Automated Methods of Predicting the Function of Biological Sequences Using GO and Rough Set

Xu-Ning Tang[1], Zhi-Chao Lian[2], Zhi-Li Pei[2,3], and Yan-Chun Liang[2,*]

[1] College of Software, Jilin University, Changchun 130012, China
[2] College of Computer Science and Technology, Jilin University, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China
[3] College of Mathematics and Computer Science, Inner Mongolia University for Nationalities, Tongliao 028043, China
ycliang@jlu.edu.cn

Abstract. With the extraordinarily increase in genomic sequence data, there is a need to develop an effective and accurate method to deduce the biological functions of novel sequences with high accuracy. As the use of experiments to validate the function of biological sequence is too expensive and hardly to be applied to large-scale data, the use of computer for prediction of gene function has become an economical and effective substitute. This paper proposes a new design of BLAST-based GO term annotator which incorporates data mining techniques and utilizes rough set theory. Moreover, this method is an evolution against the traditional methods which only base on BLAST or characters of GO Terms. Finally, experimental results prove the validity of the proposed rough set-based method.

Keywords: GO BLAST Rough Set Theory.

1 Introduction

Along with the development of modern sequencing technology, the number of gene sequence is increasing everyday. A report coming from GenBank, a major repository of genomic data, shows an exponential increase in sequence data, during the last decade. As a result, biologists have to waste amount of time in finding out some useful information within specific domain. Even worse, different biological database might use different nomenclatures, which like some dialects, making information search, especially for computer-based information search, unavailable. So, how to store and take advantage of the information has become many biologists' common concern.

1.1 Gene Ontology

The emergence of Gene Ontology (GO) project has been used to solve the nomenclature problem. Gene Ontology project provides a set of unified, standard and hierarchical terms to note the functional characters of gene products [1]. People can use

* Corresponding author.

J.C. Rajapakse, B. Schmidt, and G. Volkert (Eds.): PRIB 2007, LNBI 4774, pp. 1–10, 2007.

nomenclature provided by GO project to annotate the biological functions of biological sequences.

Each item in GO database is composed with three key parts: gene product ID, GO terms and evidence code. Among them, gene product ID uniquely identifies the sequence of a gene product. Moreover, as sequence data alone is of limited use to biologists, GO project annotates the functions of gene products from three points of view. They are biological process, cellular component and molecular function. At last, evidence code indicates how annotation to a particular term is supported.

Essentially, each of these three types of terms can be separated into more detailed sub-categories, so that those terms construct a DAG (directed acyclic hierarchical graph), shown in Figure 1. Generally speaking, GO is a unified biological tool which can annotate gene product's function with a set of dynamic controlled vocabulary and it can keep on upgrading with the development of biology.

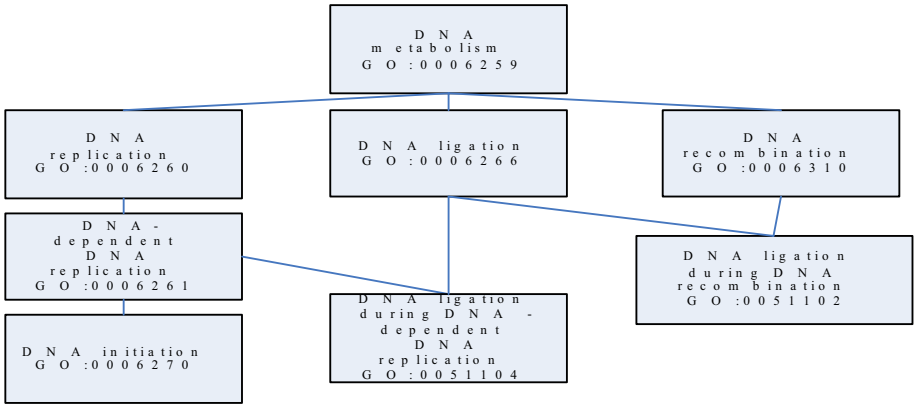

Fig. 1. Directed acyclic hierarchical graph of GO term

1.2 Basic Theory About Rough Set

Rough set has been introduced as a mathematical tool for dealing with fuzzy and uncertain knowledge in artificial intelligence application.

For convenience, we will introduce some basic concepts of rough set at first [2].

Definition 1. Given a knowledge system K= (U, R), for each subset $X \subseteq U$ and an equivalence relation $R \in ind(K)$, define two subsets:

Lower approximation: $\underline{R}X = \bigcup\{Y \in U / R \mid Y \subseteq X\}$

Upper approximation: $\overline{R}X = \bigcup\{Y \in U / R \mid Y \cap X \neq \varnothing\}$

Any subset defined by its lower and upper approximation is called a rough set.

Definition 2. Positive region: Let P and Q be equivalence relations within U, $pos_P(Q)$ is called the P-positive region of Q, such that $pos_p(Q) = \bigcup_{X \in U/Q} \underline{P}X$.

Definition 3. Let $DT = < U, C \cup D, V, f >$ be a Decision table, where C and D stand for conditional and decision attributes subsets, $C \cap D = \varnothing$, U is a non-empty, finite set called universe, V is called the value set, f stand for information function.

Definition 4. Let $\varnothing \subseteq X \subseteq C$, $\varnothing \subseteq Y \subseteq D$, $U/Y \neq \{U\}$, given $x \in X$, define significance of x with X (comparing with Y): $sig^Y_{X-\{x\}}(x) = (|S_X(Y)| - |S_{X-\{x\}}(Y)|)/|U|$.

2 Relative Work and Background

Although the emergence of GO project has been used to solve the problem of unification of nomenclature successfully, there is another remarkable problem about how to apply these nomenclatures on large-scale data effectively.

At present, a number of automated BLAST-based GO term prediction applications have been published. BLAST is the most widely used sequence alignment tool [3, 4]. It permits the user to find similar sequence according to high degrees of local similarity. Normally, it is very likely that similar sequences might be homological; therefore, the similar sequences may have the same or similar functions. For these reasons BLAST has been employed to assign GO terms to a novel sequence. Nowadays, there are several methods with the idea of predicting the function of gene product using BLAST and GO, such as TOP BLAST, GOtach, GOFigue, Goblet and some others [5-10]. These approaches can be roughly divided into several main kinds: graph-based, discriminant function-based and term distance concordance-based and so on. Among them the TOP BLAST is the most commonly used approach. However, TOP BLAST is not so accurate and convincing. As a result, this paper recommends a new design of BLAST-based GO term annotator which incorporates data mining techniques and utilizes rough set theory. Under the strict criterion, the new approach provides higher quality and more accurate functional prediction for a novel sequences than TOP BLAST can.

3 Rough Set-Based Method

3.1 Data Collection

The Gene Ontology data were downloaded and divided into three parts: training set, test set and BLAST-able database. This data consist of protein sequence data and their GO term associations. UniPort annotations, proteins and their GO term associations are

submitted by UniPort, is referred to as BLAST-able database. This data, consisting of 107,632 proteins, have high quality annotation. Non UniPort annotations, consisting of 3,537 proteins and their GO term associations are submitted by other sources, are referred to as training set and test set. In order to examine our method's validity, we employ cross-validation method. Each time we randomly select 1,200 proteins as test set and the other 2,337 proteins as training set.

Evidence code indicates how annotation to a particular term is supported. Some are supported by experiments, some are supported by literature and some are supported by computation method. According to different evidence codes, for training set and test set respectively we constructed 2 different experimental sets: one experimental set, called 7-evidence set, includes GO terms supporting by evidence codes such as: TAS, IDA, IC, IMP, IGI, IPI and IEP. Another experimental set, called NoIEA set, includes GO terms supporting by all evidence codes except IEA. For the reason that all GO terms within 7-evidence set are supported by evidence code which have high reliability, meanwhile the GO terms within NoIEA set just preclude those supported by evidence code of IEA, there is no doubt that GO terms in 7-evidenc are more reliable and accurate than those in NoIEA.

3.2 Accuracy Metrics

As we employ the strict evaluation method, precision and recall rate are defined as:

$$\text{Precision: } P = \frac{c}{p}$$

Where c is the number of correct predicted term assignments and p is the total number of predicted assignments.

$$\text{Recall rate: } R = \frac{c}{t}$$

Where c is the number of correct predicted term assignments and t is the total number of correct term.

$$\text{Harmonic Mean: } H = \frac{2}{1/P + 1/R}$$

Only if the predicted term is the right term which the source sequence indeed has, we count it as a correct prediction. Otherwise, prediction hit on either its parent term or its children term is considered as a false prediction.

3.3 Preparation

Before deducing rules from decision table, there are some preparation works to do.

3.3.1 Basic Concept

(1) Source sequence: we define those protein sequences which need prediction of function in training set as source sequence.

(2) Target sequence: we define those protein sequences returned by BLAST from BLAST-able database as target sequences which are similar to the source sequence.
(3) Unit: For each source sequence in training set, we returned 5 most similar sequences (in sort of ascending E-value) by BLAST from BLAST-able database. And these 5 most similar sequences construct a unit.
(4) Each GO term of those sequences belonging to the unit has 5 attributes described below: GO ID (which can uniquely identify the GO term), Rank (the ascending rank value of the highest matching result the term is found in), Times (the number of annotations using the term), E-value (a parameter returned by BLAST and stand for the similarity between source sequence and target sequence, the smaller the similar), and Score (another parameter returned by BLAST similar to E-value).

3.3.2 Calculate the Probability of Different Values of Each Attribute Within All Units

(1) For each source sequence in training set, we return a unit by BLAST and calculate those 5 attributes of the unit.
(2) For all units obtained, we calculate the probability of different values for each attribute in these units (P(Times=X) X=1,2,3,4,5; P(Rank=X) X=1,2,3,4,5; P(Score=X) X>0; P(E-Value=X) 0<X<1).

3.3.3 Calculate the Conditional Probability

When source sequence indeed has this GO term (K=1), calculate the conditional probability of 4 of these attributes: P(Times=X|K=1); P(Rank=X|K=1); P(Score=X|K=1); P(E-Value=X|K=1)).

For the reason that a particular GO term may occur in many different units, and those 4 attributes (excluding GO ID) of this GO term may have different values when they appear in different units, so it's very likely that we can judge whether the source sequence has this GO term by those 4 attributes' value. This preparation step will help us to make a discretization of rough set later.

3.4 Algorithm

Because of the difference among GO terms, it is very likely that we can predict whether the sequence has a particular GO term or not by checking those 4 attributes of the GO term within the unit. As a result, we treat all units as a whole set and generate a set of rules for each GO term in this set. With these rules, we can predict the terms of a sequence within the unit.

For each GO term, once it has occurred in at least one unit, we will deduce a set of rules about it by decision table. For example, GO: 000019 has emerged in 7 different units of training set, so we construct a raw decision table based on this GO term's situation, as shown in Table 1. After that, according to those conditional probabilities obtained from preparation step, we make discretization of the raw decision table and get the discrete decision table, as shown in Table 2.

Table 1. Raw decision table of GO: 000019, GO NO identify the GO term which needs deduction rules; Sequence stands for the unit which contains this GO term and is returned by BLAST according to a source sequence; K=1 means that the source sequence indeed contains this GO term

GO NO	Sequence	Rank	Times	E-Values	Score	K
GO:0000119	DDB|DDB019126	5	1	1.00E-82	305	0
GO:0000119	DDB|DDB021490	1	2	2.00E-11	72	0
GO:0000119	SGD|S000000397	5	1	4.00E-15	80	1
GO:0000119	SGD|S000001100	3	1	9.00E-22	102	1
GO:0000119	SGD|S000002382	5	3	8.00E-09	62	0
GO:0000119	SGD|S000002716	5	1	1.00E-10	64	1
GO:0000119	SGD|S000003095	1	1	2.00E-12	69	0

Table 2. Discrete Decision Table of GO: 000019, According to the result of preparation step, we divide: Times=4 or 5 as high times, Times=2 or 3 as mid times, Times=1 as low times; Rank=1 as high rank, Rank=2 or 3 as mid rank, Rank=4 or 5 as low rank; E-Value>1.00E-30 as high e-value, E-Value<1.E-100 as low e-value, others as mid e-value; Score<200 as low score, Score>800 as high score, others as mid score

GO NO	Sequence	Rank	Times	E-Value	Score	K
GO:0000119	DDB|DDB019126	low rank	low times	mid e-value	mid score	0
GO:0000119	DDB|DDB021490	high rank	mid times	high e-value	low score	0
GO:0000119	SGD|S000000397	low rank	low times	high e-value	low score	1
GO:0000119	SGD|S000001100	mid rank	low times	high e-value	low score	1
GO:0000119	SGD|S000002382	low rank	mid times	high e-value	low score	0
GO:0000119	SGD|S000002716	low rank	low times	high e-value	low score	1
GO:0000119	SGD|S000003095	high rank	low times	high e-value	low score	0

```
GO:000019
high rank >>>0
mid rank >>>1
mid times >>>0
mid e-value >>>0
mid score >>>0
low rank mid times >>>0
low rank mid e-value >>>0
low rank mid score >>>0
low rank mid times low score >>>0
low rank mid times high e-value >>>0
low rank low times high e-value >>>1
low rank low times mid e-value >>>0
low rank low times mid score >>>0
low rank low times low score >>>1
```

Fig. 2. Rules Deducing From Decision Table

At last, we run our program on the discrete decision table and obtain a set of rules, as shown in Figure 2. After that, we can understand which attribute or which combination of attributes can be used to decide whether the source sequence has the GO term or not.

Here is the specific algorithm of knowledge discovery based on the decision table: let decision table DT, where it contains n samples (a total of 11 rows), j conditional attributes (|C|=j), and one decision attribute (|D|=1). At first we calculate the significance for every single conditional attribute and describe rules of it. For each important conditional attribute, if there is decision attribute fully rely on this conditional attribute, the algorithm is over. Or else, we pick out the most important attribute (having the highest significance) among those important attributes. And then based on this most important attribute, we check each combination of two conditional attributes, with the purpose of finding out all important combinations of two conditional attributes and describing its rules. The step continues until all important knowledge is discovered.

Step1. //Algorithm first calculates significance of every single conditional attribute.

Let $C' = \varnothing$

For i=1 to j

$C' = C' \cup \{c_i\}$

Compute $pos_{C'}(D)$

IF($sig^{D}_{\varnothing}(c_i) > 0$)

Output rules

Let m=1 and FLAG=1

$C' = C' - \{c_i\}$

END FOR

IF($\exists pos_{C'}(D) = U$)

Algorithm Finish

ELSE

Find max($sig^{D}_{\varnothing}(c_i)$) and then let $J_1 = c_i$

$C' = C' \cup \{J_1\}$

Step2. //each time we add one conditional attribute to the combination and calculate its significance.

WHILE (FLAG)

m=m+1

For i=1 to j ($c_i \notin C^{'}$)

$C^{'} = C^{'} - \{c_i\}$

Compute $pos_{C^{'}}(D)$

IF($sig^{D}_{C^{'}-\{c_i\}}(c_i) > 0$)

Output rules

Let FLAG=1

$C^{'} = C^{'} - \{c_i\}$

END FOR

IF($\exists pos_{C^{'}}(D) = U$)

Algorithm Finish

ELSE

Find max($sig^{D}_{C^{'}-\{c_i\}}(c_i)$) and then let $J_m = c_i$

$C^{'} = C^{'} \cup \{J_m\}$

END WHILE

For each sequence in the test set, we also use BLAST to return 5 most similar sequences from BLAST-able database as a unit. And then we calculate the statistic result of those 4 attributes for each GO term contained in the unit. With those rules obtained from the training set, we can judge whether the source sequence contains the GO term.

4 Simulation Results and Analysis

The proposed method is examined by applying it on 7-evidence and NoIEA dataset. Moreover, it is compared with Top BLAST, as shown in Table 3:

Table 3. Comparison Between Top BLAST and Rough Set

Method	Precision		Recall		Harmonic Mean	
	7-evdience	NoIEA	7-evdience	NoIEA	7-evdience	NoIEA
Top BLAST	0.21	0.40	0.15	0.33	0.175	0.362
Rough Set	0.56	0.68	0.10	0.21	0.170	0.321

1. Either on 7-evidence dataset or on NoIEA dataset, the rough set-based method significantly improves the accuracy for the prediction of gene product. Especially, when it comes to 7-evidence dataset, the improvement could range from 21% to 56%, which is more obvious. It is almost 167% increase than before. Also, on NoIEA dataset our method improves the accuracy for prediction of gene product from 40% to 68%. It is nearly 70% increase than before. The main reason that the Rough Set-based method performs better on NoIEA dataset than on 7-evidence dataset is that the NoIEA dataset contains GO terms coming from both electronic annotation and curator-assigned. Those electronic annotations rely highly on sequences returned by BLAST, which is completely based on the similarity between sequences. As you known, our method also highly relies on those sequences returned by BLAST. As a result, precision will be greatly improved within NoIEA dataset. Similarly, within NoIEA dataset, GO terms which have evidence code such as ISS are also based on similarity. So that when there are evidence codes such as ISS, RCA in dataset, the result will be better. By looking through the results of TOP BLAST, we can also find the same situation.
2. On the contrary, Rough Set-based method is correspondingly lower than TOP BLAST in recall rate. It means that although most of its prediction is correct, Rough Set-based method covers only a small part of correct GO term.
3. At last, we know that Rough Set-based method and TOP BLAST have the similar performance by comparing the harmonic mean.

5 Conclusions

It is demonstrated that the rough set theory has great potential in bioinformatics, especially in the predicting functions of gene products. This paper proposes a data-mining-oriented method using rough set theory and applies it to prediction of gene function. Experimental results show that rough set-based method is able to provide high quality, conservative functional prediction for novel sequences. The proposed method can be used to improve the accuracy significantly by comparing with TOP BLAST. This method not only enables the electronic annotation to be more reliable but

also decreases the cost for functional prediction of novel sequences, which makes it an effective supplement of experimental method. However, we should not ignore the shortcoming of the rough set-based method, especially for the low recall rate. There are many reasons for the low recall rate: on one hand, test set contain plenty of situations never appear in the training set. On the other hand, our rules returned from the training set are too conservative to ensure the sensitive. In addition, according to the experiment we find that how to discretize the rough set is another key to improve the rough set-based method.

Acknowledgment

The authors are grateful to the support of the National Natural Science Foundation of China (60673023, 60433020), the support of the European Commission for the project of TH/Asia Link/010 (111084), and "985" project of Jilin University. The authors wish to thank Professor Jiwen Guan for guidance on rough set-based algorithm during this research project.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H.J., Cherry, M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.J., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene Ontology: tool for the unification of biology. Nature Genetics 25, 25–29 (2000)
2. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1992)
3. Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic Local Alignment Search Tool. Journal of Molecular Biology 215, 403–410 (1990)
4. Altschul, S.F., Madden, T.L., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 25, 3389–3402 (1997)
5. Hennig, S., Groth, D., Lehrach, H.: Automated Gene Ontology annotation for anonymous sequence data. Nucleic Acids Research 31, 3712–3715 (2003)
6. Groth, D., Lehrach, H., Hennig, S.: GOblet: a platform for Gene Ontology annotation of anonymous sequence data. Nucleic Acids Research 32, W313–W317 (2004)
7. Khan, S., Situ, G., Decker, K., Schmidt, C.J.: GoFigure: Automated Gene Ontology annotation. Bioinformatics 19, 2484–2485 (2003)
8. Martin, D.M.A., Berriman, M., Barton, G.J.: GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. BMC Bioinformatics 5, 178 (2004)
9. Joslyn, C., Mniszewski, S., Fulmer, A., Heaton, G.: The Gene Ontology Categorizer. Bioinformatics 20, i169–i177 (2004)
10. Verspoor, K., Cohn, J., Mniszewski, S., Joslyn, C.: A Categorization Approach to Automated Ontological Protein Function Annotation. Protein Science 15, 1544–1549 (2006)