J. Microbiol. Biotechnol. (2012), **22**(8), 1054–1058
http://dx.doi.org/10.4014/jmb.1203.03050
First published online April 27, 2012
pISSN 1017-7825 eISSN 1738-8872

JOURNAL OF MICROBIOLOGY AND BIOTECHNOLOGY
© The Korean Society for Microbiology and Biotechnology

Algorithm for Predicting Functionally Equivalent Proteins from BLAST and HMMER Searches

Yu, Dong Su[1,2], Dae-Hee Lee[1], Seong Keun Kim[1], Choong Hoon Lee[1], Ju Yeon Song[1], Eun Bae Kong[2]*, and Jihyun F. Kim[1,3]*

[1]*Systems and Synthetic Biology Research Center, Korea Research Institute of Bioscience and Biotechnology, Daejeon 305-806, Korea*
[2]*Department of Computer Science and Engineering, Chungnam National University, Daejeon 305-764, Korea*
[3]*Department of Systems Biology, Yonsei University, Seoul 120-749, Korea*

Received: March 21, 2012 / Revised: March 31, 2012 / Accepted: April 1, 2012

In order to predict biologically significant attributes such as function from protein sequences, searching against large databases for homologous proteins is a common practice. In particular, BLAST and HMMER are widely used in a variety of biological fields. However, sequence-homologous proteins determined by BLAST and proteins having the same domains predicted by HMMER are not always functionally equivalent, even though their sequences are aligning with high similarity. Thus, accurate assignment of functionally equivalent proteins from aligned sequences remains a challenge in bioinformatics. We have developed the FEP-BH algorithm to predict functionally equivalent proteins from protein–protein pairs identified by BLAST and from protein–domain pairs predicted by HMMER. When examined against domain classes of the Pfam-A seed database, FEP-BH showed 71.53% accuracy, whereas BLAST and HMMER were 57.72% and 36.62%, respectively. We expect that the FEP-BH algorithm will be effective in predicting functionally equivalent proteins from BLAST and HMMER outputs and will also suit biologists who want to search out functionally equivalent proteins from among sequence-homologous proteins.

Keywords: Functionally equivalent protein, error back-propagation algorithm, sequence-based method, artificial neural network, bioinformatics

Given that high-throughput sequences are generated in exponential quantities by next-generation or third-generation sequencing technologies with state-of-the-art computing resources [13, 26], many biologists utilize bioinformatics tools to predict the functions of unknown protein sequences. Specifically, BLAST [1] and HMMER [4] are widely employed for carrying out biological research tasks, such as pathway prediction, genome annotation, and phylogenetic analysis, all of which depend on fast computational performance and reliable accuracy [2, 6, 10–11].

BLAST searches out highly aligned subsequence regions in a pair of sequences, using substitution matrices such as BLOSUM and PAM [1]. Like BLAST, HMMER also identifies conserved regions, but it uses profiles for the patterns of conserved domains. These profiles are built with a hidden Markov model (HMM) [4], with the Needleman–Wunsch algorithm [20], or with the Smith–Waterman algorithm [23]. Both BLAST and HMMER are based on the concept that functional domains of proteins are conserved in the protein sequence, and both report the degree of sequence similarity for aligned sequence pairs by using numerical values such as percent identity, e-value, and matching score. These numerical values have been used as thresholds for predicting functionally similar proteins.

Equivalogs are defined as members of a set of homologous proteins that are conserved with respect to function since their last common ancestor [7]. However, not all well-aligned and high-scoring pairs are functionally equivalent proteins (FEPs), and some of them may be false-positive or false-negative FEPs [24]. Analysis of protein–protein pairs with BLAST and protein–domain pairs with HMMER has revealed that some FEPs have a relatively low matching score and low percent identity in BLAST (Fig. 1A and 1B), with HMMER giving similar results (Fig. 1C and 1D). They indicate that patterns of sequence conservation may vary, despite functional equivalence. It is

*Corresponding author
J.F. K.
Phone: +82-2-2123-5561; Fax: +82-2-312-5657;
E-mail: jfk1@yonsei.ac.kr
E.B. K.
Phone: +82-42-821-6656; Fax: +82-42-822-9959;
E-mail: keb0202@gmail.com

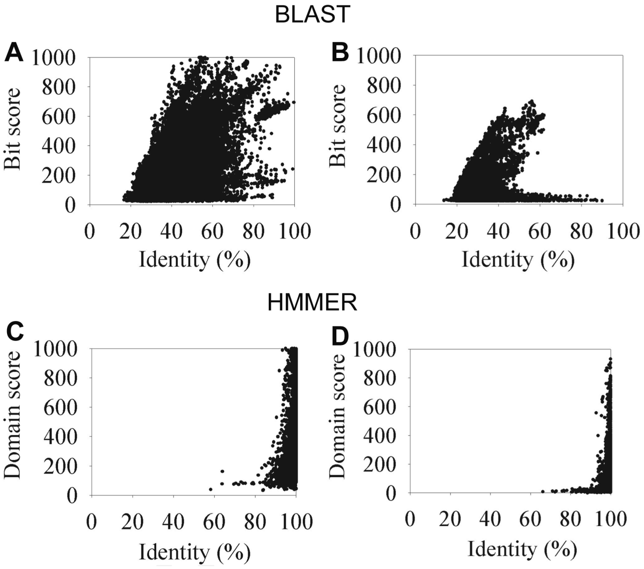

Fig. 1. Distribution of functionally equivalent protein (FEP) pairs (**A** and **C**) and non-FEP pairs (**B** and **D**) predicted from protein–protein pairs generated by BLAST and from protein–domain pairs generated by HMMER against 5,278 proteins of equivalogs in TIGRFAMs.

x-Axis, percent identity (BLAST and HMMER) for the aligned region (HMMER); y-axis, bit score (BLAST) or domain score (HMMER).

difficult to determine a uniform threshold among numerical values, making many biologists use just one of them [17, 22]. However, the numerical values collectively indicate the degree of sequence similarity. Accordingly, we can postulate that these values are correlated to one another and that, if the correlation among numerical values is determined based on a training set of FEPs, the determined correlation should predict FEPs from BLAST and HMMER.

On the basis of this assumption, we have designed the FEP-BH algorithm to predict FEPs from protein–protein pairs generated by BLAST and from protein–domain pairs generated by HMMER. FEP-BH uses patterns among the numerical values of the BLAST and HMMER outputs, and determines the types of correlation using the error back-propagation (EBP) algorithm [21] with the equivalogs from TIGRFAMs [7].

Materials and Methods

EBP Algorithm and EBP-Trained Weight Matrices

FEP-BH uses the EBP algorithm as a preprocessing step to establish the patterns of correlation among the numerical values for protein–protein pairs from BLAST and for protein–domain pairs from HMMER. EBP has been used for a variety of analyses, including clustering and pattern recognition [8, 16]. In biological fields, the EBP algorithm has also been widely used to predict biological information, such as protein solubility, protein–protein interactions, and prokaryotic transcription terminators [14, 18, 19]. Although the EBP algorithm requires many iterations and extensive training time, we have used it to train the weight matrices with the mixed and challenging data sets produced by BLAST and HMMER, because the EBP algorithm is well-suited for imbalanced data sets [21].

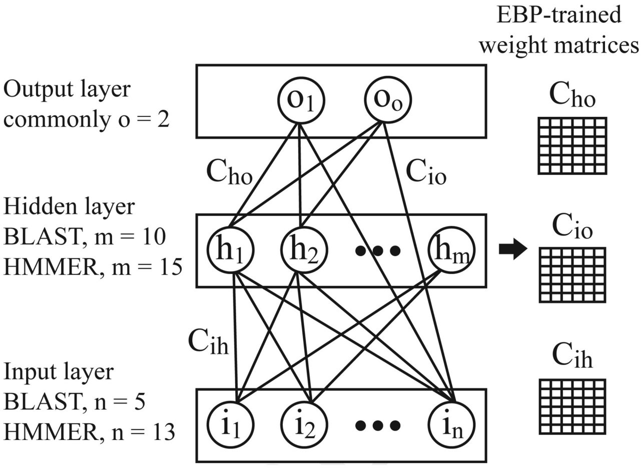

Fig. 2. Architecture of the error back-propagation (EBP) algorithm.

C_{xy} is the weight matrix for the edge from x to y, with the layers denoted as follows: i, input; h, hidden; and o, output.

In our algorithm, EBP was built using a bridged multilayer perceptron (BMLP) topology [25] and it generates EBP-trained weight matrices as the pattern to distinguish FEPs among the numerical values (Fig. 2). When sets of BLAST or HMMER pairs are separately used for training with the EBP algorithm, 5 numerical values of BLAST (percent identity, number of mismatches, bit score, coverage, and e-value) are used to train the matrices as input perceptrons for BLAST, whereas 13 values of numerical values of HMMER are used to train the matrices for HMMER, including sequence score, sequence bias, domain score, identity, coverage, e-value, c-evalue, and i-evalue. All input perceptron values are normalized to a range from 0 to 1, because the values otherwise have different ranges. In the FEP-BH algorithm, EBP-trained weight matrices are used to assign a pair to 1 of 3 classes: FEP, non-FEP (assigned if the pair is not in the same equivalog group), and candidate (undetermined whether the pair is a FEP or not). Although some of non-FEPs can be the potential FEPs as members of a superfamily, we did not consider them as equivalogs.

FEP-BH Algorithm

The FEP-BH algorithm follows a simple progression. First, it predicts FEPs, non-FEPs, and candidates from BLAST or HMMER pairs, using EBP-trained weight matrices. Second, the algorithm merges the FEPs, non-FEPs, and candidates from BLAST with those from HMMER. That is, the non-FEPs and the candidates assigned with the EBP-trained weight matrices for BLAST are reassigned to one of the classes determined by the matrices for HMMER. Finally, the merged FEPs, non-FEPs, and candidates are output.

Since the training is completed as preprocessing, FEP-BH does not retrain with each set of protein–protein (or –domain) pairs from BLAST and HMMER. Therefore, if the EBP-trained weight matrices are already constructed, FEP-BH can perform quickly to predict FEPs and non-FEPs from unknown protein–protein pairs and protein–domain pairs. In addition, FEP-BH does not require

Table 1. Training and test data sets.

Count	Training data set		Test data set	
Classes	1,291		2,999	
Proteins	5,278		39,854	
Pairs	BLAST	HMMER	BLAST	HMMER
Total	103,749	15,436	3,024,627	124,943
FEPs	42,288	5,277	1,745,878	45,760
Non-FEPs	61,461	10,159	1,278,749	79,183

any threshold values for coverage, percent identity, bit score, or e-value in order to predict FEPs, because EBP-trained weight matrices are performed as the threshold and do not require additional experimental information for sequences or other tasks, such as matching EC numbers or protein names or using another database; it can reduce the effort required to predict FEPs.

Training and Test Data Set

The choice of training data sets for the EBP-trained weight matrices for BLAST and HMMER is important to the accuracy of the FEP predictions, which rely on the matrices. Therefore, a training data set should be exactly defined about whether paired proteins are FEPs of each other or not and also should include many FEPs and non-FEPs with various qualities, in order to increase the accuracy of training and to avoid overfitting and local optimization. In this sense, equivalogs from TIGRFAMs are quite suitable as components of the training data set, because equivalogs are sets of proteins that are homologous with respect to conserved function since the last common ancestor [7]. We selected 1,291 equivalogs from TIGRFAMs (release version 11.0), composed of 5,278 Swiss-Prot protein sequences [7, 12], and they were used to construct the training data sets. The training data set for BLAST consisted of 103,749 protein pairs, which were generated by “all vs. all” blastp in BLAST and selected with the highest bit score among the same protein–protein pairs. The training data set for HMMER consisted of 15,436 protein–domain pairs, which were generated by hmmsearch in HMMER against the selected TIGRFAMs HMM profiles and selected by the same way of BLAST (Table 1).

During 100 rounds of weight matrix training for every equivalog using leave-one-out cross-validation (LOOCV) across the training data set [3], the average mean squared error (MSE) tended toward stability at around 0.14 for BLAST and 0.05 for HMMER (Fig. 3). Although local optimization with the EBP algorithm was not overcome within 100 rounds, we were able to achieve reliable EBP-trained weight matrices for which the average MSE for the training data set was stable at around 0.069 (BLAST) and 0.037 (HMMER). Independent of the training data sets, we constructed a test data set using the Pfam-A seed database [5]. The purpose of the test data sets is to examine the accuracy of FEP predictions for unknown pairs from BLAST and HMMER. The Pfam-A seed database is composed of curated classes, domains, and families, and it provides profiles for HMMER. We selected 39,854 Swiss-Prot proteins in 2,999 domain classes from the Pfam-A seed database (release version 26.0), and both the test data sets for BLAST and HMMER consisted of 3,024,627 protein–protein pairs generated by “all vs. all” blastp and 124,943 protein–domain pairs generated by hmmsearch against the Pfam-A-derived HMM profiles, respectively (Table 1).

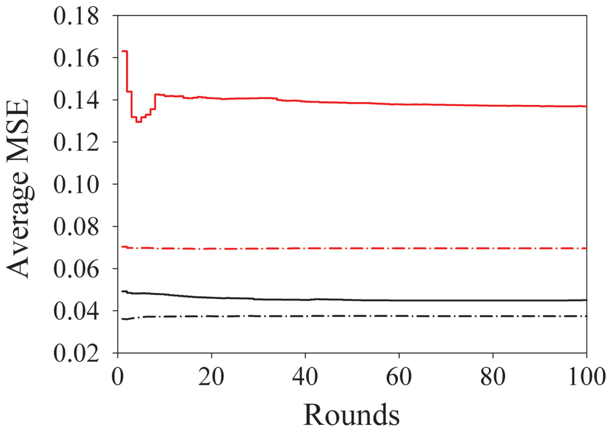

Fig. 3. Average mean squared error (MSE) for validation of the EBP algorithm.

Red, BLAST; black, HMMER; solid, LOOCV; dash-dots, the training data set.

RESULTS

When examining the training data set, the FEP-BH algorithm showed 93.75% accuracy, whereas the accuracy values for BLAST and HMMER were 40.76% and 34.19%, respectively, when no threshold was used for any numerical value (Table 2). These results indicate FEP-BH to be well-trained and well-performing. However, they do not ensure the accuracy of FEP-BH for unknown proteins because FEP-BH uses EBP-trained weight matrices trained with the training data set. Thus, we tried to examine the accuracy of the algorithm on a test data set. As with the training data set, FEP-BH showed strong performance, with 71.53% accuracy, whereas BLAST was 57.72% and HMMER was 36.62% without any threshold.

Furthermore, we tried comparing FEP-BH with a combination of BLAST and HMMER. Since FEP-BH merges the FEPs, non-FEPs, and candidates of BLAST and HMMER, we can expect that the combination of BLAST and HMMER will also have higher accuracy than either BLAST or HMMER alone. In addition, we applied a bit-score cut-off as a threshold to select the best FEP predictions from BLAST alone and from a combination of BLAST and HMMER. As shown in Fig. 4, FEP-BH performed better than either BLAST alone or the combination of BLAST and HMMER on both the training data set (Fig. 4A) and the test data set (Fig. 4B). The accuracy was far better for FEP-BH than for BLAST or for the combination of BLAST and HMMER, using bit-score thresholds ranging from 0 to 200. Moreover, even though BLAST and the BLAST–HMMER combination were maximally 67.93% and 69.57% accurate at a bit-score threshold of 50 in the test data set, their accuracies were below that of FEP-BH. When the bit score increased over 200, FEP-BH, BLAST, and the combination of BLAST

Table 2. Prediction of FEP against the training and test data sets.

Statistics[a]	Training data set			Test data set		
	FEP-BH	BLAST[b]	HMMER[b]	FEP-BH	BLAST[b]	HMMER[b]
TP	41,601	42,288	5,277	1,319,342	1,745,878	45,760
TN[c]	55,608	0	0	822,040	0	0
FP	5,853	61,461	10,159	447,983	1,278,749	79,183
FN[c]	628	0	0	404,292	0	0
Candidate	59	0	0	30,970	0	0
Sensitivity	0.9851	1.0000	1.0000	0.7654	1.0000	1.0000
Specificity	0.9048	0.0000	0.0000	0.6473	0.0000	0.0000
Accuracy	0.9375	0.4076	0.3419	0.7153	0.5772	0.3662

[a] TP, true positive; TN, true negative; FP, false positive; FN, false negative; Sensitivity = TP/(TP + FN); Specificity = TN/(TN + FP); Accuracy = (TP + TN)/(TP + TN + FP + FN).

[b] These methods were preformed without any threshold.

[c] Since the training and test data sets are composed only of pairs from BLAST and HMMER, TN and FN from BLAST and HMMER are zero.

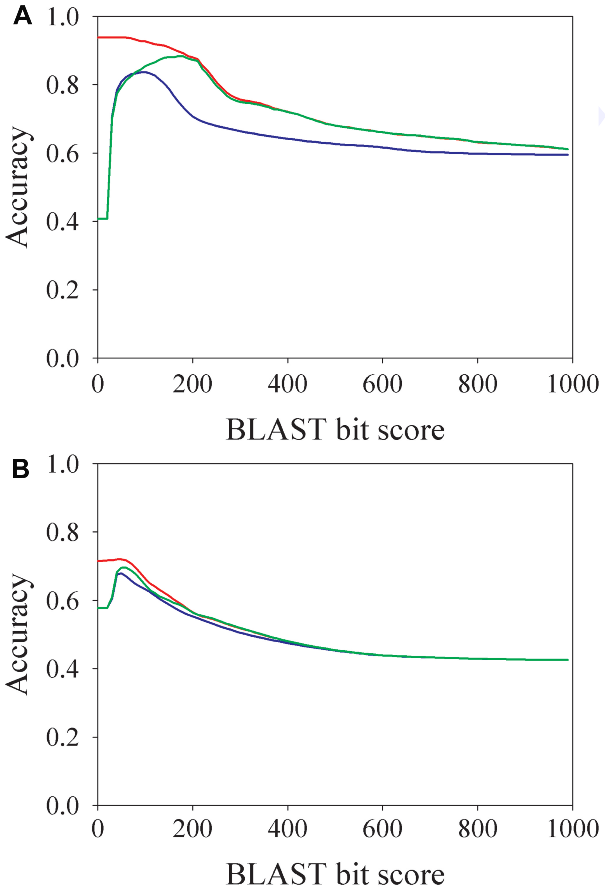

Fig. 4. Accuracy for the FEP-BH algorithm (red), BLAST (blue), and a combination of BLAST and HMMER (green) run against the training (**A**) and test (**B**) data sets.

FEP-BH does not require any threshold, but BLAST alone and the combination of BLAST–HMMER may require thresholds to identify a maximum number of FEPs. Therefore, in order to examine FEP-BH as well as to find the best threshold for BLAST or BLAST–HMMER in combination, we arbitrarily applied a BLAST bit score as a threshold. HMMER was not compared with FEP-BH because the output format of HMMER is different from that of BLAST.

and HMMER converged similarly for both the training and the test data sets. Although FEP-BH performs better than the other methods, it showed 71.53% accuracy for the Pfam-A seed database, which is lower than the one for the training data set and BLAST (Table 2). In a strict sense, Pfam-A seed is not a database for FEPs but contains clusters of proteins with the same domain. Therefore, some of the Pfam-A seed proteins in the same domain class may be non-FEPs, and a lot of proteins have multiple domains.

Discussion

Predicting FEPs using sequence-based methods such as BLAST and HMMER is an important challenge in the fields of biology and bioinformatics, because currently available sequence-based methods only suggest homologous proteins. We have developed the FEP-BH algorithm to predict FEPs from protein–protein pairs generated by BLAST and from protein–domain pairs generated by HMMER. The FEP-BH algorithm uses weight matrices trained against numerical values of BLAST and HMMER using the EBP algorithm [21]. Although the numerical values are mathematically correlated (*e.g.*, bit score and e-value), their values are not always in parallel. The EBP algorithm can be suitable to determine the pattern among unparallel numerical values because of training against various data sets. FEP-BH does not require experimental evidence or numerical value thresholds for FEP prediction. Therefore, preprocessed FEP-BH can perform efficiently, by reducing the otherwise substantial effort required to predict FEPs.

We searched for other applications to compare with our algorithm and identified some utilities related to FEPs, namely FOSTA [15] and Visalign [9]. However, neither FOSTA nor Visalign is suitable for predicting FEPs from unknown proteins because FOSTA uses BLAST with a text-mining approach to compare the prefix names of proteins, the EC numbers, and the product names, and Visalign is intended to analyze the aligned pattern among FEPs, not to predict FEPs. Therefore, we were unable to compare FEP-BH with them.

Many biologists want to predict function from protein sequences through homology searches. Since FEP-BH can be applied to a variety of biological fields in which BLAST and HMMER are used, we expect that FEP-BH can be an application that fulfills biologists' demand of effectively and accurately finding functionally equivalent proteins among homologous sequences.

Acknowledgments

We thank Sung Ho Yoon and Min-Jung Kwak for helpful discussions and critique. This work was supported by the National Research Foundation of Korea (2011-0017670), the Global Frontier Intelligent Synthetic Biology Center (2011-0031952), and the 21C Frontier Microbial Genomics and Applications Center of the Ministry of Education, Science and Technology, Republic of Korea.

REFERENCES

1. Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. *J. Mol. Biol.* **215:** 403–410.
2. Caspi, R., T. Altman, K. Dreher, C. A. Fulcher, P. Subhraveti, I. M. Keseler, *et al.* 2012. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.* **40:** D742–D753.
3. Deb, K. and A. Raji Reddy. 2003. Reliable classification of two-class cancer data using evolutionary algorithms. *Biosystems* **72:** 111–129.
4. Eddy, S. R. 1998. Profile hidden Markov models. *Bioinformatics* **14:** 755–763.
5. Finn, R. D., J. Mistry, J. Tate, P. Coggill, A. Heger, J. E. Pollington, *et al.* 2010. The Pfam protein families database. *Nucleic Acids Res.* **38:** D211–D222.
6. Fischer, S., B. P. Brunk, F. Chen, X. Gao, O. S. Harb, J. B. Iodice, *et al*. 2011. Using OrthoMCL to assign proteins to OrthoMCL-DB groups or to cluster proteomes into new ortholog groups. *Curr. Protoc. Bioinformatics* **35:** 6.12.1–6.12.19.
7. Haft, D. H., B. J. Loftus, D. L. Richardson, F. Yang, J. A. Eisen, I. T. Paulsen, and O. White. 2001. TIGRFAMs: A protein family resource for the functional identification of proteins. *Nucleic Acids Res.* **29:** 41–43.
8. Karlik, B., M. O. Tokhi, and M. Alci. 2003. A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis. *IEEE Trans. Biomed. Eng.* **50:** 1255–1261.
9. Keim, D. A., D. Oelke, R. Truman, and K. Neuhaus. 2006. Finding correlations in functionally equivalent proteins by integrating automated and visual data exploration, pp. 183–192. *In: Proceedings of the Sixth IEEE Symposium on BioInformatics and BioEngineering*, 16–18 October 2006. IEEE Computer Society Washington, DU, USA.
10. Koski, L. B., M. W. Gray, B. F. Lang, and G. Burger. 2005. AutoFACT: An automatic functional annotation and classification tool. *BMC Bioinformatics* **6:** 151.
11. Ludwig, W., O. Strunk, R. Westram, L. Richter, H. Meier, Yadhukumar, *et al.* 2004. ARB: A software environment for sequence data. *Nucleic Acids Res.* **32:** 1363–1371.
12. Magrane, M. and U. Consortium. 2011. UniProt Knowledgebase: A hub of integrated protein data. *Database (Oxford)* **2011:** bar009.
13. Mardis, E. R. 2008. The impact of next-generation sequencing technology on genetics. *Trends Genet.* **24:** 133–141.
14. Ma, Z., C. Zhou, L. Lu, Y. Ma, P. Sun, and Y. Cui. 2007. Predicting protein-protein interactions based on BP neural network, pp. 3–7. *In: Proceedings of IEEE International Conference on Bioinformatics and Biomedicine Workshops, 2007*. IEEE Computer Society Washington, DC, USA.
15. McMillan, L. E. and A. C. Martin. 2008. Automatically extracting functionally equivalent proteins from SwissProt. *BMC Bioinformatics* **9:** 418.
16. Michalopoulos, D. and C.-K. Hu. 2002. An error backpropagation artificial neural networks application in automatic car license plate recognition, pp. 1–8. *In: Lecture Notes in Computer Science*. Vol. 2358. Springer Berlin/Heidelberg.
17. Moreno-Hagelsieb, G. and K. Latimer. 2008. Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics* **24:** 319–324.
18. Naik, A. D. and S. S. Bhagwat. 2005. Optimization of an artificial neural network for modeling protein solubility. *J. Chem. Eng. Data* **50:** 460–467.
19. Nair, T. M., S. S. Tambe, and B. D. Kulkarni. 1994. Application of artificial neural networks for prokaryotic transcription terminator prediction. *FEBS Lett.* **346:** 273–277.
20. Needleman, S. B. and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48:** 443–453.
21. Oh, S.-H. 2011. Error back-propagation algorithm for classification of imbalanced data. *Neurocomputing* **74:** 1058–1061.
22. Ponting, C. P. 2001. Issues in predicting protein function from sequence. *Briefings Bioinformatics* **2:** 19–29.
23. Smith, T. F. and M. S. Waterman. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* **147:** 195–197.
24. Watson, J. D., R. A. Laskowski, and J. M. Thornton. 2005. Predicting protein function from sequence and structural data. *Curr. Opin. Struct. Biol.* **15:** 275–284.
25. Wilamowski, B. M. 2009. Neural network architectures and learning algorithms. *Ind. Electron. Mag. IEEE* **3:** 56–63.
26. Zhang, W., J. Chen, Y. Yang, Y. Tang, J. Shang, and B. Shen. 2011. A practical comparison of *de novo* genome assembly software tools for next-generation sequencing technologies. *PLoS One* **6:** e17915.