Prediction of O-Glycosylation Sites in Proteins using PSO-Based Data Balancing and Random Forest

Hebatallah A. Hassan[1], M. B. Abdelhalim[1], Amr Badr[2]

[1.]Department of Computer Science, College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
[2.]Department of Computer Science, Faculty of Computers and Information, Cairo University, Cairo, Egypt
hebaatef.ibrahim@gmail.com

Abstract: O-glycosylation of mammalian proteins is one of the most important post-translational modifications (PTMs). Hence, there is significant interest in the development of computational methods for reliable prediction of O-Glycosylation sites from amino acid sequences. One particular challenge in training the classifiers comes from the fact that the available dataset is highly imbalanced, which makes the classification performance for the minority class becomes unsatisfactory. Traditional sampling approaches generally rely on random re-sampling from a given dataset. However, these methods cannot utilize all the information available in the training set and it increases the false positive rate. This paper proposes a new approach for predicting the O-glycosylation sites which is based on Particle Swarm optimization (PSO) and Random Forest (RF). PSO is used as evolutionary under-sampling technique for balancing the dataset, and Random Forest is used as a classifier. The results obtained from the proposed approach and other related researches, demonstrate that the proposed approach outperforms the performance of other approaches for the experimented dataset.

[Hebatallah A. Hassan, M. B. Abdelhalim, Amr Badr. **Prediction of O-Glycosylation Sites in Proteins using PSO-Based Data Balancing and Random Forest.** *Life Sci J* 2014;11(12):1019-1025]. (ISSN:1097-8135). http://www.lifesciencesite.com. 175

Keywords: Post-translational modification; O-glycosylation; Imbalanced learning; PSO; Random Forest

1. Introduction

Glycosylation is one of the most common and complex post-translational modifications of proteins (PTMs) in eukaryotic cells, which involves the attachment of carbohydrate chains to amino acids in proteins. Glycosylated proteins (glycoproteins) play an important role in a number of biological processes [1]. Hence, they believed to occur in the development and progression of several diseases, such as Alzheimer's disease, cancer, autoimmune diseases, respiratory illness, diabetes and congenital disorders [2, 3]. Therefore, the prediction of glycosylation sites (glycosylated amino acid residues) in proteins is of great interest to biologists.

The experimental identification of glycosylation sites in proteins is expensive and laborious. That's why there is a significant interest in the development of computational methods for reliable prediction of glycosylation sites from amino acid sequences.

There are four types of protein glycosylation, N-linked glycosylation to the amide nitrogen of asparagine side chains, O-linked glycosylation to the hydroxyl of serine and threonine side chains, C-linked glycosylation to the tryptophan side chains and glycosylphosphatidylinositol (GPI). The prediction of O-linked glycosylation sites in proteins is a challenging problem because the O-linked glycosylation is not yet known to occur on any amino acid consensus sequence in eukaryotes (unlike the N-linked glycosylation) [4, 5]. Thus, we will focus only on predicting O-linked glycosylation protein sequence in this paper.

Several approaches to predict O-glycosylation sites have been reported, which are based on Artificial Neural Networks (ANNs) [6], PCA [7], Support Vector Machines (SVMs) [8-11], Random Forests (RFs) [12], and other machine learning techniques. The overall prediction accuracy achieved is about 70-85%. The most widely used glycosylation sites predictors are NetOGlyc and Oglyc [1]. NetOGlyc is reported to predict with accuracy over 83% [8], and Oglyc with a reported accuracy of 85% correctly classified instances [9], both are Support Vector Machine-based predictors.

Although work on predicting glycosylation sites exists in the literature, there is significant room for improvement. One particular challenge in training the classifiers, using standard machine learning algorithms, comes from the fact that the available dataset is highly imbalanced; the fraction of glycosylation sites (positive class) is relatively small compared to the fraction of non-glycosylation sites (negative class), which makes the classification performance for the positive class becomes unsatisfactory.

One solution for the imbalanced dataset problem on the prediction of glycosylation sites task was to apply an ensemble technique for resampling the original dataset where samples are randomly selected with replacement [13]. One of the main disadvantages of random resampling based approach is that it does not utilize additional information such as sample quality and their discriminative ability among classes, which could be useful in data classification.

In recent researches, Particle Swarm optimization (PSO) has been used as evolutionary under-sampling technique for identifying a subset of highly discriminative samples in the majority class for the imbalanced dataset problem in some bioinformatics applications [14-16], these researches showed that the PSO under-sampling approach can improve the quality of base classifiers and increase the classification accuracy.

In this paper, we proposed a new approach for predicting the O-glycosylation sites which is based on PSO and Random Forest. PSO is used as evolutionary under-sampling technique, and then the Random Forest is used as a classifier. The results obtained from the proposed approach and other related researches, demonstrate that the proposed approach outperforms the performance of other approaches on the problem of predicting O-glycosylation sites for the experimented dataset.

This paper is organized as follows: Section 2 explains the methods used for constructing the dataset, reviews the RF and the PSO algorithms, and it also explains the proposed approach which is based on the PSO and RF algorithms. In Section 3, several experimental results are presented and discussed. Finally, the conclusion is given in Section 4.

2. Methods

2.1 Dataset Construction

The dataset used in the experiments comes from O-GlycBase database [17], which contains experimentally verified glycosylation sites compiled from protein databases. The dataset has 242 glycoproteins from different spices. Each protein sequence in the dataset has http-linked cross-references to other protein sequence databases. In our study, the sequences that don’t have cross-reference to SWISS-PROT database were excluded, so 220 glycoproteins sequences were left. Out of the 220 sequences, 207 sequences have verified Serine or Threonine (S or T) sites. Those are the sequences that were used in our experiments.

The sequences were truncated by a sliding window (window size: *W*) into several sub-sequences to only include the verified O-glycosylation sites (serine/threonine) region windows, we used *W*=21 for our experiments as suggested by similar researches [6, 13]. This process is shown in the Figure 1.

We represent the sub-sequence which has S or T residue at the center and experimentally verified to be glycosylated as a positive instance. The sub-sequence that has S or T at the center but not annotated experimentally as being glycosylated is represented as negative instance. A dataset of 2091 positive and 11110 negative instances were obtained which has a class ratio of 0.188.

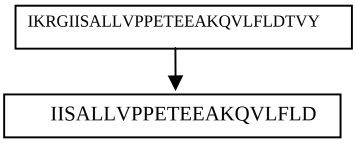

Figure 1. Sequence window of size 21

The protein sequences (excluding S or T at the center because only the sequence of the surrounded residues are needed to indicate whether the S or the T in the middle is glycosylated or not) with a length of *W*-1 are used for analysis. There are many methods for protein sequence coding, such as sparse coding, 5-letter coding, hydropathy coding and physical properties based coding. In this study, we used the sparse coding scheme for representation of the protein sequence as it has widely been used in similar researches [6, 7, 9]. The common 20 amino acids are coded by 20-D vectors only composed of 0 and 1 (the site of amino acid A is coded as 10000000000000000000, C is coded as 01000000000000000000 and so on). Thus the total length of coded sequence or dimension of sample vector is (*W*-1) * 20.

2.2 Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) is a robust stochastic optimization technique based on simulation of the social behavior of birds within a flock. PSO uses a number of particles that constitute for a swarm moving around in the search space looking for the best and randomly initialized. A swarm consists of *N* particles, where each particle represents a candidate solution moving around *D*-dimensional search space [18-21].

All of the particles have fitness values, which are evaluated by a fitness function to be optimized, and have velocities which direct the movement of the particles. During movement, the changes to a particle within the swarm are influenced by the experience, or knowledge, of its neighbors and each particle adjusts its position according to two fitness value, *pbest* and *gbest*. *pbest* is the personal best fitness value occurred during the history of a particle, whereas *gbest* constitutes the global best fitness value occurred during the history of the whole swarm.

Along all iterations, the particles are flying through search space and always accelerated towards better solutions. This process can be achieved by updating the velocity v of each particle i with Eq. (1):

$$v_i(t+1) = wv_i(t) + c_1 * r_1 * (pbest_i - x_i(t)) + c_2 * r_2 * (gbest - x_i(t)) \quad (1)$$

The first part of equation (1) represents the inertia of the previous velocity, the second part is the cognition part and it tells us about the personal thinking of the particle, the third part represents the cooperation among particles and is therefore named as the social component. Acceleration constants c_1, c_2 and inertia weight w are predefined by the user and r_1, r_1 are uniformly generated random numbers in the range of [0, 1].

In this paper, a binary version of a PSO algorithm (BPSO) is used for particle swarm optimization [22]. Each particle represents its position in binary values which are 0 or 1. In binary PSO the velocity of a particle defined as the probability that a particle might change its state to one. Each particle position is updated according to Eq. (2) and Eq. (3):

$$S(v_i(t+1)) = \frac{1}{1+e^{-v_i(t+1)}} \quad (2)$$

$$x_i(t+1) = \begin{cases} 1, & r_i < S(v_i(t+1)) \\ 0, & otherwise \end{cases} \quad (3)$$

Where r_i is a uniform random number in the range [0-1].

The basic process of the PSO algorithm can be presented as follows:

Step 1 – Initialization: Randomly create initial particles.

Step 2 – Fitness: Measure the fitness of each particle in the population.

Step 3 – Update: Calculate the velocity of each particle using Eq. (1).

Step 4 – Construction: For each particle, move to the next position according to Eq. (2) and Eq. (3).

Step 5 – Termination: Stop the algorithm if the termination criterion is satisfied; return to Step 2 otherwise.

PSO possesses the advantages such as high-performance and global optimization, which make it very popular in many biological related applications [14]. In this paper, the PSO is used to select the best possible samples from the majority class in order to construct a balanced dataset. In *Section* 2.4, the solution representation and the flowchart of PSO for subset selection is discussed.

2.3 Random Forest Classifier (RF)

Random forest classifier (RF) [23] is an ensemble classifier that consists of several decision trees. The output of this classifier is the class number that most frequently occurs individually in the output of decision trees classifiers. The main idea of decision trees is to predicate a target based on a group of input data. Decision trees also named classification trees, where the tree leaves represent the class labels and the branches represent the conjunction of feature vectors that lead to class labels.

RF is one of the state-of-the-art machine learning classifiers and has been used for a large number of biological problems [24]. One important advantage of RF is that it provides the importance information of each input variable, which is suitable for information retrieving from a dataset of high dimension with noise.

The forest error rate depends on two factors:

- The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate.
- The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

RF is considered as an appropriate model to handle large number of input dataset, imbalance, and due to its averaging strategy, RF classifier is robust to outliers and noise, avoids overfitting, and it is relatively fast, simple, and performs well in many classification problems.

2.4 The Proposed Approach

PSO has been used as a frequency ranking procedure, to detect the most useful subset of samples from the majority class that can be combined with the samples from the minority class so that the subset could best represent the decision boundary between the two classes.

The samples from majority class that are most frequently included in the optimized subsets are selected to match the number of minority samples to generate a balanced dataset, then the balanced dataset has been used for training a Random Forest classifier which is developed using WEKA [25], a widely-used machine learning work-bench in bioinformatics implemented in Java. Figure 2 illustrates the proposed approach.

PSO for best possible sample subset selection is constructed through the following procedures:

(1) Cross-validation: The *K*-fold cross-validation approach [26] is applied for a given dataset to partition the dataset into training sets (sampling sets) and test sets (evaluation sets). The training sets are used for sampling, while the testing sets are used for guiding the optimization process. We used a 3-fold cross-validation for our experiments similar to [14].

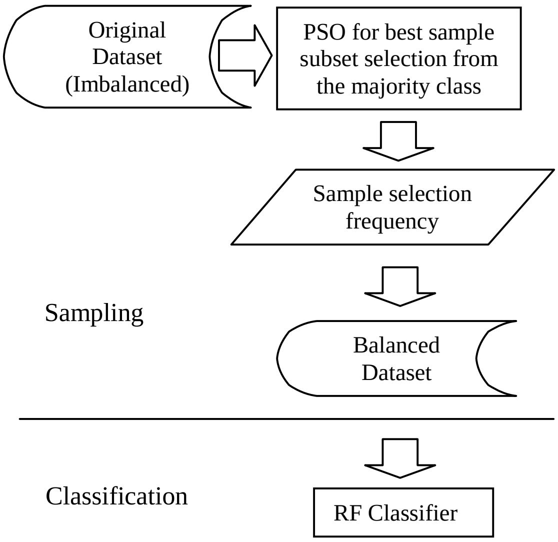
Original
Dataset
(Imbalanced)
PSO for best sample subset selection from the majority class
Sample selection frequency
Sampling
Balanced Dataset
Classification
RF Classifier

Figure 2. The Proposed Approach

(2) Solution representation: As shown in Figure 3, for each sample from the majority class, a dimension in the particle space is assigned. Assuming that we have *D* majority samples for a training fold, a particle i in PSO can be coded as an indicator function set P_i= {I_1, I_2... I_D}. for each dimension, an indicator function I_j takes value 1 when the corresponding *j*th sample is included to train a classifier. Similarly, a 0 denotes that the corresponding sample is excluded from training.

I_1	I_2	…………	I_D

I_j: Sample *j* is selected or not

Figure 3. Solution representation

(3) Sampling: As shown in Figure 4, the population of particles is initialized, each particle having a random position within the *D*-dimensional space and a random velocity for each dimension. Each particle's fitness is compared with the particle's best fitness and the global best fitness. If the particle's fitness is better than its best previous experience, its best previous experience is updated accordingly. Furthermore, if the particle's fitness is better than the global best fitness, the global best fitness is also updated. When the termination criterion is met, the selected samples from the last iteration are ranked by their selection frequency in the optimization process.

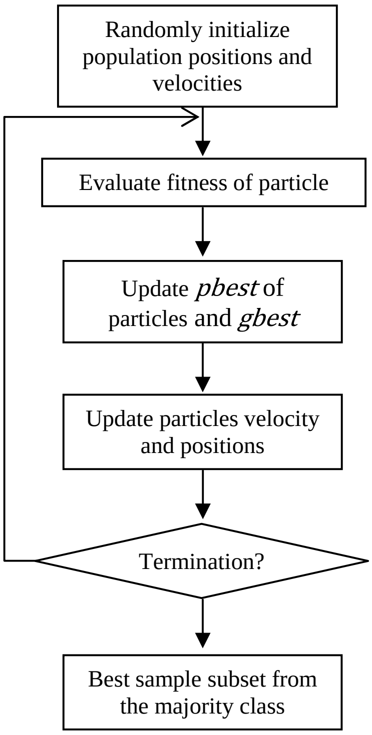
Randomly initialize population positions and velocities
Evaluate fitness of particle
Update pbest of particles and gbest
Update particles velocity and positions
Termination?
Best sample subset from the majority class

Figure 4. The flowchart of PSO for sample subset selection

(4) Fitness and evaluation metrics: The fitness of each particle here is a function of classification accuracy in terms of the area under the ROC Curve (AUC) [27]. The subsets that can create more accurate classification are favored and optimized in each PSO iteration. For each training subset generated the decision tree classifier (J48 implementation) is trained on the training subset and then tested on the corresponding test subset to obtain the classification accuracy. We used decision tree as a classifier for guiding the optimization process because it is sensitive to small perturbation on datasets.

Table 1.PSO setting parameters

Number of Particles	20
Maximum Iterations	100
Cognitive acceleration constant (c_1)	1.43
Social acceleration constant (c_2)	1.43
Inertia Weight (w)	0.689

(5) Termination criteria: If the number of iterations reaches the pre-determined maximum number of iterations, then the algorithm is terminated.

The parameters of PSO algorithm are summarized in Table 1. They have been used for solving similar problems [15] and they were found to give good results in our experiments.

The following example demonstrates the procedures of PSO for sample subset selection:

Suppose that the majority class has 3 samples, and the location of global best is (1, 0, 1), which means the first and the third samples are selected. The location of the particle's best of particle *i* is (0, 1, 1), which means the second and the third samples are used. The current position of particle *i* is (0, 0, 1), which means only the third sample is selected. The current velocity vector of particle *i* is assumed to be (0.98, 0.02, 0.51).

The velocity vector for the particle is updated using Eq. (1) as follows,

$$v_i(t+1) = 0.689 * \begin{bmatrix}0.98\\0.02\\0.51\end{bmatrix} + 1.43 * \begin{bmatrix}0.06\\0.99\\0.73\end{bmatrix} * \left(\begin{bmatrix}0\\1\\1\end{bmatrix} - \begin{bmatrix}0\\0\\1\end{bmatrix}\right) + 1.43 * \begin{bmatrix}0.81\\0.93\\0.85\end{bmatrix}\left(\begin{bmatrix}1\\0\\1\end{bmatrix} - \begin{bmatrix}0\\0\\1\end{bmatrix}\right) = \begin{bmatrix}1.83352\\1.42948\\0.35139\end{bmatrix}$$

Then using these new velocities, the new location of particle *i* can be obtained using Eq. (2) and Eq. (3),

$$S(v_i(t+1)) \approx \begin{bmatrix}0.86\\0.81\\0.59\end{bmatrix}$$

Supposing that $r_i = \begin{bmatrix}0.62\\0.01\\1.03\end{bmatrix}$,

Then, $x_i(t+1) = \begin{bmatrix}1\\1\\0\end{bmatrix}$

Now the location of the particle is moved to (1, 1, 0), meaning that the first and the second samples are selected. The J48 classifier is thus constructed using the selected samples from the majority class (according to the new positions), combined with all the samples from the minority class to construct the training dataset, and with the testing dataset of the fold, the classification accuracy in terms of AUC is calculated, which is the fitness value of particle *i*. In each iteration, the same procedure is applied for all particles. If the fitness of the *i*th particle is better than that particle's best fitness, then the position vector is saved for the particle best (*pbest*). If one of the particle's fitness is better than the global best fitness, then the position vector is saved for the global best (*gbest*). The procedure is iterated until termination condition is met. For each sample in the majority class, the frequency of including the sample in the optimized subsets is calculated, so that the samples that are most frequently included in the optimized subsets are selected to match the number of minority samples, in order to generate a balanced dataset.

3. Results and Discussion

We experimented the performance of the single Support Vector Machine (SVM) and the Random Forest (RF) classifiers, as well as the ensembles of SVM on the original (imbalanced) dataset, as these methods are the most widely used for the O-Glycosylation sites prediction. We also evaluated the performance of the SVM and RF classifiers based on the balanced dataset which is sampled using the PSO evolutionary under-sampling technique, (PSO+SVM) and (PSO+RF) as shown in Table 2.

WEKA has been chosen as the data mining system in this analysis because of its user-friendliness. For the SVM models, we used the LibSVM [28] classifier with the radial basis function (RBF) kernel for the classification, the other parameters were fixed at the default values. For evaluating the ensemble methods, we experimented the Bagging[29] and the AdaBoost[30] classifiers with LibSVM as base classifier, using 5 and 10 iterations. And we experimented the RF classifier based on 100, 200 and 500 as number of trees. The experiments were performed on Intel(R) Core(TM) i5-4200U CPU @ 1.60 GHz 2.30GHz computer, with 4GB of RAM.

We evaluated the performances of the classifiers in terms of AUC based on 10-fold cross-validation of our datasets (for both cases balanced and imbalanced) in order to avoid over fitting. 10-fold cross-validation is selected as it is the most common for machine learning applications [31]. Also the classifier building time is reported in seconds for each method. The results of the experiments indicate that:

- The RF classifier in general gives better performance in terms of AUC over the SVM and the ensembles of the SVM for the problem of predicting the O-Glycosylation sites.
- The SVM classifier didn't give better accuracy over the ensembles of SVMs when the dataset was resampled using the PSO as an evolutionary under-sampling technique.
- RF classifiers gives higher accuracy when the dataset is resampled using PSO as an evolutionary under-sampling technique.
- The classifiers building time enhances when the database is sampled using PSO.

Table 2.Performance comparison between different classification methods and the proposed PSO+RF approach

Methods	Num. Iterations	Num. Trees	Building Time	AUC
SVM [28]	-	-	18.04	0.815
Bagging [29]	5 10	-	98.03 187.62	0.8255 0.8286
AdaBoost [30]	5 10	-	793.01 1681.64	0.9027 0.9065
RF	-	100 200 500	26.01 54.87 142.45	0.933 0.9469 0.9507
PSO+SVM	-	-	3.62	0.8357
PSO+RF	**-**	100 200 500	7.79 15.18 43.4	**0.947** **0.9497** **0.9511**

4. Conclusion

This study proposes a new approach for predicting the O-Glycosylation sites based on PSO and Random Forest. PSO is used to select the best possible sample subset from a majority class in order to enhance the Random Forest classifier results. Comparison of the obtained results with those of other approaches demonstrates that the proposed approach has higher classification accuracy in terms of AUC than other tested approaches. That is, the proposed approach can be applied to remove unnecessary samples and further enhancing the overall classification results.

Corresponding Author:
Hebatallah A. Hassan
Department of Computer Science
Arab Academy for Science, Technology and Maritime Transport
Cairo, Egypt
E-mail: hebaatef.ibrahim@gmail.com

References

1. Julenius K, Johansen MB, Zhang Y, Brunak S, Gupta R. Prediction of Glycosylation Sites in Proteins. In: Lieth CW, Luetteke T, Frank M, ed. Bioinformatics for glycobiology and glycomics: an introduction. John Wiley & Sons, Ltd. Chichester, UK. 2009:163-92.
2. Dwek R. Biological importance of glycosylation. Dev Biol Stand 1998;96:43-47.
3. Haltiwanger R, Lowe J. Role of Glycosylation in Development. Annual Review of Biochemistry 2004;73:491-537.
4. Wilson B, Gavel Y, von Heijne G. Amino acid distributions around Olinked glycosylation sites. Biochem J., 1991;275:529-34.
5. Christlet THT, Veluraja K. Database analysis of O-glycosylation sites in proteins. Biophys J., 2001;80:952-60.
6. Yang Xue-mei. Prediction of Protein O-Glycosylation Sites by Kernel Principal Component Analysis and Artificial Neural Network. Communications in Computer and Information Science, 2011;176:445-50.
7. Yang Xue-mei, Cui Xue-wei, Yang Xue-zhu. Predictionof O-Glycosylation Sites in Protein Sequence by Kernel Principal Component Analysis. Computational Aspects of Social Networks (CASoN) 2010:267-70.
8. Steentoft C, Vakhrushev SY, Joshi HJ, Kong Y, Vester-Christensen MB, Schjoldager KT, Lavrsen K, Dabelsteen S, Pedersen NB, Marcos-Silva L, Gupta R, Bennett EP, Mandel U, Brunak S, Wandall HH, Levery SB, Clausen H. Precision mapping of the human O-GalNAc glycoproteome through Simple Cell technology. EMBO J, 2013;32(10):1478-88.
9. Li S, Liu B, Zeng R, Cai Y, Li Y. Predicting O-glycosylation sites in mammalian proteins by using SVMs. ComputBiolChem., 2006; 30(3): 203-8.
10. Chauhan JS, Rao A, Raghava GPS. *In silico* Platform for Prediction of N-, O- and C-Glycosites in Eukaryotic Protein Sequences. PLoS ONE 2013;8(6):e67008.
11. Torii M, Liu H, Hu Z-Z. Support Vector Machine-Based Mucin-Type O-linked Glycosylation Site Prediction Using Enhanced Sequence Feature Encoding. AMIA Annual Symposium Proceedings, 2009:640-4.
12. Hamby SE, Hirst JD. Prediction of glycosylation sites using random forests. BMC Bioinformatics 2008;9:500.
13. Caragea C, Sinapov J, Silvescu A, Dobbs D, Vasant Honavar V. Glycosylation site prediction using ensembles of Support Vector Machine classifiers. BMC Bioinformatics, 2007;8:438.
14. Yang P, Xu L, Zhou BB, Zhang Z, Zomaya AY. A particle swarm based hybrid system for imbalanced medical data sampling. BMC Genomics 2009;10(Suppl 3):S34.
15. Yang P, Zhang Z, Zhou BB, Zomaya AY. Sample subsets optimization for classifying imbalanced biological data. Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2011: 333-44.

16. Yang P, Yoo PD, Fernando J, Zhou BB, Zhang Z, Zomaya AY. Sample Subset Optimization Techniques for Imbalanced and Ensemble Learning Problems in Bioinformatics Applications. IEEE Trans Cybern 2013.
17. Gupta R, Birch H, Rapacki K, Brunak S, Hansen JE. O-GLYCBASE version 4.0: a revised database of O-glycosylated proteins. Nucleic Acids Research 1999;27(1):370-2.
18. Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of IEEE Conference on Neural Network 1995;4:1942-8.
19. Kennedy J. The Particle Swarm: Social Adaptation of Knowledge. IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ., 1997:303-8.
20. Shi Y, Eberhart RC. Particle Swarm Optimization: developments, Applications and Resources. IEEE International Conference on Evolutionary Computation, 2001;1:81-6.
21. Shi Y, Eberhart RC. A Modified Particle Swarm Optimizer. Evolutionary Computation, 1998:69-73.
22. Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. IEEE International Conference on Systems, Man, and Cybernetics 1997.
23. Breiman L. Random Forests. Machine Learning 2001;45:5-32.
24. Qi Y. Random Forest for Bioinformatics. In: Cha Z, Yunqian M, ed. Ensemble Machine Learning. Springer US. 2012:307-23.
25. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I.H. The WEKA Data Mining Software: An Update SIGKDD Explorations 2009;11(1):10-18.
26. Salzberg SL. On comparing classifiers: pitfalls to avoid and a recommended approach. Data Mining and Knowledge Discovery 1997; 1(3): 317-28.
27. Bradley A. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 1997; 30(7): 1145-59.
28. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011;2(3).
29. Breiman L. Bagging predictors. Machine Learning, 1996;24(2):123-40.
30. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. Annals of Statistics 1998;28:337-407.
31. Payam R, Lei T, Huan L. Cross-Validation. In: Tamer Özsu M, Liu L, ed. Encyclopedia of Database Systems 2009:532-8.

12/23/2014