Prediction of O-glycosylation Sites Using Random Forest and GA-Tuned PSO Technique

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

[1]Department of Computer Science, College of Computing and Information Technology, Arab Academy for Science and Technology and Maritime Transport (AASTMT), Cairo, Egypt. [2]Department of Computer Science, Faculty of Computers and Information, Cairo University, Cairo, Egypt.

ABSTRACT: O-glycosylation is one of the main types of the mammalian protein glycosylation; it occurs on the particular site of serine (S) or threonine (T). Several O-glycosylation site predictors have been developed. However, a need to get even better prediction tools remains. One challenge in training the classifiers is that the available datasets are highly imbalanced, which makes the classification accuracy for the minority class to become unsatisfactory. In our previous work, we have proposed a new classification approach, which is based on particle swarm optimization (PSO) and random forest (RF); this approach has considered the imbalanced dataset problem. The PSO parameters setting in the training process impacts the classification accuracy. Thus, in this paper, we perform parameters optimization for the PSO algorithm, based on genetic algorithm, in order to increase the classification accuracy. Our proposed genetic algorithm-based approach has shown better performance in terms of area under the receiver operating characteristic curve against existing predictors. In addition, we implemented a glycosylation predictor tool based on that approach, and we demonstrated that this tool could successfully identify candidate glycosylation sites in case study protein.

KEYWORDS: O-glycosylation, imbalanced learning, genetic algorithm, PSO, random forest

CITATION: Hassan et al. Prediction of O-glycosylation Sites Using Random Forest and GA-Tuned PSO Technique. *Bioinformatics and Biology Insights* 2015:9 103–109 doi: 10.4137/BBI.S26864.

RECEIVED: April 02, 2015. **RESUBMITTED:** May 24, 2015. **ACCEPTED FOR PUBLICATION:** May 26, 2015.

ACADEMIC EDITOR: Thomas Dandekar, Associate Editor

TYPE: Original Research

FUNDING: Authors disclose no funding sources.

COMPETING INTERESTS: Authors disclose no potential conflicts of interest.

CORRESPONDENCE: hebaatef.ibrahim@gmail.com

Introduction

Glycosylation is one of the most important and complex post-translational modification of protein in eukaryotic cells.[1,2]; it is crucial to many molecular functions, such as structure, biological activity, and protein–protein interaction.[2] More than 50% of the proteins are estimated to be glycosylated. O-glycosylation and N-glycosylation are the two main types of mammalian protein glycosylation. O-glycosylation is an enzyme-directed process that links a carbohydrate to the hydroxyl group of serine (S) and threonine (T) amino acid residues in proteins. Unlike N-glycosylation, the O-linked glycosylation is not yet identified to occur on any amino acid consensus sequence.[3] Thus, computational prediction of O-glycosylation sites in mammalian proteins is challenging and has received considerable attention.

Several classifiers for predicting the O-glycosylation sites in proteins have been proposed,[4] such as EnsembleGly[5], which used the ensembles of support vector machine (SVM) classifiers for predicting the O-glycosylation sites and achieved an area under the receiver operating characteristic (ROC) curve (AUC) value of 0.91. CKSAAP_OGlySite,[6] a SVM-based approach, employed a sequence-encoding scheme based on the composition of *k*-spaced amino acid pairs (CKSAAP) to predict O-linked glycosylation. GPP[7] adopted the random forest (RF) method and integrated frequencies of amino acids surrounding modified residue and significant pairwise patterns for predicting the glycosylation sites. NetOGlyc[8] is a SVM classifier based on sequence context and surface accessibility. Recently, GlycoEP[9] is a SVM-based tool that reported an accuracy of 86.9% for O-glycosylation sites prediction.

One challenge in training the classifiers is that the available datasets are highly imbalanced, which makes the classification accuracy for the minority class to become unsatisfactory. In our previous work,[10] we have proposed an approach for predicting the O-glycosylation sites, which is based on particle swarm optimization (PSO) and RF; we call it PSO + RF. PSO was used as an evolutionary undersampling technique for balancing the dataset, and RF was used as a classifier. We have chosen the PSO as an evolutionary undersampling technique over the random resampling or ensembles based techniques, because the evolutionary based techniques prove to achieve a good trade-off between data reduction and the accuracy of the classification. In addition, different classifiers were experimented to be trained on the PSO undersampled dataset; the RF classifier gave the best results among all.

In this study, we enhance our previous method by implementing parameters optimization/tuning mechanism, for optimizing the PSO parameters based on genetic algorithm (GA) in order to increase the classification accuracy. The results show that this GA-based approach significantly improved the prediction performance in terms of AUC compared with our previous approach and with other existing prediction tools.

This paper is organized as follows: Methods section describes the protein sequence data and its encoding, then it reviews our previous PSO-based O-glycosylation prediction technique; it also describes the basic GA concepts, and it discusses the proposed GA-based parameters optimization approach. In the Results and Discussion section, the results are presented and discussed. Finally, the conclusion and future work are given in the last section.

Methods

Protein sequence data and encoding. Similar to our previous study,[10] the protein sequence data used in this paper come from O-GlycBase[11] (Version 6.00), which contains experimentally verified glycosylation sites compiled from protein databases. O-GlycBase database has 242 glycoproteins from different spices. In our study, the sequences that do not have http-linked cross-reference to SWISS-PROT[12] database were excluded, and only the sequences that have verified serine or threonine (S or T) glycosylation sites were used in our experiments.

The sequences were truncated by a sliding window (window size: W) into several subsequences in order to obtain the verified O-glycosylation sites (S/T) region windows. In this work, we used $W = 15$, so the target residue is located at position 8. This choice of window size was based on previous works.[7,13] We represent the subsequence that has S or T residue at the center and experimentally verified to be glycosylated as a positive instance. The subsequence that has S or T at the center and not annotated as being glycosylated is represented as negative instance. To prevent overestimation of the predictive performance, the positive and the negative datasets were further filtered by a 30% identity cut off similar to Li and Wang[14] and Caragea et al.[15] using CD-HIT,[16] means two glycosylated protein sequences with >30% identity were defined as homologous sequences. The nonhomologous negative data were generated using the same approach as the positive one. Considering the middle residue in each fragment is always the same (S/T), the central position is excluded when calculating the sequence identity. After filtration, we obtained a dataset of 2118 positive and 11,266 negative instances, which has a class ratio of 0.188.

The protein sequences with a length of $W - 1$ are used for analysis (excluding S or T at the center; the sequence of the surrounded residues indicates whether the S or the T in the center is glycosylated or not). We used the sparse coding scheme for representing the protein sequence similar to Li et al.[17] and Cruz-Cano et al.[18] The 20 amino acids are coded by 20-D vectors composed of 0 and 1 (for example, the amino acid A is coded as 10000000000000000000 and C is coded as 01000000000000000000). Thus, the total length of coded sequence or dimension of sample vector is $(W - 1) * 20$.

RF classifier. RF classifier[19] is an ensemble classifier that constructs multiple decision trees with randomly selected features. The final classification is obtained by combining the classification results from the individual decision trees. Combining multiple trees produced in randomly selected subspaces can improve the generalization accuracy.

RF shows a significant performance improvement over the single tree classifiers. Owing to its averaging approach, RF classifier is robust to outliers and noise; it avoids over-fitting, is relatively fast and simple, and it performs well in many classification problems.

Ensemble methods, including RF, bagging, and boosting, have been increasingly applied in bioinformatics. When compared to bagging and boosting ensemble methods, RF has a unique advantage of using multiple feature subsets, which is well suited for high-dimensional data as demonstrated by several bioinformatics studies.[20,21]

PSO for undersampling. In our previous study, binary particle swarm optimization (BPSO)[22] has been used as a frequency ranking procedure, to detect the most useful subset of samples from the majority class of imbalanced dataset, 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 of the O-glycosylation classification problem.

For each sample from the majority class, a dimension in the particle space is assigned. For each dimension, an indicator function takes value 1 when the corresponding sample is included to train a classifier. Similarly, a 0 denotes that the corresponding sample is excluded from training. The fitness of each particle is a function of classification accuracy in terms of the AUC.[23]

The subsets from the majority class that can create more accurate classification are favored and optimized in each PSO iteration. Samples from the last PSO iteration are ranked by their selection frequency in the optimization process. The samples from the 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. See Supplementary Files Imbalanced.arff and PsoUndersampled.arff for imbalanced and PSO undersampled datasets, respectively. The balanced dataset has been used for training a RF.[19] classifier implemented in WEKA,[24] a widely used machine learning workbench in bioinformatics implemented in Java. Figure 1 summarizes the PSO for undersampling technique.

The PSO parameters used in our previous study are shown in Table 1. They were chosen according to similar problems, and they were found to give good results in our experiments. However, in this study, we optimize these

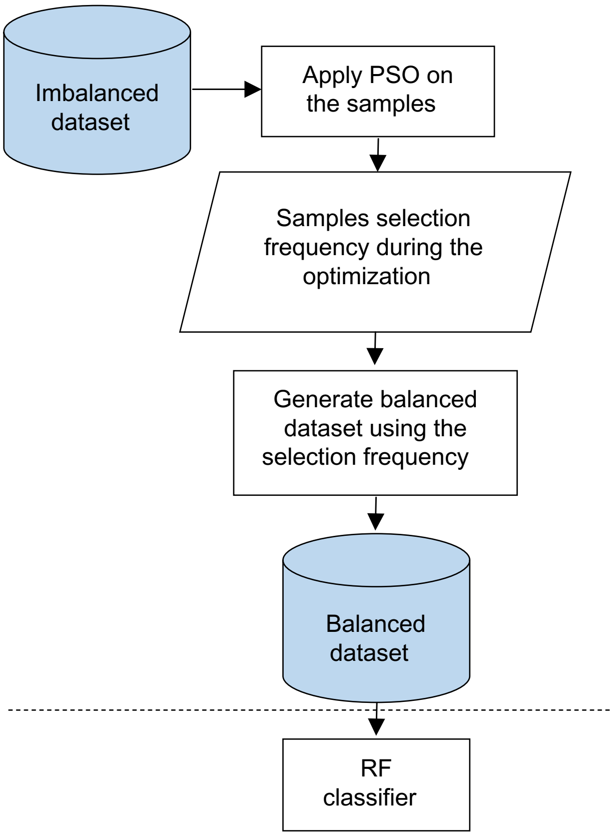

Imbalanced dataset
Apply PSO on the samples
Samples selection frequency during the optimization
Generate balanced dataset using the selection frequency
Balanced dataset
RF classifier

Figure 1. PSO for undersampling.

parameters using GA in order to even produce higher classification results.

Genetic algorithm. GA is a heuristic search and optimization technique based on the principles of natural selection and genetics in biological systems. The bases of GA were proposed by Holland.[25] GA has been used to solve a wide range of problems of significant complexity. It works with a set of candidate solutions called a population, and each potential solution in the population is known as chromosome.

Each chromosome is composed of a sequence of genes. These genes represent different aspects of the solution just like in individuals in the nature; each gene in the chromosome represents different aspects like hair color, eye color, etc. GA obtains the optimal solution after a series of iterative computations. GA generates successive populations of alternative solutions until acceptable results are obtained. GA can deal with large search spaces efficiently, and hence, it has less chance to get local optimal solution than other techniques.

Table 1. PSO parameters setting.

PARAMETER	VALUE
Number of particles	20
Cognitive acceleration (C_1)	1.43
Social acceleration (C_2)	1.43
Inertia weight (W)	0.689

Starting with an initial population that is generated by random individuals, the population is evolved for a number of generations while gradually improving the qualities of the individuals. A fitness function assesses the quality of a solution in the evaluation step. The crossover and mutation functions are the main operators that randomly affect the fitness value. Chromosomes are selected for reproduction by evaluating the fitness value. The fitter the chromosomes are, the higher the probability to be selected.

Figure 2 illustrates the GA evolutionary cycle. Crossover allows new solutions in the search space to be explored; it is a random mechanism for exchanging genes between two chromosomes using the one-point crossover, two-point crossover, or homolog crossover. In mutation, the genes may occasionally be altered, ie, in binary code genes, changing genes code from 1 to 0 or vice versa.[26,27] Offspring replaces the old population using the elitism or diversity replacement strategy and forms a new population in the next generation. The evolutionary process operates many generations until the termination condition is satisfied.

GA-based PSO optimization. In our previous study,[10] we used PSO for undersampling the O-glycosylation sites dataset by selecting the most important samples from the majority class. Proper setting for the PSO algorithm parameters can improve the algorithm performance, and consequently, the classification accuracy. In this paper, we use the GA for optimizing the parameters of the PSO. As shown in Figure 3, the PSO-based undersampling technique (the dashed part) is integrated with the GA. Each GA solution (chromosome) represents different alternatives for the values of the PSO parameters to be used in the sample selection from the majority class step. GA obtains the optimized PSO parameters after a series of iterative GA operations (crossover and mutation). Based on the balanced dataset, the classification accuracy using the RF classifier is used for evaluating the fitness of each GA solution. The GA terminates after exceeding the maximum number of generations.

The Java Genetic Algorithms Package (JGAP)[28] was chosen as the programming platform for implementing the GA-based optimization approach; JGAP is a GA, and Genetic Programing open source framework is written in

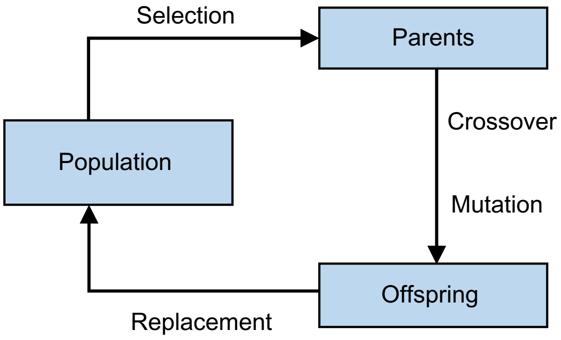


Figure 2. GA evolutionary cycle.

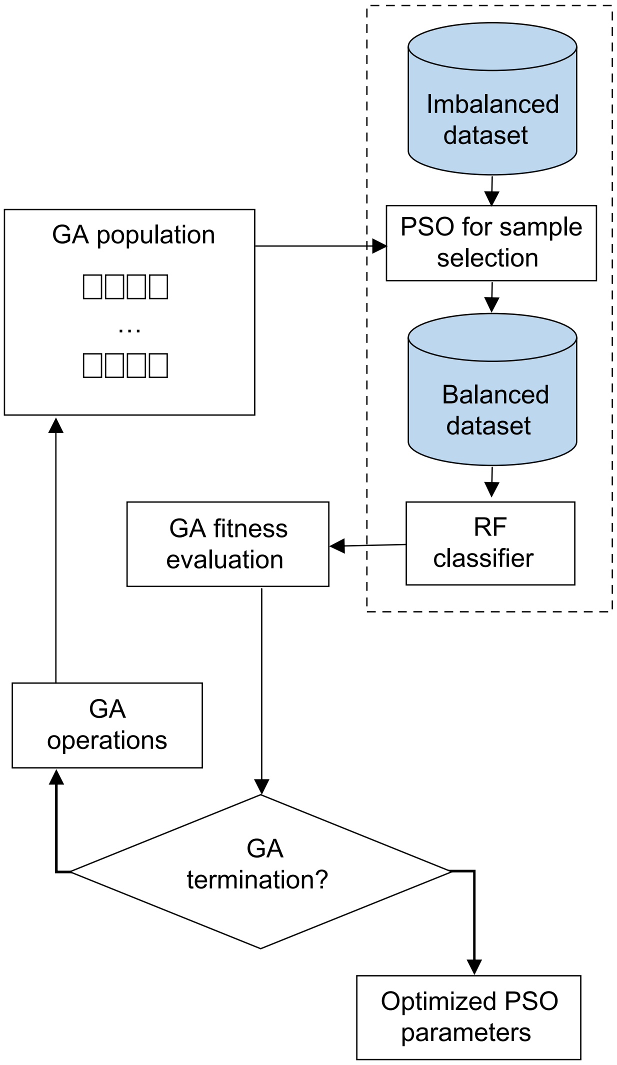


Figure 3. GA-based PSO optimization.

Java. It is designed to be flexible and modular. It is possible to create specific chromosomes, genetic operators, natural selection, and others. To support these possibilities, JGAP uses a configuration object. Setting the configuration object with all these new definitions, prior to running the genetic search is the first task. It is necessary to provide the following information: how the chromosomes are set, what fitness function will be used, and how many chromosomes create a population. In our experiments, we used the Best Chromosome selection criterion (retaining the fittest chromosome or configuration for next iteration) implemented in JGAP. We used a population of 60 individuals to be randomly generated in each generation. A maximum of 20 generations was configured. We applied the default crossover technique implemented in the JGAP tool, which is population size/2 (which means 0.5), and we employed 0.06 as the mutation rate.

Chromosome design. The PSO parameters: number of particles, inertia weight (W), cognitive acceleration constant (C_1), and social acceleration constant (C_2) should be optimized. Therefore, the chromosome comprises four genes as shown in Table 2. We set the range of each gene according to the values used in the literatures.

Fitness function. The classification accuracy in terms of AUC for the RF classifier, shown in Figure 3, is the criteria used to design a fitness function for the GA. Thus, the individual (chromosome) with high classification accuracy produces a high fitness value. The chromosome with high fitness value has high probability to be preserved to the next generation.

Results and Discussion

GA-tuned PSO + RF results and comparisons. In order to reduce the computation time, we used a subset of the O-glycosylation dataset that is described in Protein sequence data and encoding section, considering to have the same ratio of the minority class to the majority class (1:5), for applying the proposed GA-based PSO optimization technique. The resulted PSO optimized values are listed in Table 3. We used these parameters with the PSO for undersampling mechanism that we have proposed in our previous study,[10] in order to undersample the full dataset. See Supplementary File Tuned-PsoUndersampled.arff for the GA-tuned PSO undersampled dataset.

We built a predictor based on the RF classifier implemented in WEKA and trained on the undersampled dataset after the PSO optimization; we call it GA-tuned PSO + RF. The predictor is implemented in Java and Perl; it is available in the Supplementary Files. We evaluated the performance of the predictor based on 10-fold stratified cross-validation.[29] The experiments were performed on Intel(R) Core(TM) i5–4200U CPU @ 1.60 GHz 2.30 GHz computer, with 4 GB of RAM.

We calculated four different measures for evaluating the GA-tuned PSO + RF classifier performance: sensitivity, specificity, accuracy, and Matthews correlation coefficient (MCC). Sensitivity assesses the effectiveness of classifying the positive samples. Specificity assesses the accuracy of clas-

Table 2. Chromosome representation.

GENE	TYPE	RANGE
Num. of particles	Integer	15–25
W	Double	0.4–0.9
C_1	Double	0.5–2.0
C_2	Double	0.5–2.0

Table 3. Optimized PSO parameters.

GENE	OPTIMIZED VALUE
W	0.4
C_1	0.75
C_2	1.6
Number of particles	20

sifying the negative samples. Accuracy assesses the effectiveness of classifying both the positive and the negative samples. MCC is also a measure of accuracy designed to take into account the ability of a classifier to classify correctly both positive and negative instances. It produces a value between −1 and 1, with 1 being a perfect prediction and −1 a completely incorrect prediction.

The four measurements are expressed in terms of true positive (TP), false negative (FN), true negative (TN), and false positive (FP) predictions. Each measurement is given as following:

$$Sensitivity = \frac{TP}{TP+FN} \tag{1}$$

$$Specificity = \frac{TP}{TP+FP} \tag{2}$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{3}$$

And,

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}} \tag{4}$$

The prediction accuracy was also measured by using the AUC. The ROC.[30] curve is an effective method for evaluating the performance of the prediction system. It is commonly defined as a plot of sensitivity on the *Y*-axis versus the FP rate on the *X*-axis. The bigger the AUC is, the better the overall prediction system performance is.

We compared the results of the GA-tuned PSO + RF classifier with our previous PSO + RF classifier after making refinement for the PSO + RF classifier code, where some minor mistakes are corrected and sequence homology cutoff is applied. As shown in Table 4, the GA-tuned PSO + RF predictor has higher performance in terms of sensitivity, specificity, accuracy, MCC, and AUC. We performed statistical significance tests between the two classifiers using the paired *t*-tests (corrected), provided by WEKA, and all the differences in accuracy were found to be statistically significant at confidence level $P = 0.95$. Hence, this proves the superiority of the proposed approach compared with our previous one.

We also compared the prediction results of the proposed approach with the results of EnsembleGly, GPP, and GlycoEP predictors reported by the tools and by the comparison reported by Chauhan et al.[9]

We have used the same dataset as that used in EnsembleGly and GPP for the training and the evaluation processes. GlycoEP is also comparable with our classifier as it also used sequence-based nonredundant dataset for training the classifier. As given in Table 4, the sensitivity and AUC have significantly improved using our proposed method. The other measures have slightly decreased, but this is reasonable as we decreased the number of negative samples from the dataset during the undersampling process.

In general, the AUC is a statistically consistent and more discriminating measure than accuracy for evaluating the binary datasets[31]; the accuracy rate of the predictive model is not a good indicator when there is imbalanced problem, as a result of the fact that it will be biased toward the majority class. Our approach reached AUC value of 0.942; based on this value, we can state that our approach is more efficient for the prediction of the O-glycosylation sites than the other approaches.

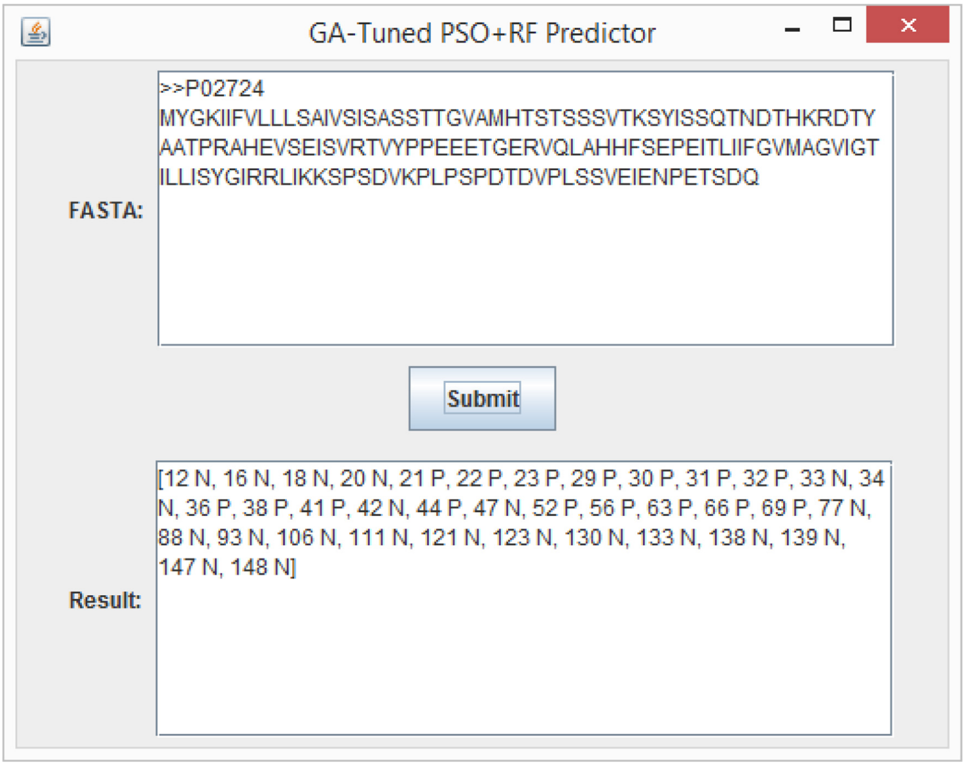
Case study. To further illustrate the performance of the GA-tuned PSO + RF predictor, we performed a case study of one protein extracted from UniProt[32] benchmark database. The protein was glycophorin-A (GYPA, UniProt ID: P02724), a major intrinsic membrane protein with a high proportion of O-glycosylated residues in erythrocytes. It has 16 experimentally verified O-linked glycosylation sites. Our predictor could predict all of those sites (labeled with P as shown in Fig. 4). These results suggest that GA-tuned PSO + RF predictor can be a useful tool for in silico glycosylation site prediction.

Conclusion and Future Work

This study proposes a GA-based approach for predicting the O-glycosylation sites in proteins. The O-glycosylation dataset is highly imbalanced which affects the classification accuracy. In our previous work, we used PSO as an evolutionary undersampling technique, in order to select the best possible sample subset from the majority class. Proper setting for the PSO algorithm parameters can improve the algorithm performance. In this study, we optimized the PSO parameters using GA in order to improve the PSO undersampling technique performance and consequently the O-glycosylation sites classification accuracy. We compared the proposed technique with our

Table 4. Comparing performances of existing techniques with our technique.

METHOD	SENSITIVITY	SPECIFICITY	ACCURACY	MCC	AUC
EnsembleGly[5]	68.0%	64.0%	89.0%	0.59	0.91
GPP[7]	94.9%	**90.7%**	91.4%	**0.83**	0.897
GlycoEP[9]	89.37%	88.82%	**91.89%**	**0.83**	0.783
PSO+RF[10]	96.5%	73.4%	85.4%	0.73	0.94
GA-Tuned PSO+RF	**97%**	73.7%	85.7%	0.735	**0.942**



FASTA:

```
>>P02724
MYGKIIFVLLLSAIVSISASSTTGVAMHTSTSSSVTKSYISSQTNDTHKRDTY
AATPRAHEVSEISVRTVYPPEEETGERVQLAHHFSEPEITLIIFGVMAGVIGT
ILLISYGIRRLIKKSPSDVKPLPSPDTDVPLSSVEIENPETSDQ
```

Result:

```
[12 N, 16 N, 18 N, 20 N, 21 P, 22 P, 23 P, 29 P, 30 P, 31 P, 32 P, 33 N, 34
N, 36 P, 38 P, 41 P, 42 N, 44 P, 47 N, 52 P, 56 P, 63 P, 66 P, 69 P, 77 N,
88 N, 93 N, 106 N, 111 N, 121 N, 123 N, 130 N, 133 N, 138 N, 139 N,
147 N, 148 N]
```

Figure 4. Case study using glycoprotein P02724.

previous one (before optimizing the PSO parameters), and we also compared it with other several existing techniques. The obtained results demonstrate that the proposed technique is more efficient. We achieved higher classification accuracy in terms of AUC with comparison to other techniques. In addition, we demonstrated that the proposed technique could identify O-glycosylation sites using a case study protein.

Some of the scopes of further enhancement are given below:

- In our study, we predict the amino acid residues that are likely to be glycosylated using the information derived from the target amino acid residue and its sequence neighbors. Other features such as the physical properties of amino acids can be combined with the sequence information for predicting the O-glycosylation sites. A future study is to consider other amino acid features while predicting the O-glycosylation sites in proteins based on the GA-tuned evolutionary undersampling approach that we have proposed.
- Another future study is to analyze the effect of other protein sequence coding techniques such as five letter coding, hydropathy coding, and physical properties based coding, in combination with our proposed GA-tuned PSO + RF approach.

Author Contributions

Conceived and designed the experiments: HH. Analyzed the data: HH. Wrote the first draft of the manuscript: HH, AB, MBA. Contributed to the writing of the manuscript: HH, AB, MBA. Agree with manuscript results and conclusions: HH, AB, MBA. Jointly developed the structure and arguments for the paper: HH, AB, MBA. Made critical revisions and approved final version: HH, AB, MBA. All authors reviewed and approved of the final manuscript.

Supplementary Material

Imbalanced. Multiple sequence alignment relation

PsoUndersampled.

TunedPsoUndersampled.

REFERENCES

1. Haltiwanger RS, Lowe JB. Role of glycosylation in development. *Annu Rev Biochem*. 2004;73:491–537.
2. Jensen ON. Interpreting the protein language using proteomics. *Nat Rev Mol Cell Biol*. 2006;7(6):391–403.
3. Wilson IB, Gavel Y, von Heijne G. Amino acid distributions around O-linked glycosylation sites. *Biochem J*. 1991;275(pt 2):529–34.
4. Joshi H, Gupta R. Eukaryotic glycosylation: online methods for site prediction on protein sequences. In: Lütteke T, Frank M, eds. *Glycoinformatics SE – 9. Methods in Molecular Biology*. Vol. 1273. New York: Springer; 2015:127–37.
5. Caragea C, Sinapov J, Silvescu A, Dobbs D, Honavar V. Glycosylation site prediction using ensembles of support vector machine classifiers. *BMC Bioinformatics*. 2007;8:438.
6. Chen Y-Z, Tang Y-R, Sheng Z-Y, Zhang Z. Prediction of mucin-type O-glycosylation sites in mammalian proteins using the composition of k-spaced amino acid pairs. *BMC Bioinformatics*. 2008;9:101.
7. Hamby SE, Hirst JD. Prediction of glycosylation sites using random forests. *BMC Bioinformatics*. 2008;9:500.
8. Steentoft C, Vakhrushev SY, Joshi HJ, et al. Precision mapping of the human O-GalNAc glycoproteome through SimpleCell technology. *EMBO J*. 2013;32(10):1478–88.
9. 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):1–10.

10. Hassan H, Abdelhalim MB, Badr A. Prediction of O-glycosylation sites in proteins using PSO-based data balancing and random forest. *Life Sci J*. 2014;11(12):1019–25.
11. Gupta R, Birch H, Rapacki K, Brunak S, Hansen JE. O-GLYCBASE version 4.0: a revised database of O-glycosylated proteins. *Nucleic Acids Res*. 1999;27(1):370–2.
12. Bairoch A, Apweiler R. The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Res*. 1999;27(1):49–54.
13. Trost B, Kusalik A. Computational prediction of eukaryotic phosphorylation sites. *Bioinformatics*. 2011;27(21):2927–35.
14. Li J, Wang W. Grouping of amino acids and recognition of protein structurally conserved regions by reduced alphabets of amino acids. *Sci China C Life Sci*. 2007;50(3):392–402.
15. Caragea C, Sinapov J, Honavar V, Dobbs D. Assessing the performance of macromolecular sequence classifiers. In: Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2007). Boston, MA: IEEE; 2007:320–6.
16. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*. 2012;28(23):3150–2.
17. Li S, Liu B, Zeng R, Cai Y, Li Y. Predicting O-glycosylation sites in mammalian proteins by using SVMs. *Comput Biol Chem*. 2006;30(3):203–8.
18. Cruz-Cano R, Lee M-LT, Leung M-Y. Logic minimization and rule extraction for identification of functional sites in molecular sequences. *BioData Min*. 2012;5(1):10.
19. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32.
20. Yang P, Hwa Yang Y, Zhou BB, Zomaya AY. A review of ensemble methods in bioinformatics. *Curr Bioinform*. 2010;5(4):296–308.
21. Qi Y. *Random Forest for Bioinformatics*. Berlin: Springer; 2012:1–18.
22. Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. In: International Conference on Systems, Man, and Cybernetics. Orlando, FL: IEEE; 1997:5.
23. Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit*. 1997;30(7):1145–59.
24. Hall M, National H, Frank E, et al. The WEKA data mining software: an update. *SIGKDD Explor*. 2009;11(1):10–8.
25. Holland JH. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press; 1975.
26. Goldberg DE. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc; 1989.
27. Sivanandam SN, Deepa SN. *Introduction to Genetic Algorithms*. Berlin: Springer; 2008.
28. Meffert K, Meseguer J, Marti E D, Meskauskas A, Vos J, Rotstan N. JGAP – Java genetic algorithms and genetic programming package. Available at: http://jgap.sourceforge.net/. (Last visited, June 2015).
29. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 2. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1995:1137–43. Available at: http://dl.acm.org/citation.cfm?id = 1643031.1643047.
30. Fawcett T. An introduction to ROC analysis. *Pattern Recognit Lett*. 2006; 27(8):861–74.
31. Huang J, Ling CX. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans Knowl Data Eng*. 2005;17(3):299–310.
32. Apweiler R, Bairoch A, Wu CH, et al. UniProt: the universal protein knowledgebase. *Nucleic Acids Res*. 2004;32(Database issue):D115–9.