Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 647296, 7 pages
http://dx.doi.org/10.1155/2015/647296

Research Article

An Optimized Prediction Model Based on Feature Probability for Functional Identification of Large-Scale Ubiquitous Data

Gangman Yi

Department of Computer Science & Engineering, Gangneung-Wonju National University, Gangwon-do 220-711, Republic of Korea

Correspondence should be addressed to Gangman Yi; gangman@cs.gwnu.ac.kr

Received 18 August 2014; Accepted 9 September 2014

Academic Editor: Jong-Hyuk Park

Recently, there is a growing interest in the sequence analysis. In particular, the next generation sequencing (NGS) technique fragments the base sequence and analyzes the functions thereof. Its essential role is to arrange pieces of the base sequence together based on sequencing and to define the functions. The organization of unarranged piece of sequence is one of the active research areas; moreover, definition of gene function automatically is a popular research topic. The previous studies about the automatic gene function have mainly utilized the method that automatically defines protein functions by using the similarities of base sequence or the disclosed database and the protein interaction or context free method. This study aims to predict the category of protein whose function was not defined after learning automatically with GO by extracting the characteristics of protein inside the cluster. This study conducts clustering by using the protein interaction that is generated by the similarities of base sequence under the assumption that the proteins inside the cluster have similar function. The proposed method is to show an optimized result in accordance with the option after finding the option value that can give the outperformed prediction of GO, which classifies the functions based on the IPR and keywords inside the same cluster as the unique features.

1. Introduction

There is a growing demand to automatically predict the protein functions since there is a growing interest in the DNA analysis due to the development of technical equipment. Sequencing the sequence, which is composed of pieces, in the order is the basic technology of NGS. However, the studies to seek a method to predict automatically the protein functions or the so-called connected contig are being actively undertaken. The conventional method to predict the protein functions is to make groups as relevent functions mainly based on experiments. However, this method reached the limit in terms of time and efforts due to an increase in the quantity of data. In recent years, there have been a large amount of studies on the method to predict the protein functions in the most effective way by finding the relationship with the classifiers automatically through finding the features that can identify the functions.

The most basic protein classifier method is the method of defining the structure or functions through the protein family DB or finding the homology after determining the similarities between the sequences by using the sequence similarities program such as BLAST. One of the simple methods to define the protein functions automatically [1] is the method of conducting one-to-one mapping on the features and functions of protein. The most prominent example is InterPro2GO [2]. InterPro is protein domain database and GO is controlled vocabulary for gene annotation. GO is broadly subdivided into the three categories since it is the controlled vocabulary to annotate the functions of gene of various organisms and also it consisted of the hierarchical structure. InterPro2GO is the method that consisted of a simple manual mapping between InterPro term and GO term; this mapping table is made by checking the conserved common annotation and finding the specific level of GO term from the relevant family. This method is still possible to obtain InterPro by using InterProScan [3] even when one is not being able to know InterPro term just like a new sequence whose function is not defined. In addition, the study on a simple mapping work as to the GO annotation has been conducted in GOA [4, 5] projects.

Moreover, there have been some studies in which the accuracy as to the prediction was enhanced as compared with the conventional simple mapping method by defining the categories of features and protein functions more specifically with the machine learning method after extracting the DB based on the sequence similarities and the features based on the similarities [6–10]. The method for the protein prediction models using protein-interaction is being also utilized as a protein function classifier method [11, 12] in addition to the studies that defined the features and functional categories through the mathematical models or patterns. Enright et al. suggest the probability for the protein of each interaction using Bayesian [12]; thus, it suggested the probability of having functions unlike those conventional methods that would predict based on whether to "have relevant functions" or "have no relevant functions" [13, 14]. MCL [12] is the method of clustering by giving Option "*I*" in accordance with the mutual correlation between proteins by Markov model using the BLAST outcomes that is mainly used to measure the sequence similarity. Option "*I*" is the value to conduct the inflation and expansion in the Markov model; thus, it is possible to see the changes of elements inside the clustering depending on the value of *I*. Another function prediction method is the context-free method; it utilizes the context of several DBs of Uniprot or PubMed [15].

The research about sequence clustering algorithm is also one of the methods to attempt to group the relevant biological sequences. MCL [12] is a graph based unsupervised clustering by employing the Markov. BLASTClust [16] is an application of clustering the sequence by the single-linkage clustering. CD-HIT [17] is a fast method for clustering protein using the greedy algorithm. If a sequence is sufficiently similar to some clusters, that sequence is assigned to it; otherwise, make a new cluster. UBLAST [18] makes a cluster using USEARCH algorithm which searches high-scoring global alignment. Among these sequence algorithms, MCL provides unsupervised clustering option by the *I* value, which is able to look for the optimized performance by the sequence similarity; thus MCL is used as a clustering method at this suggested model.

The proposed model is to predict the classifier of that function by training the relationship with the features [19, 20] after clustering through protein-interaction. Protein-interaction makes a relational graph by using MCL tool based on the sequence similarities of protein, resulting in making a clustering for the protein. Protein inside the same cluster can be assumed to have similar functions. Yi et al. and Nhat et al. clustered by using the GO of gene and compared the results hereof with the other clustering algorithm [21, 22]. This study shows that there is a constant correlation between GO and clustering. In other words, protein inside the clustering can be explained to have the same GO since it performs a similar function. Based on this idea, that proteins in the same cluster can have similar GO, the suggested method seeks the related features. The features used in the proposed model are InterPro and keyword. They already proved the relationship between InterPro and GO in [23–25]. In conclusion, the purpose of this research is that the investigation of the relationship between selected feature such like IPR, keyword, and GO term within the cluster. The clusters make groups by the sequence similarity.

Table 1: Total data set.

	Protein	IPR	GO	Keyword
Number	5285	4862	4673	461

This study is managed as follows. It described the relationship diagram as to the model and test method that would describe and learn the properties of data used in Section 2 and also mainly stated the detailed description as to the learning model and test method. In Section 3, the most optimized option value of the proposed model is to be presented by comparing and analyzing the result values based on the model explained in Section 2. In Section 4, the conclusion and future direction are to be presented.

2. Method

2.1. Data Set. As for data, *Saccharomyces cerevisiae's* data was utilized at Uniprot. The features to be used are GO (gene ontology) and IPR in DR (database cross-reference) and keyword in the KW (keyword) line from FLAT format. The gene ontology (GO) is a controlled vocabulary of terms to describe protein functions. It consists of the three large categories that include "biological process," "molecular process," and "cellular component" and it is uniquely composed of the hierarchal structure. InterPro terms [26] are defined in the InterPro database, which is a curated protein domain database that acts as a central reference for several protein family and functional domain databases, including Prosite, Prints, Pfam, Prodom, SMART, TIGRFams, and PIR SuperFamily. We have previously shown that InterPro was an important source of features for identifying GO terms for proteins [25, 27]. Keyword performs the important role of reference for sequence as a predefined entry based on the functions, structure, or other categories. Among the extracted *Saccharomyces cerevisiae* data, those proteins not having GO were excluded and IEA (Inferred Electronic Annotation) out of the evidence codes of GO was not utilized. IEA is the property that was automatically extracted from the database, and usually it is not the feature that was revealed experimentally by manual. Thus, it is inappropriate for creating a model based on the calculating technique. To summarize, those proteins having at least one IPR or keyword while having more than one GO among the data of *Saccharomyces cerevisiae* were utilized as the data. As shown in Table 1, a total number of proteins used were 5,285. A total number of GOs owned by these proteins were 4,673, whereas a total number of IPRs and keywords owned by these proteins were 4862 and 461, respectively.

To verify the validity of the proposed model and find the optimized option by using the above data, the 10-fold cross validation technique was utilized. The 10-fold cross validation utilizes a certain part of data as the learning model and another part as the test. In other words, it is mainly utilized when testing the validity whether or not, the proposed method is correctly performed, or finding

the optimal parameter values of the proposed model. On average, one-fold consisted of approximately 520 proteins by dividing the proteins into 10 subsets randomly and only one subset out of those subsets is to be used as the testing data and the remaining 9 folds are to be used as the training data. All the protein data can obtain test results by the data learned by the data that excludes oneself one time through conducting the aforementioned task for each fold.

2.2. Training Method. Firstly, the training model executes BLAST in order to conduct the learning for the folds that consisted of the 9 subsets. BLAST is the program to determine the similarity between sequences; thus, it is the precedent phase process to execute Markov Cluster Algorithm (MCL), which is the program for creating a cluster. MCL algorithm is an unsupervised algorithm that can vary the cluster size promptly and variably based on graph. It is possible to create a matrix that can determine the similarity of sequence for each protein by using BLAST. The data to be expressed in the matrix at this point can become a real number or binary number such as *e*-value. The matrix becomes a node for each protein and can draw a graph by creating a weighted edge by the similarity of connection. Also, it can cluster with a certain threshold value in this graph. In the used training data, *e*-value option value was set at 0.01 when executing BLAST. *E*-value tends to have more similarity when being closer to 0. It represents all the cases that are shown when there is no option as to cut off; thus, the threshold value was set at 0.01 in order to reduce the weight as to those proteins whose similarity was small in terms of graph configuration by setting *e*-value.

Second, I value, which was the option value of MCL, was set at various values. I is the value to execute the inflation and expansion in the Markov model; thus, it is possible to obtain the optimized MCL results with the modified I value. The purpose is to find which parameter represents the optimal result condition by setting I with the four methods of 1.4, 2, 4, and 6. Figure 1 represents the number of proteins owned by each cluster ID when the default value of I is 2.0. As shown in the figure, there is a large quantity of proteins inside one cluster as for those clusters whose cluster ID number is small. However, those IDs with a higher number have a small quantity of proteins inside the cluster. In addition, it was found that it would be changed to a graph of long tail in accordance with I value.

As shown in Figure 1, the number of proteins in each cluster may be 2 or less. In other words, only the internal proteins belong to the corresponding cluster and at least one is to be used as the learning data and another one is to be used as the test data. Thus, those cases in which the number of proteins inside the cluster is less than one are to be excluded. This option is stipulated by Cutoff_2. As a result, 627 clusters are used as data. In addition, it is set in the cases of more than 5 (Cutoff_5) and 10 (Cutoff_10) to confirm the test results.

To express the common IPR, GO and keyword for each feature in each cluster in a formula, it can be derived from

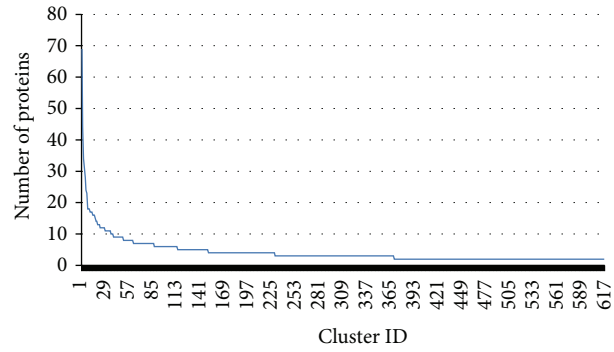

Figure 1: Number of proteins for each cluster at $I = 2.0$ (mcl).

the following formula. P_IPR_l means IPR owned by Protein P. The following can be derived by this formula,

$$\bigcap_{i=1}^{n} P_i\_\mathrm{IPR}_l. \tag{1}$$

Protein P_i contained in the cluster has n units and IPR contained in each protein has the quantity of l. l at this point may vary for each P_i in the cluster. GO and keyword can be expressed as shown in the following formula. m and n mean GO and keyword owned by each protein in the cluster and the quantity may vary depending on Protein P_i.

$$\bigcap_{i=1}^{n} P_i\_\mathrm{GO}_m, \qquad \bigcap_{i=1}^{n} P_i\_\mathrm{KW}_n. \tag{2}$$

The formula derived by the above two formulas is as follows. This formula stands for the common IPR, GO and keyword in each cluster, which is used as a training information.

$$C_{j=1}^{k}\left(\bigcap_{i=1}^{n} C_{jP_i\_\mathrm{IPR}_l}, \bigcap_{i=1}^{n} C_{jP_i\_\mathrm{GO}_m}, \bigcap_{i=1}^{n} C_{jP_i\_\mathrm{KW}_n}\right). \tag{3}$$

$C_{j=1}^{k}$ means a total number of Cluster ID and there is a total of k cluster IDs and the proteins in each cluster ID are to be expressed by C_{jP_i}. The common IPR, GO and keyword owned by the proteins of each cluster can be expressed by $C_{jP_i\_\mathrm{IPR}_l}$, $C_{jP_i\_\mathrm{GO}_m}$ and $C_{jP_i\_\mathrm{KW}_n}$, respectively.

In short, the flow chart of the procedure as to the testing is as shown in (a) in Figure 2. It is required to first execute BLAST using the learning data and set the E-value cutoff value at 0.01 at this point. It is then required to obtain the result composed of the 4 clustering results for each training data through the 4 Option I by using the BLAST results. At this point, it is required to examine whether the common IPR, GO, and keyword owned by all the proteins inside each cluster is as many as the number of clusters. The protein in which the common IPR and keyword exist while there is at least more than one common GO in each cluster is defined as the learning data. When the number of proteins inside the cluster is 2, 5, or 10, each of these cases is defined as Cutoff_2, Cutoff_5, and Cutoff_10.

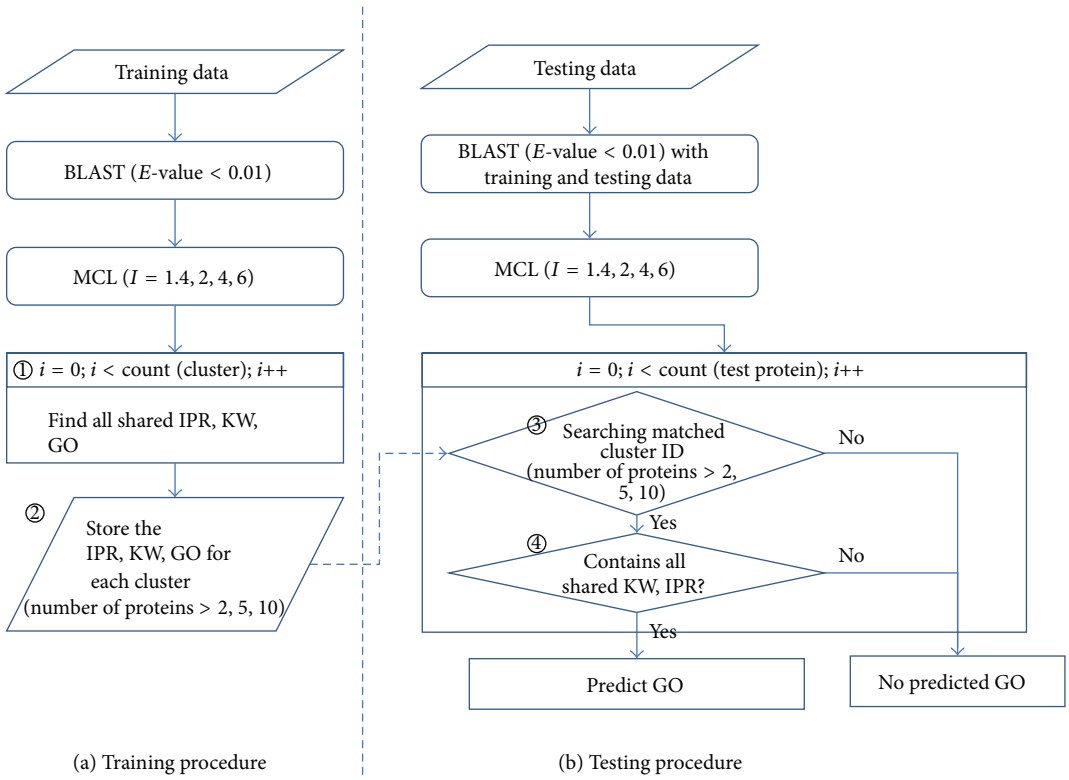

Figure 2: Flowchart for training and testing procedure.

Table 2: Percentage of number of proteins for each Option I at training result.

	$10 \leqq x$	$9 \leqq x < 10$	$8 \leqq x < 9$	$7 \leqq x < 8$	$6 \leqq x < 7$	$5 \leqq x < 6$	$4 \leqq x < 5$	$3 \leqq x < 4$	$2 \leqq x < 3$
I1.4	0.67	0.22	0.18	0.67	1.27	2.45	4.91	15.87	63.15
I2.0	0.47	0.25	0.29	0.90	1.35	2.58	5.36	18.36	69.43
I4.0	0.35	0.20	0.37	0.86	1.12	3.01	5.56	20.86	76.67
I6.0	0.27	0.16	0.49	0.76	1.06	2.99	6.03	21.49	77.75

As for Figure 3, the corresponding cluster has the three proteins (MCFS1_YEAST, MCFS2_YEAST, and YM60_YEAST) when I is 1.4 and the cluster ID is 292 and it has GO:00051792 jointly, which is expressed in blue. IPR and keyword also have "IPR000952," "IPR000073," "IPR012020," "hydrolase," "reference proteome," and "complete proteome" in common, which are represented in green and yellow, respectively. As for cluster ID 292, the number of proteins inside the cluster is 3; thus, it becomes the trained result that belongs to Cutoff_2. Nonetheless, it would not belong to Cutoff_5 and Cutoff_10, since the number of proteins inside the cluster is less than 5 or 10. Table 2 represents the percentage as to the mean number of proteins of 10-fold in accordance with each Option I when it is Cutoff_2 in the training result. For instance, the first-fold among the 10 folds when I is 1.4 has a total of 444 clusters and there are only 2 whose number of proteins inside the cluster are more than 10 while having at least more than one GO inside the cluster in common and having IPR and keyword as well. When converting it into percentage, it becomes $(2/444) * 100 = 0.4$ and 0.67 is its mean value as shown in Table 2 when obtaining the mean value of other 2~9-fold. In other words, it can be regarded as the mean probability distribution as to the number of proteins in accordance with Option I as the learning data of 10-fold. When I is 1.4, 2, 4, and 6, I has 63.15, 69.43, 76.67, and 77.75, respectively, when the number of proteins inside the cluster is 2. Mostly, the number of proteins inside the cluster tends to be small.

2.3. Testing Method. The flow chart of the procedure as to the testing is as shown in (b) in Figure 2. As a result of training model, each cluster ID has the common IPR, GO, and keyword (Figure 2-②). The test data first executes BLAST to test the sequence similarity just like the training method as shown in shown in Figure 2-(b). At this point, DB and query that executes BLAST utilize all the proteins used in the training as well as the test proteins. MCL conducts clustering based on the BLAST result; thus, the cluster ID in which the test data belongs can be obtained only when including the training data containing the test. It is required to find the cluster ID obtained from MCL for each tested protein

MCFS1_YEAST
GO:0034321 GO:0034338 GO:0051792 GO:0051793
IPR012020 IPR000073 IPR000952
Acyltransferase Complete proteome Hydrolase Reference proteome Serine esterase Transferase.

MCFS2_YEAST
GO:0005811 GO:0034319 GO:0017171 GO:0034338 GO:0044255 GO:0051792 GO:0051793
IPR012020 IPR000073 IPR000952
Acyltransferase Complete proteome Hydrolase Reference proteome Serine esterase Transferase.

YM60_YEAST
GO:0016746 GO:0051792
IPR012020 IPR000073 IPR000952
Complete proteome Hydrolase Reference Serine esterase.

Figure 3: Example of cluster ID 292 at $I = 1.4$.

FKBP_CANAL

GO:0008144 GO:0003755 GO:0035690 GO:0000413

IPR023566 IPR001179

Complete proteome Cytoplasm Isomerase Reference proteome Rotamase.

CLUSTERID:85

GO:0003755

IPR001179 IPR023566

Reference proteome Complete proteome Isomerase

Figure 4: Example of tested protein.

(Figure 2-③) and confirm whether the corresponding cluster ID has the common IPR, GO, and keyword in the results obtained by the training model (Figure 2-④). At this point, the matching probability of IPR and keyword was set at more than 0.5. For instance, the test protein FKBP_CANAL has the 4 GOs, 2 IPRs, and 4 keywords (Figure 4). This protein belongs to cluster ID 85 and the proteins belonging to cluster ID 85 are shown to have the 5 common features such as "IPR001179," "IPR023566," "reference proteome," "complete proteome," and "isomerase" by referring to the data generated by the learning (Figure 2-②). The test protein FKBP_CANAL shows that the 5 features out of the 7 features of IPR and keyword are matched. The matching probability is more than 0.5; thus, GO:0003755 is to be set as the prediction GO of FKBP_CANAL. As for this protein, the accuracy is 0.25 since only one GO (GO:0003755) is predicted out of a total of 4 GOs (GO:0008144, GO:0003755, GO:0035690, and GO:0000413).

The formula thereof is as follows. If the protein to be test is Test_p, it will be possible to obtain a cluster ID to be obtained by the result of BLAST and MCL (j). If the ratio of number of common IPR ($C_{jP_i\_\text{IPR}_l} \cap \text{Test}_{p_\text{IPR}}$), KW ($C_{jP_{i\text{KW}_n}} \cap \text{Test}_{p\_\text{KW}}$) that is owned by the corresponding j cluster among IPR and KW owned by the proteins to be tested as compared with the number of lists of IPR ($\text{Test}_{p\_\text{IPR}}$) and KW ($\text{Test}_{p\_\text{KW}}$) of the proteins to be tested is more than 0.5, then the prediction will be conducted by GO ($C_{jP_{i\text{GO}_m}}$) of the proteins to test the common GO owned inside the cluster

$$\text{Predict}\left(j, \text{Test}_p\right) = C_{jP_{i\text{GO}_m}},$$
$$\text{if } \frac{\left(\left(C_{jP_{i\text{IPR}_l}} \cap \text{Test}_{p\_\text{IPR}}\right) \cup \left(C_{jP_{i\text{KW}_n}} \cap \mathit{Test}_{p\_\text{KW}}\right)\right)}{\left(\text{Test}_{p\_\text{IPR}} \cup \text{Test}_{p\_\text{KW}}\right)} > 0.5. \tag{4}$$

3. Performance Evaluations

Figure 5 represents the number of proteins that had at least more than one GO by the proposed method in accordance with the Option I and cutoff value of MCL. At this point, the three cutoff options mean that the minimum numbers of proteins inside each cluster from the data generated by the training model were more than n. For example, "Cutoff_10" represents that the numbers of proteins in each cluster are more than 10. Absolutely, cluster should have more protein if the cutoff option is small. The more data can be utilized

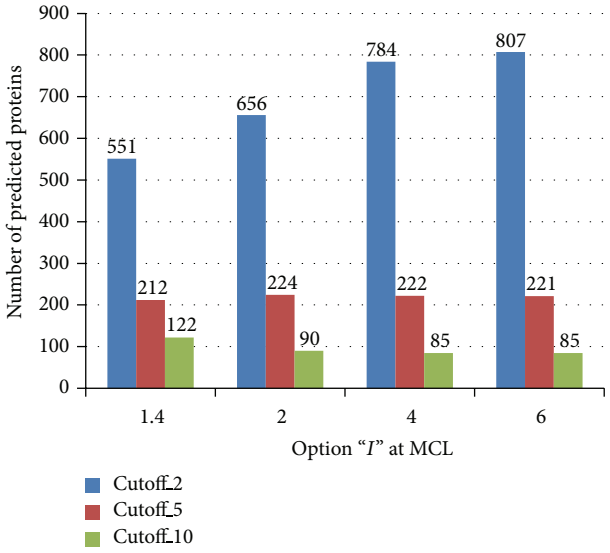

Figure 5: Number of proteins which has at least one GO by the test model.

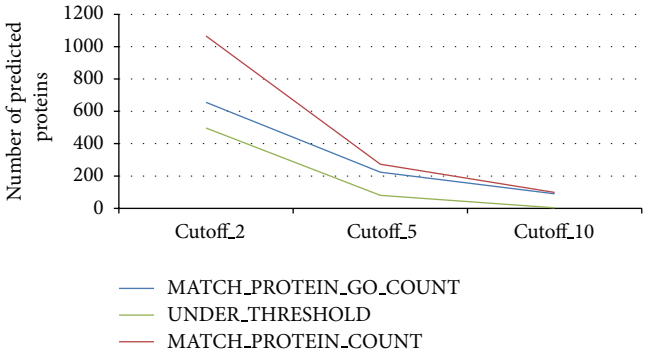

Figure 6: Tested number of proteins at $I = 2.0$.

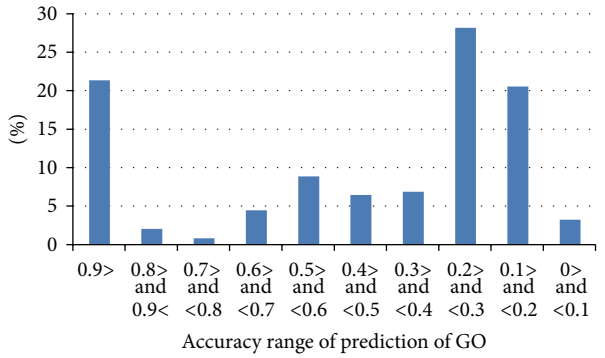

Figure 7: Accuracy of prediction of GO at $I = 2$ and Cutoff_5.

as a learning data, if the cutoff option is larger; thus, the number of predicted proteins would also be larger. However, the accuracy thereof would be lower with a smaller number of proteins learned inside the cluster. In the case of Cutoff_5, in other words, the date learned by at least more than 5 proteins, it will represent the largest number of predictions when I is 2. In addition, it will be shown that the number of predictions will slow down gradually from $I = 2$ option in the case of Cutoff_10.

MATCH_PROTEIN_COUNT, MATCH_PROTEIN_GO_COUNT, and UNDER_THRESHOLD when I was 2.0 based on the diagram of Figure 5 were investigated (Figure 6). MATCH_PROTEIN_COUNT is the protein having more than the predicted GOs as to the case where I is 2.0 by the entire learning data. Since all proteins are not predicted correctly, the numbers of proteins as to the case in which the accurately predicted GO were matched by comparing with the actual test data were defined as MATCH_PROTEIN_GO_COUNT. The gap of MATCH_PROTEIN_COUNT and MATCH_PROTEIN_GO_COUNT is incorrectly predicted GO. On the other hand, those proteins that could not have GO since IPR or keyword were matched with less than a certain threshold, which is defined as 0.5 in this experiment, were defined as UNDER_THRESHOLD. As for Cutoff_2, a large quantity of proteins had GO as expected when the number of proteins inside the learned cluster were at least more than 2. However, IPR and keyword were not matched since a majority was below the standard level; thus, there were many proteins that could not be predicted. As a result of Figures 5 and 6, the accuracy was enhanced with a higher cutoff value; however, the optimal condition was the case in which I was 2 and the cutoff was 5 since the number of predicted proteins was reduced. In other words, it was confirmed that they would be meaningful as the learning data only when the number of proteins used for learning inside the cluster was at least more than 5.

Figure 7 represents the prediction accuracy by diagram based on each protein when testing at each fold as to the case when the selected I is 2 and the cutoff is 5 as the optimal option. The number of GOs of the FKBP_CANA defined in Uniprot in the example of Figure 4 are GO:0008144, GO:0003755, GO:0035690, and GO:0000413. Of those, the accuracy was 0.25 by predicting GO:0003755. Using the number of GOs that was originally owned by UniProt and suggested method, it represents the prediction possibility of GO by the 10 fold cross validation and the horizontal axis means the percent within the scope of prediction accuracy. 0.9 represents the ratio between 0.9 and 1 and it is represented as a relatively high percentage since it is approximately 20 percent. A value of higher than 50 percent means the accuracy of 0.3 or higher. It is represented as the one to predict at least one GO out of the 3 GOs that have protein. It shows that the accuracy is high in the case where there is GO predicted through the suggested model.

4. Conclusions

This study conducted clustering under the assumption that the functional classifier inside the cluster had similar functions and utilized the features extracted inside the cluster as the learning data. When finding protein whose function is unknown, the model that predicts GO (or the controlled vocabulary) was defined through the learning and learned data documents of those proteins whose function was already

defined. This is the existing functional prediction, which is the method to harmonize appropriately those frequently used methods such as sequence similarity, protein-interaction, and context-free; thus, it could increase the prediction probability of GO.

However, in the case of used MCL cluster, the proteins in the cluster are often small in number with two or three higher serial number. When the number of proteins inside the cluster is small, they might not be able to perform the role of a cluster that would bind similar functions of protein since they are divided into pieces. The objective is to create a model to increase the GO predictability by using the features other than IPR and keywords as increasing the number of proteins inside the cluster by applying the other cluster methods in addition to MCL.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2063006).

References

[1] E. Elsayed, K. Eldahshan, and S. Tawfeek, "Automatic evaluation technique for certain types of open questions in semantic learning systems," *Human-Centric Computing and Information Sciences*, vol. 3, article 19, 2013.

[2] E. Camon, M. Magrane, D. Barrell et al., "The gene ontology annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro," *Genome Research*, vol. 13, no. 4, pp. 662–672, 2003.

[3] P. Jones, D. Binns, H.-Y. Chang et al., "InterProScan 5: genome-scale protein function classification," *Bioinformatics*, vol. 30, no. 9, pp. 1236–1240, 2014.

[4] D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan, and R. Apweiler, "The GOA database in 2009—an integrated Gene Ontology Annotation resource," *Nucleic Acids Research*, vol. 37, supplement 1, pp. D396–D403, 2009.

[5] E. Camon, M. Magrane, D. Barrell et al., "The Gene Ontology Annotation (GOA) database: sharing knowledge in uniprot with gene oncology," *Nucleic Acids Research*, vol. 32, pp. D262–D266, 2004.

[6] J. Jung, G. Yi, S. A. Sukno, and M. R. Thon, "PoGO: prediction of gene ontology terms for fungal proteins," *BMC Bioinformatics*, vol. 11, article 215, 2010.

[7] S. Khan, G. Situ, K. Decker, and C. J. Schmidt, "GoFigure: automated gene ontology annotation," *Bioinformatics*, vol. 19, no. 18, pp. 2484–2485, 2003.

[8] D. M. A. Martin, M. Berriman, and G. J. Barton, "GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes," *BMC Bioinformatics*, vol. 5, article 178, 2004.

[9] A. Vinayagam, C. del Val, F. Schubert et al., "GOPET: a tool for automated predictions of gene ontology terms," *BMC Bioinformatics*, vol. 7, article 161, 2006.

[10] A. Vinayagam, R. König, J. Moormann et al., "Applying Support Vector Machines for gene ontology based gene function prediction," *BMC Bioinformatics*, vol. 5, article 116, 2004.

[11] M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun, "Prediction of protein function using protein-protein interaction data," *Journal of Computational Biology*, vol. 10, no. 6, pp. 947–960, 2003.

[12] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Research*, vol. 30, no. 7, pp. 1575–1584, 2002.

[13] K. Salim, B. Hafida, and R. S. Ahmed, "Probabilistic models for local patterns analysis," *The Journal of Information Processing Systems*, vol. 10, no. 1, pp. 145–161, 2014.

[14] K. J. Nishanth and V. Ravi, "A computational intelligence based online data imputation method: an application for banking," *Journal of Information Processing Systems*, vol. 9, no. 4, pp. 633–650, 2013.

[15] N. Daraselia, A. Yuryev, S. Egorov, I. Mazo, and I. Ispolatov, "Automatic extraction of gene ontology annotation and its correlation with clusters in protein networks," *BMC Bioinformatics*, vol. 8, article 243, 2007.

[16] ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.html.

[17] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, 2012.

[18] R. C. Edgar, "Search and clustering orders of magnitude faster than BLAST," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[19] A. P. James, B. Mathews, S. Sugathan, and D. K. Raveendran, "Discriminative histogram taxonomy features for snake species identification," *Human-Centric Computing and Information Sciences*, vol. 4, article 3, 2014.

[20] R. Pan, G. Xu, and P. Dolog, "Improving recommendations by the clustering of tag neighbours," *Journal of Convergence*, vol. 3, no. 1, pp. 13–20, 2012.

[21] G. Yi, S.-H. Sze, and M. R. Thon, "Identifying clusters of functionally related genes in genomes," *Bioinformatics*, vol. 23, no. 9, pp. 1053–1060, 2007.

[22] V. V. M. Nhat and N. H. Quoc, "A model of adaptive grouping scheduling in obs core nodes," *Jorunal of Convergence*, vol. 5, no. 1, pp. 9–13, 2014.

[23] S. Letovsky and S. Kasif, "Predicting protein function from protein/protein interaction data: a probabilistic approach," *Bioinformatics*, vol. 19, no. 1, pp. i197–i204, 2003.

[24] M. Deng, Z. Tu, F. Sun, and T. Chen, "Mapping gene ontology to proteins based on protein-protein interaction data," *Bioinformatics*, vol. 20, no. 6, pp. 895–902, 2004.

[25] J. Jung and M. R. Thon, "Automatic annotation of protein functional class from sparse and imbalanced data sets," in *Proceedings of the 1st International Workshop on Data Mining and Bioinformatics (VDMB '06)*, pp. 65–77, September 2006.

[26] N. Mulder and R. Apweiler, "InterPro and InterProScan: tools for protein sequence classification and comparison," *Methods in Molecular Biology*, vol. 396, pp. 59–70, 2007.

[27] J. Jung, *Automatic assignment of protein function with supervised classifiers [Doctoral thesis]*, 2008.