a fitness function. GP uses this process iteratively to generate new collections of potential solutions until some stopping criterion is met. Once GP optimization is finished, the resulted evolved kernel is embedded in SVM for solving classification problem. Figure 1 illustrates the general diagram of the optimization process. The optimization process goes through the following steps:

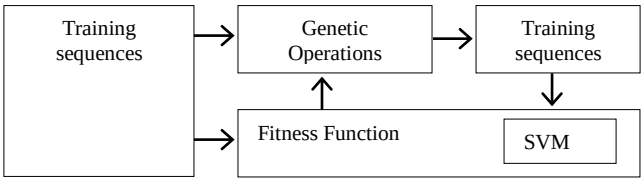

Figure 1. A block diagram of the optimization process.

Constructing a set of candidate solutions in a semi-random manner, where each solution is represented as a tree structure. The terminal nodes of the trees are selected from a set of predefined string kernels with the corresponding parameters. Other terminal nodes are allowed to hold numeric values that represent the weights of the importance of the different kernels. The non-terminal nodes contain the possible mathematical operations on the kernels that preserve Mercer theorem and closure properties.

An example of a GP potential solution expressed as:

$$a_1 \cdot k_1(x_1)+a_2 \cdot k_2(x_2)+\cdots+a_n \cdot k_n(x_n)$$

Where a_i is a weight adjusted during the evolution of the solution, x_i represent a corresponding parameter set to the kernel k_i, which is also optimized using the GP. Another example of GP potential solution expressed as more general mathematical expression is:

$$a_1 \cdot k_1(x_1)+ a_2 \cdot e^{(k_2((x_2)))} \times a_3 \cdot k_3(x_3)$$

For the implementation of GP, we used the GPLAB [17] library under Matlab environment. A Summary of the single kernels used in this work and their mapping formula is shown in Table 1. These kernels are implemented in Shogun toolbox [18].

1. Applying the GP operations to the candidate solutions to produce new candidate solutions. The effect of these operations will influence the contribution of the different kernels, their weights of importance and their corresponding parameters.
2. Selecting the partial optimal candidate solutions based on the fitness function in Equation (1). This fitness involves testing the candidate solution using SVM.

The fitness of the solution is measured according to the following Equation:

$$\text{Fitness}=\text{Sensitivity}\times\text{Specificity} \quad (1)$$

The sensitivity and specificity are defined as:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \quad (2)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN + FP}} \quad (3)$$

Where *TP*, *TN*, *FP*, and *FN* are True Positive, True Negative, False Positive and False Negative, respectively.

4. The set of candidate solutions, which leads to better classifications is allowed to survive for another GP generations.
5. In order to terminate the optimization process, the convergence of the optimization process was monitored by examining the average fitness values.

2.2. String Dataset

To evaluate the proposed approach, we tested it on a benchmark dataset of peptide strings that are labeled either as binder or non-binder to MHC-II. This particular model is still among the most challenging classification problems in bioinformatics.

Table 1. Summary of string kernels used in this work.

Kernel name	Formula
Spectrum	$k_p(s,t) = \langle \emptyset^p(s), \emptyset^p(t) \rangle = \sum_{u \in \Sigma^p} \emptyset_u^p(s)\, \emptyset_u^p(t),$ $\emptyset_u^p(s) = \lvert\{(v_1, v_2) : s = v_1 u v_2\}\rvert, u \in \Sigma^p$
Weighted Spectrum	$k_p(s,t) = \sum_{p=1}^{P} \beta_p \emptyset_p(s) \emptyset_p(t)$
Fixed degree	$k_p(s,t) = \langle \emptyset^p(s), \emptyset^p(t) \rangle = \sum_{u \in \Sigma^p} \emptyset_u^p(s)\, \emptyset_u^p(t),$ $\emptyset_u^p(s) = \lvert\{i : u = s(i)\}\rvert, u \in \Sigma^p$
Polynomial	$k_p(s,t) = (\sum_{i=0}^{L} I(s_i = t_i) + c)^d$
Locality improved	$k_p(s,t) = (\sum_{p=1}^{N} win_p(s,t))^{d_2}$ $win_p(s,t) = (\sum_{j=-l}^{+l} w_j match_{p+j}(s,t))^{d_1}$
Local Alignment	$k_{LA}(s,t) = \sum_{i=0}^{N} k_i(s,t)$ $k_{(n)}(s,t) = k_{const} \cdot (k_{align} \cdot k_{gap})^{(n-1)} \cdot k_{align} \cdot k_{const}$ $k_{const}(s,t) = 1$ $k_{align}(s,t) = \{(0, \text{if } \lvert s\rvert \neq 1 \vee \lvert s\rvert \neq 1), (e^{\beta * S(s,t)}, \text{otherwise})\}$ $k_{gap}(s,t) = e^{\beta(g(\lvert s\rvert) + g(\lvert t\rvert))}$
Weighted Degree Position	$k(s,t) = \sum_{w=1}^{d} w_w \sum_{i=1}^{N-d} I(u_{w,i}(s) = u_{w,i}(s))$
Mismatch	$k_{(k,m)}(s,t) = \langle \emptyset_{(k,m)}(s), \emptyset_{(k,m)}(t) \rangle$ $\emptyset_{(k,m)}(s) = \sum_{k-mers\ \alpha\ in\ s} \emptyset_{(k,m)}(\alpha)$ $\emptyset_{(k,m)}(\alpha) = (\emptyset_\beta(\alpha))_{\beta \in A^k}$
TOP	$k(s,t) = v(s,\theta)v(t,\theta)$ $v(s,\theta) = \log(P(y = +1\lvert s,\theta)) \log(P(y = -1\lvert s,\theta))$
Salzberg	$k(s,t) = (\sum_{p=1}^{N} win_p(S_s, S_t))^{d_2}$ $win_p(S_s, S_t) = (\sum_{j=-l}^{+l} w_j match_{p+j}(S_s, S_t))^{d_1}$ $s_p(x) = \log \frac{P(x_p \text{ at pos. p in Truedata} \lvert x_{p-1} \text{ at pos. } p-1 \text{ in Truedat}}{P(x_p \text{ at pos. p in Alldata} \lvert x_{p-1} \text{ at pos. } p-1 \text{ in Alldata})}$

In fact, the different conventional string kernels showed a relatively poor performance in solving this problem. The peptide dataset for the MHC-II benchmark was obtained from the NetMHCII 2.2 server(www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php).The data used in our experiments is DRB1*0101 datasets. We use 4794 sequences, divided into 1448 non-binder sequences, and 3346 binder sequence.

3. Validation

In order to compare the performance of the evolved kernel versus the conventional string kernel, the following performance measures were calculated for each kernel: specificity Equation (2), sensitivity, Equation (3), accuracy Equation (4), Positive Predictive Value (PPV) (Equation 5), Negative Predictive Value (NPV) Equation (6), and fitness Equation (1). The Receiver Operating Characteristic (ROC) curve, which plots the true positives (sensitivity) vs. false positives (1-specificity), was used to calculate the Area Under the Curve (AUC).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$PPV = \frac{TP}{TP + FP} \quad (5)$$

$$NPV = \frac{TN}{TP + FN} \quad (6)$$

4. Experimental Results

To test the performance of the different single kernels versus the evolved GP kernel, a k=3-fold cross validation was performed with the data being shuffled each fold. The whole 3-fold cross-validation experiment was repeated 5 times on equal size of binder and non-binder data to ensure a good coverage. Each experiment consisted of a population of 150 kernels that were sufficient for exploring the search space.

Table 2. Performance comparison of conventional string kernels versus the newly evolved kernel.

Kernel	Spec.	Sens.	Acc.	PPV	NPV	Fitness
Spectrum	67.53	66.17	66.66	67.09	66.85	45.86
Weighted-Spectrum	69.66	67.11	67.95	68.92	68.39	46.67
Salzberg	87.74	20.19	52.36	62.26	53.96	17.89
Local-alignment	65.61	65.29	65.39	65.54	65.45	42.77
Fixed-degree	61.56	61.31	61.41	61.51	61.44	38.07
TOP	54.38	58.73	56.89	56.35	56.55	32.48
Locality-improved	91.78	26.37	55.51	76.45	59.06	24.87
Match-word	66.16	64.63	65.17	65.66	65.40	42.08
Polymatch	68.38	69.03	68.84	68.61	68.70	46.64
Weighted-degree	66.69	67.47	67.23	66.99	67.08	44.76
Wdpos-mismatch	72.68	60.75	65.12	69.89	66.72	43.42
New evolved kernel	**76.95**	**72.20**	**73.51**	**75.89**	**74.57**	**56.05**

During the GP optimization experiments, the best results were obtained when using crossover and mutation ratios of 80% and 20%, respectively. We also noticed that the average fitness value converges after 50 GP generations. The average performance measures of each experiment were calculated and a comparison between the GP kernel and the single kernels were performed (Table 2).

The evolved kernel generated in this work outperforms all the tested single kernels in accuracy, sensitivity and specificity, fitness, and AUC.

Once the newly evolved kernel was constructed, it was possible to perform the ROC analysis and calculate the area under the curve for further comparing the performance of the different kernels.

It is worth mentioning that ROC analysis involved studying the sensitivity and specificity under different cut-off values. The ROC scores were calculated using the closest Euclidean distance between each sample and the SVM hyperplane. The best cut-off value for each experiment was considered in calculate new values for sensitivity, specificity and the area under the ROC as depicted in Figure 1. A summary of the area under the curve for each kernel is illustrated in Table 3.

Table 3. The area under curve for the conventional string kernels versus the newly evolved kernel.

Kernel	AUC
Spectrum	71.60
Weighted-Spectrum	74.80
Salzberg	59.50
Local-alignment	69.90
Fixed-degree	65.80
TOP	59.00
Locality-improved	68.30
Match-word	70.20
Polymatch	75.00
Weighted-degree	73.00
Wdpos-mismatch	69.70
Newly evolved kernel	**80.40**

5. Discussion and Conclusions

Currently, a large collection of string kernels is available for developing SVM classifiers. However, the major challenge is to select an optimal kernel for the problem of interest. Several interesting optimization approaches have been suggested to tackle this problem. In fact, most of the recent efforts were directed to the problems of numerical kernels and little attention has been paid to explore the power of evolutionary approaches in the field of string kernels [20].

In the present work, we have explored and evaluated a novel approach that uses GP to generate a superior evolved kernel form a set of conventional string kernels. The experimental results showed that the evolved kernel, when embedded with SVM, is capable of outperforming all the tested single kernels. Our model has the potential to discover new string kernels that can lead to better classification results. In addition, it allows interpreting the results to better understand the problem.

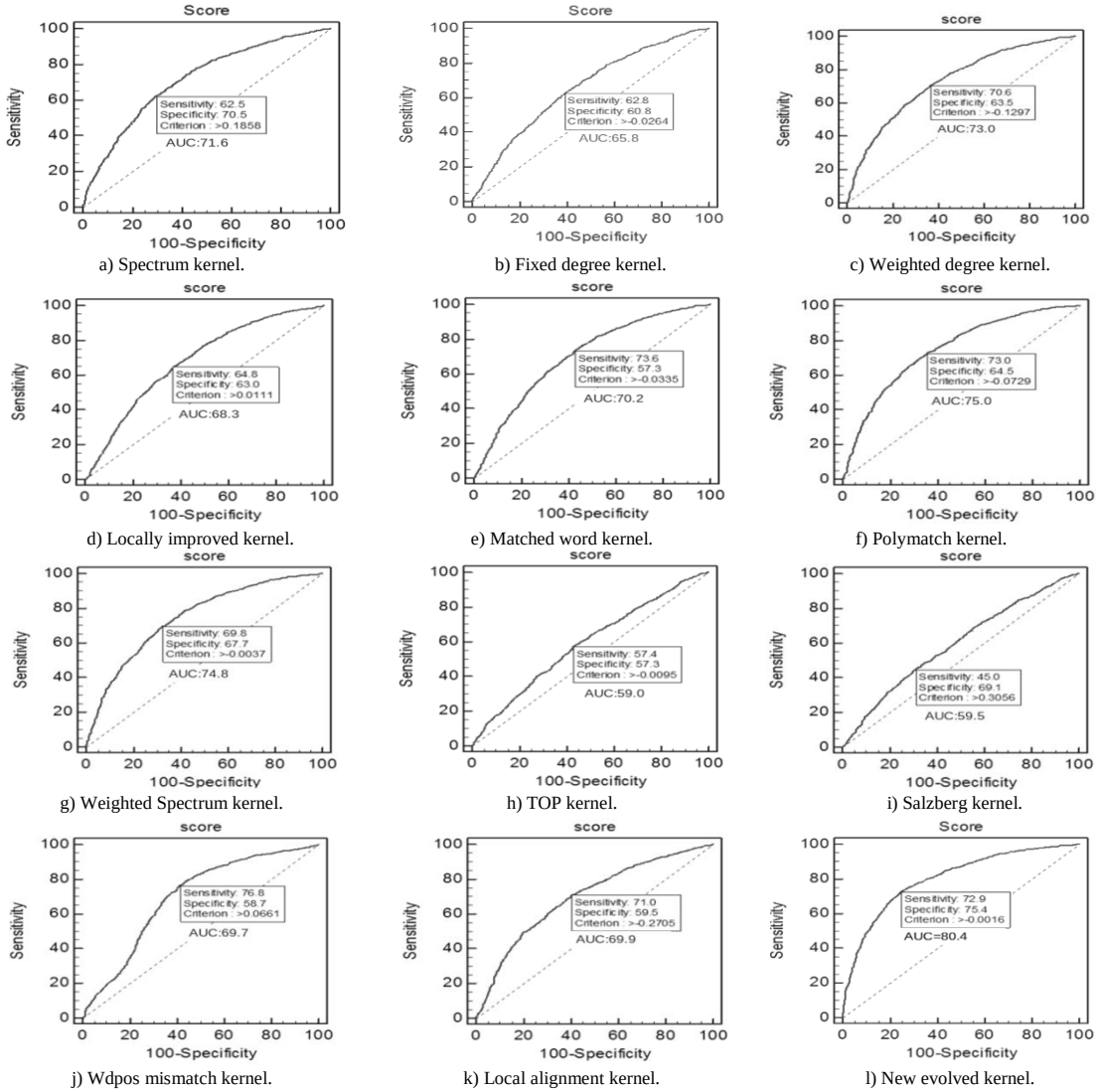

Figure 2. ROC curves of the conventional string kernels (a-k) versus the newly evolved kernel

In order to establish a proof of concept, we started with a group of 11 string kernels (Table 1) that were available at the time of developing the approach. However, we believe that including more string kernels will not only substantiate our approach, but it will enhance the classification performance since GP will start with a larger number and more diverse initial solutions.

Our evolved kernel reached an AUC of 80.4 for solving the MHC-II classification problem. However, some recently published reports showed a comparable or even a slightly better AUC for this specific problem using single string kernels [4, 7]. It is important to recall that the objective of using the MHC-II data in our work was to demonstrate the hypothesis that the evolved combined kernel generated by GP is superior to single string kernels and not to compare it with recently developed tools that were dedicated to the MHC-II prediction problem. Furthermore, it is worth mentioning that the improvements reported in these recent works were not solely dependent on the used string kernel. Nevertheless, enhancement of data encoding and employment of hybrid machine learning tools were in fact behind the improved performance [14].

The selection of the MHC-II problem was made because it has been considered a challenging string classification problem. However, the present approach can be applied to other problems of string nature such as text data and natural language processing.

In conclusion, we demonstrate that GP evolutionary approach is a good methodology to generate and optimize an enhanced evolve kernel from a collection of single string kernels so as to improve the performance of SVM classification

References

[1] Ahn H., Lee K., and Kim K., "Global Optimization of SVMs Using Genetic Algorithms for Bankruptcy Prediction," *in Proceedings of International Conference on Neural Information Processing*, Berlin, pp. 420-429, 2006.

[2] Ben-Hur A., Horn D., Siegelmann H., and Vapnik V., "Support Vector Clustering," *Journal of Machine Learning Research 2*, pp. 125-137, 2001.

[3] Bennet J., Ganaprakasam C., and Kumar N., "A Hybrid Approach for Gene Selection and Classification using Support Vector Machine," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 695-700, 2015.

[4] Bhasin M. and Raghava G., "SVM based Method for Predicting HLA-DRB1*0401 Binding Peptides in an Antigen Sequence," *Bioinformatics*, vol. 20, no. 3, pp. 421-423, 2004.

[5] Crammer K., Keshet J., and Singer Y., *Kernel Design Using Boosting*, MIT Press, 2002.

[6] Dioşan L., Rogozan A., and Pecuchet J., "Optimising Multiple Kernels for SVM by Genetic Programming," *in Proceedings of European Conference on Evolutionary Computation in Combinatorial Optimization*, Naples, pp. 230-241, 2008.

[7] Giguere S., Marchand M., Laviolette F., Drouin A., and Corbeil J., "Learning a Peptide-Protein Binding Affinity Predictor with Kernel Ridge Regression," *BMC Bioinformatics*, vol. 14, no. 82, pp. 2-16, 2013.

[8] Gönen M. and Alpaydın E., "Multiple Kernel Learning Algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211-2268, 2011.

[9] Gunn S., "Support Vector Machines for Classification and Regression," Technical Report, University oF Southampton, 1998.

[10] Howley T. and Madden M., "The Genetic Kernel Support Vector Machine: Description and Evaluation," *Artificial Intelligence Review*, vol. 24, no. 3-4, pp. 379-395, 2005.

[11] Li L., Rakitsch B., and Borgwardt K., "ccSVM: Correcting Support Vector Machines for Confounding Factors in Biological Data Classification," *Bioinformatics*, vol. 27, no. 13, pp. 342-348, 2011.

[12] Liao W. and Arthur J., "Predicting Peptide Binding to Major Histocompatibility Complex Molecules," *Autoimmunity Reviews*, vol. 10, no. 8, pp. 469-73, 2011.

[13] Mullan L., "Pairwise Sequence Alignment-It's All About Us!," *Brief Bioinform*, vol. 7, no. 1, pp. 113-115, 2006.

[14] Nielsen M., Lund O., Buus S., and Lundegaard C., "MHC Class II Epitope Predictive Algorithms," *Immunology*, vol. 130, no. 3, pp. 319-28, 2010.

[15] Scholkopf B. and Smola A., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press Cambridge, 2001.

[16] Scholkopf B., Tsuda K., and Vert J., *Kernel Methods in Computational Biology*, MIT Press, Cambridge, 2004.

[17] Silva S., "A Genetic Programming Toolbox for MATLAB," 2009.

[18] Sonnenburg S., Raetsch G., Henschel S., Widmer C., Behr J., Zien A., Bona F., Binder A., Gehl C., and Franc V., "The SHOGUN Machine Learning Toolbox," *Journal of Machine Learning Research*, pp. 1799-1802, 2010.

[19] Sonnenburg S., Rätsch G., Schäfer C., and Schölkopf B., "Large Scale Multiple Kernel Learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531-1565, 2006.

[20] Suykens J., Argyriou A., De Brabanter K., Diehl M., Pelckmans K., Signoretto M., Van Belle V., and Vandewalle J., "International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines: theory and applications (ROKS 2013)," *Book of Abstracts*, Leuven, pp. 128, 2013.

[21] Taylor J. and Cristianini N., *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[22] Zhang L., Udaka K., Mamitsuka H., and Zhu S., "Toward More Accurate Pan-Specific MHC-Peptide Binding Prediction: A Review of Current Methods and Tools," *Brief Bioinform*, vol. 13, no. 3, pp. 350-364, 2012.

Ruba Sultan received her MSc. in Informatics in 2012 from Palestine Polytechnic University, Palestine. She is currently working as a teaching and research assistant in the College of Information Technology and Computer Engineering in Palestine Polytechnic University. Her recent research focuses on developing computerized Bioinformatic tools.

Yaqoub Ashhab is an associate professor of molecular biology and bioinformatics in the Biotechnology Research Center at Palestine Polytechnic University and a visiting professor at the Autonomous University of Barcelona. His recent research focuses mainly on developing bioinformatic tools to improve classification of genomic and immunomic data related to host-pathogen interaction.

Hashem Tamimi received his Ph.D. in computer science from the University of Tubingen, Germany in 2006. Currently, he is an assistant professor at the College of Information Technology and Computer Engineering, Palestine, Polytechnic University, Hebron, Palestine. His research interests include machine learning and bioinformatics.