SVM Approach to GeneRIF Annotation

Wen-Juan Hou, Chun-Yuan Teng,

Chih Lee and Hsin-Hsi Chen

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

E-mail: {wjhou, cydeng, clee}@nlg.csie.ntu.edu.tw;

hh_chen@csie.ntu.edu.tw

Abstract

In the biological domain, to extract the newly discovered functional features from massive literature is a major challenging issue. To automatically annotate GeneRIF in a new literature is the main goal in this paper. We try to find function words and introducers in the training corpus, and then apply such informative words to annotate the GeneRIF. The experiments showed that 48.15%, 49.78%, 32.31%, and 35.63% for the measure of Classic Dice, Modified unigram Dice, Modified bigram Dice, and Modified bigram Dice phrases. After applying SVM learning mechanism combing new weighting scheme and position information, we get much better performance.

1 Introduction

Information explosion in molecular biology and biomedicine is evolving rapidly, and becomes one of challenging problems in the new information era. How to obtain relevant information, for example, gene/protein functions, from a large amount of data collection is indispensable for bioinformatics researchers and experts. In the past, researchers in biomedicine have already constructed large scale of databases such as UMLS [1], Gene Ontology [2], SwissProt [3], GenBank [4], DIP [5], SNOMED [6], and LocusLink [7] *etc*., which are useful for researches to capture and organize information. However, creating and maintaining the knowledge bases requires enormous work. For example, if the paper includes a sentence like "*probably exist a binding between gene x and gene y*", we cannot assert that the paper is related to the molecular function. Thus, it needs careful judgment to add new information into a knowledge base. In other words, if we want to retrieve the relevant data from the massive literatures automatically, it needs a lot of efforts.

MEDLINE is a massive biomedical corpus for information extraction and knowledge discovery. Biomedical experts explore new development of some special topics by retrieving relevant documents from MEDLINE through search engines or information retrieval (IR) systems. These systems only return documents satisfying users' information needs instead of locating the relevant sentences denoting the specific functions. For example, during exploring molecular functions, users have to go through the whole documents to find the relevant information, and align it to a suitable database entry. To solve the above problem, some efforts have been made to extract functional relations [8, 9, 10, 11, 12]. Those only extract protein or gene interactions rather than the whole functions in the text.

This paper investigates how to extract molecular functions from the literatures. More precisely, the particular goal is to reproduce the GeneRIF annotation as stated in the secondary task of TREC 2003 Genome Track [13]. The Gene References into Function (GeneRIF) exists in LocusLink database [14] and it provides a simple mechanism to allow scientists to add functional annotation of loci. The rest of the paper is organized as follows. Section 2 presents the architecture of our extracting procedure. The basic idea and the experimental methods in this study are introduced in Section 3. Section 4 shows the results and makes some discussions. Finally, Section 5 concludes the remarks and lists some future works.

2 Architecture and the Extracting Method

2.1 Background

Generally, a gene name may have several aliases, and different functions may be discovered in different literatures. A complete annotation system consists of two major stages, including extraction of molecular function for a gene from a literature and alignment of this function with a GO identifier. Figure 1 shows an example. The left part is an MEDLINE abstract with the function description highlighted. The middle part is the corresponding GeneRIF, which is extracted from the last sentence of the abstract. The matching words are in bold, and the similar words are underlined. The right part is the GO annotation. This figure shows the feasibility of maintaining the knowledge bases and ontology using natural language processing technology. To complete this annotating procedure, we have to deal with the first stage automatically since the coverage of GeneRIF records in LocusLink depends on human experts and it cannot come up with the speedy growth of the literatures.

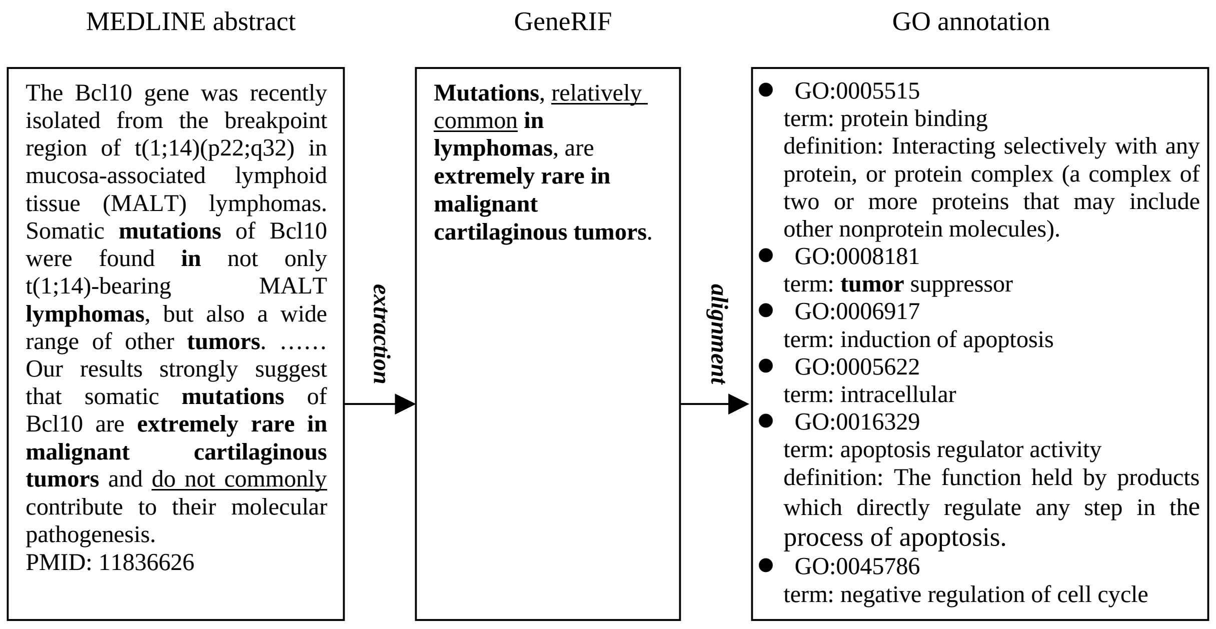
MEDLINE abstract
GeneRIF
GO annotation
The Bcl10 gene was recently isolated from the breakpoint region of t(1;14)(p22;q32) in mucosa-associated lymphoid tissue (MALT) lymphomas. Somatic **mutations** of Bcl10 were found **in** not only t(1;14)-bearing MALT **lymphomas**, but also a wide range of other **tumors**. …… Our results strongly suggest that somatic **mutations** of Bcl10 are **extremely rare in malignant cartilaginous tumors** and do not commonly contribute to their molecular pathogenesis.
PMID: 11836626
extraction
Mutations, relatively common **in lymphomas**, are **extremely rare in malignant cartilaginous tumors**.
alignment
● GO:0005515 term: protein binding definition: Interacting selectively with any protein, or protein complex (a complex of two or more proteins that may include other nonprotein molecules).
● GO:0008181 term: **tumor** suppressor
● GO:0006917 term: induction of apoptosis
● GO:0005622 term: intracellular
● GO:0016329 term: apoptosis regulator activity definition: The function held by products which directly regulate any step in the process of apoptosis.
● GO:0045786 term: negative regulation of cell cycle

Figure. 1. An Example of Complete Annotation from the Literature GO

A GeneRIF contains a few sentences that describe the function introduced in the scientific document identified by PMID. But how could we recognize the sentence exactly contains such information? We introduce two cues in this paper: function words and introducers. The details will be explained in Section 3.

2.2 Overall architecture

The overall architecture of the extraction from Medline to candidate GeneRIF is shown in Figure 2.

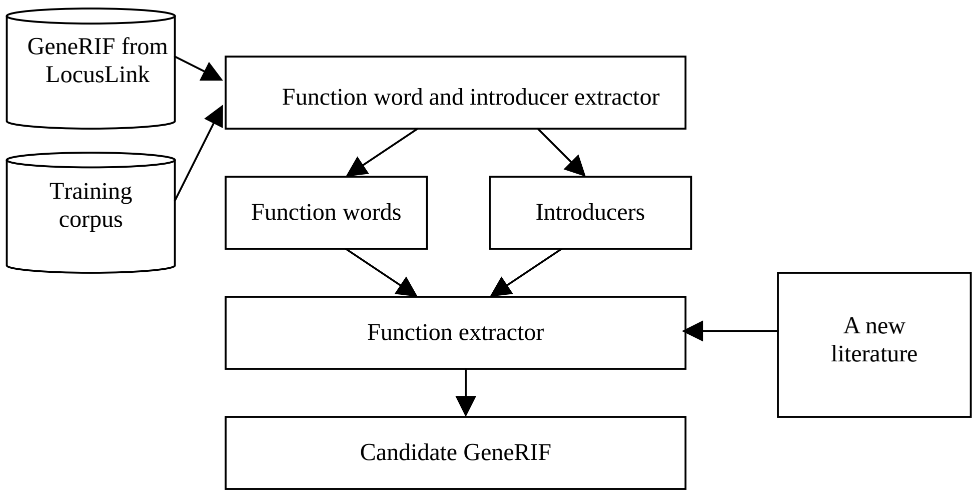

Figure 2. Architecture of Extracting Candidate GeneRIF

For getting the informative words, i.e., function words and introducers in this paper, from training data, we gather GeneRIF from LocusLink. Those are mutually exclusive with testing data in Genome Task and our testing data. And then, the system will compute the score that functions words and introducers contributed to. After applying the function extraction algorithm, the candidate GeneRIF is generated.

3 Function Extraction Approach

As described in Section 2, the score for each sentence depends on function words and introducers. The key issue is how to get function words and introducers and how to measure such scores. First, we prepare the training data and testing data, including those GeneRIFs existed in LocusLink and the corresponding Medline abstracts. We divided the corpus into three parts: training corpus, testing corpus, and testing topics for TREC 2003. The training corpus included 27,236 abstracts and the testing corpus including 9,005 abstracts. The details of our function extraction approach are illustrated as follows.

3.1 Training material preparation

Since GeneRIFs are written by human experts, some parts may include opinions of humans and/or some parts may be cited from papers. We focus on the latter parts. GeneRIF is not directly cut from papers but it has some relationship with paper content. Because the goal is to reproduce GeneRIF automatically, we find the most similar sentence with the corresponding GeneRIF in this paper. The measure is achieved by matching stemmed words between GeneRIF and each sentence. The sentence of matching the most number of words with GeneRIF is selected as the training data for the next stage. However, if more than one sentence matches the most number of words with GeneRIF, this abstract will be aborted because we cannot tell out which is correct. In this way, we get 27,236 sentences extracted from Medline abstracts.

3.2 Function words extraction

We call the matched words between GeneRIF and the selected sentence as *function words* in this paper. Function words form the favorite vocabularies that human experts used to describe the gene functions. Applying stopped word removal and stemming procedure, there are 22,275 function words extracted.

3.3 Introducers extraction

In the training data, there exists some important information except function words and we call it as the *introducer*. Function words are those words that human experts usually adopt to describe gene functions while introducers are the words that often co-occur with function words. In our approach, introducers are words appearing in the selected training sentence but not appearing in the other parts of the abstracts. Under such constraints, we get 621 introducers.

3.4 Compute the weight for each function words, weight(w_i)

Let $|w_{i:}|$ denote the frequency of the function word $w_{i:}$ in the training corpus. Then, weight(w_i)=$|w_i|/\sum_1^n |w_j|$, where n is the total number of function words. In this way, we can give the weight for each function words extracted in Section 3.2.

3.5 Compute the score for each sentence in the testing abstract

For the testing abstract, we compute two scores for each sentence using the weight defined in Section 3.4. The first score of sentence k, Score(S_k), is shown as follows.

Score(S_k)= $\sum \text{weight}(w_j)$, where w_j appear both in S_k and the set of function words.

To avoid the preference for the long sentence, we normalize score of sentence k, Score(Norm(S_k)), by the sentence length. The second score is defined as follows.

Score(Norm(S_k))= Score(S_k)/$|S_k|$, where $|S_k|$ is the total number of words where stop words have been removed from sentence k.

3.6 Function extraction algorithm

When a new literature comes in, we use the function extraction algorithm to annotate the candidate GeneRIF in the literature. This algorithm employs function words and introducers mentioned before. Besides, the statistics show that GeneRIF is often cited from the title or the first/the last sentence of the abstract. We adopt position information as the heuristic cues. The function extraction algorithm is illustrated as follows.

```
For each sentence k in test document d
          Compute Score($S_k$);
                    Sort Score($S_k$) in descending order;
          Since we cannot guarantee the sentence with the highest score is the candidate GeneRIF, we remain sentences
          with minor difference with the highest score where minor difference is gotten from the training data so that the
          reported set can cover the correct answer.
          If there is only one sentence remained
                    Produce this sentence as candidate GeneRIF
          Else
                    Count the number of matched words with introducers in the sentence;
                    If there is only one sentence with the highest matched numbers
                              Produce this sentence as candidate GeneRIF
                    Else
                              Produce the sentence with the following precedence
                                        1.   The title sentence.
                                        2.   The first sentence in the abstract.
                                        3.   The last sentence in the abstract.
                                        4.   Other position in the abstract.
```

The above algorithm compute the score with Score(S_k). If we compute the score with Score(Norm(S_k)), we get another set of candidate GeneRIF.

4 Results and Discussions

4.1 Results of official runs

We sent two runs to Genome Track on secondary task. The first run is called "we" and the score is computed with Score(S_k). The second run is called "nwe" and the score is computed with Score(Norm(S_k)). The evaluation result is shown in Table 1. The first column shows the measure criteria. "Classic Dice" is the classic Dice formula using a common stop word list and the Porter stemming algorithm. "Modified unigram Dice" gives added weight to terms that occur multiple times in both strings. "Modified bigram Dice" gives some addition weights to proper word order. Instead of measuring the Dice coefficient on single words it measures it on bigrams. For "Modified bigram Dice phrases", this measure only includes bigrams that have not had intervening stop words filtered. The second column "we" and the third column "nwe" denote the average performance for each measure and each run. The fourth to sixth columns represent the average score performed by 24 submissions from 14 groups attended in the secondary task.

Table 1. Experiments with "we" and "nwe"

Measure	we	nwe	best	median	worst
Classic Dice	48.15%	47.62%	57.83%	49.31%	9.42%
Modified unigram Dice	49.78%	49.37%	59.63%	51.30%	14.20%
Modified bigram Dice	32.31%	31.61%	46.75%	33.62%	0.15%
Modified bigram Dice phrases	35.63%	34.80%	49.11%	36.99%	0.17%

Compared with the other submissions, we summarize the number of topics performed as the best, between best and worst, and the worst. The result is shown in Table 2.

Table 2. Analysis with other submissions

Measure	we			nwe		
	best	between	worst	best	between	worst
Classic Dice	54	122	2	53	123	1
Modified unigram Dice	53	118	3	51	121	2
Modified bigram Dice	57	124	30	56	124	31
Modified bigram Dice phrases	61	122	35	59	123	37

From Tables 1 and 2, we find that although the performance is below the average median, we achieve the best results among 139 topics. This shows much effort should be made for further improvement.

4.2 Results with different weight schemes

To improve the result, we try different weight schemes used in Section 3.4 as follows.

- wga/nwga: Let $|w_{i,g:}|$ denote the frequency of the function word w_i in the GeneRIF and $|w_{i,a}|$ denote the frequency of the function word w_i in all articles. Then, weight(w_i)=$|w_{i,g}|/|w_{i,a}|$. We call this weight as wga. If normalization is applied, we call it as nwga.
- wgn/nwgn: Let $|w_{i,g:}|$ denote the frequency of the function word w_i in the GeneRIF and $|w_{i,ng}|$ denote the frequency of the function word w_i in all articles but not in the GeneRIF. Then, weight(w_i)=$|w_{i,g}|/|w_{i,ng}|$. We call this weight as wgn. If normalization is applied, we call it as nwgn.

Replacing the weight used in Section 3.4, new results are expressed in Table 3. Compared with Table 1, although "nwga" is improved, the others are reduced. It shows new weight schemes are not good enough.

Table 3. Experiments with "wga", "nwga", "wgn" and "nwgn"

Measure	wga	nwga	wgn	nwgn
Classic Dice	35.56%	50.18%	35.56%	48.53%
Modified unigram Dice	35.23%	46.71%	35.23%	50.38%
Modified bigram Dice	19.23%	33.47%	19.23%	36.72%
Modified bigram Dice phrases	21.76%	38.83%	21.76%	37.24%

4.3 Results with SVM method

Marcotte *et al.* [15] incorporated a weight-based method and a Bayesian approach in detecting abstracts discussing protein interactions. Several most-discriminating words are first identified by the *p*-score of each word assuming the number of occurrences of a word in an abstract conforms to a Poisson distribution under known dictionary frequency of this word. We therefore investigated the performance of this weighting scheme on GeneRIF sentences and Non-GeneRIF sentences. The weight for each word is calculated by taking negative log of the following probability density function.

$p(n \mid N, f) = e^{-Nf}\,\frac{(Nf)^n}{n!}$, where n is the number of occurrence of a given word in an abstract of N words, and f is the dictionary frequency of this word.

According to some preliminary study of the secondary task, it was observed that the position of a sentence in the abstract is an important clue to determine where the answer sentence is. Inspired by the work by the highest-scored team [16] in TREC 2003 Genome secondary task, we also combined sentence positions in our weighting scheme. With our weighting scheme of –log p, given an abstract, we first compute the scores of the title, the first three and the last five sentences, and then this feature vector is fed to a support vector machine (SVM) [17] to make the final decision.

Further, we'd like to know how SVM performs on the features used by the highest-scored team, which we called it sentence-wise bag-of-word model. In this case, 10,506 words were used, and therefore, the feature vector is 94,554 in length. For comparison, we design experiments called "wlog" and "nwlog" which did not contain SVM model, i.e., pure weight scheme used in Section 3.2. As usual, "nwlog" is the normalization version of "wlog". The results are shown in Table 4. It shows the SVM method really works well.

Table 4. Experiments with "wlog", "nwlog", "- log p" and "sentence-wise bag-of-word model"

Measure	wlog	nwlog	- log p	sentence-wise bag-of-word model
Classic Dice	31.55%	48.23%	56.86%	58.92%
Modified unigram Dice	30.14%	50.38%	58.81%	61.46%
Modified bigram Dice	16.11%	32.52%	45.08%	47.86%
Modified bigram Dice phrases	19.17%	36.03%	48.10%	50.84%

5 Concluding Remarks

This paper proposed automatic approaches to extract gene function in the literature. It is helpful to the work of conducting the GeneRIF in LocusLink database. The result shows that 48.15%, 49.78%, 32.31%, and 35.63% for the measure of Classic Dice, Modified unigram Dice, Modified bigram Dice, and Modified bigram Dice phrases in one of our official runs.

Combining the sentence position information, new weighting scheme, and SVM learning mechanism, the performance is improved significantly, i.e., 56.86%, 58.81%, 45.08%, and 48.10% for the measure of Classic Dice, Modified unigram Dice, Modified bigram Dice, and Modified bigram Dice phrases in "-log p" weighting scheme. It directs us to consider another training method for the next stage.

References

[1] Humphreys, B.L., Lindberg, D.A., Schoolman H.M. and Barnett G.O. (1998) "The Unified Medical Language System: an Informatics Research Collaboration," J*ournal of American Medical Information Association*, **5**, pp.1-11, 1998.

[2] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M. *et al*. (2000) "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics*, **25**, pp. 25-29, 2000.

[3] Bairoch, A. and Apweiler, R. (2000) "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000," *Nucleic Acids Research*, **28**, pp. 45-48, 2000.

[4] Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A. and Wheeler, D.L. (2000) "GenBank," *Nucleic Acids Research*, **28**, pp. 15-18, 2000.

[5] Xenarios, I., Fernandez, E., Salwinski, L., Duan, X.J., Thompson, M., J., Marcotte, E.M. and Eisenberg, D. (2001) "DIP: the Database of Interacting Protins: 2001 update," *Nucleic Acids Research*, **29**, pp. 239-241, 2001.

[6] SNOMED, http://www.snomed.org.

[7] Pruitt, K.D., Katz, K.S., Sicotte, H. and Maglott, D.R. (2000) "Introducing RefSeq and LocusLink: Curated Human Genome Resources at the NCBI, " *Trends Genet*, **16**(1): pp. 44-47, 2000.

[8] Blaschke, C., Andrade, M.A., Ouzounis, C. and Valencia, A. (1999) "Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions," *Proceedings of 7th International Conference on Intelligent Systems for Molecular Biology*, pp. 60-67.

[9] Park, J.C., Kim, H.S., and Kim, J.J. (2001) "Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorial Grammar," *Proceedings of Pacific Symposium on Biocomputing*, **6**, pp. 396-407, 2001.

[10] Rindflesch, T.C., Hunter, L. and Aronson A.R. (1999) "Mining Molecular Binding Terminology from Biomedical Text," *Proceedings of AMIA Symposium*, pp. 127-131, 1999.

[11] Rindflesch, T.C., Tanabe, L., Weinstein, J.N. and Hunter, L. (2000) "EDGAR: Extraction of Drugs, Genes, and Relations from Biomedical Literature," *Proceedings of Pacific Symposium on Biocomputing*, **5**, pp. 517-528, 2000.

[12] Sekimizu, T., Park, H.S. and Tsujji, J. (1998) "Identifying the Interaction Between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts," *Genome Information*, **9**, pp. 62-71, 1998.

[13] TREC 2003 Genome TRACK, http://medir.ohsu.edu/~genomics/.

[14] LocusLink database, http://www.ncbi.nlm.nih.gov/LocusLink/.

[15] Marotte, E.M., Xenarios, I., and Eisenberg, D. (2001) "Mining Literature for Protein-protein Interactions," *Bioinformatics*, **17**(4), pp. 359-363, 2001.

[16] Jelier, R., Schuemie, M., Eijk, C.V.E., Weeber, M., Mulligen, E.V., Schijvenaars, B., Mons, B., and Kors, J. (2003) "Searching for geneRIFs: concept-based query expansion and Bayes classification," *TREC 2003 work notes*, 2003.

[17] Hsu, C.W., Chang, C.C., and Lin, C.J. "A Practical Guide to Support Vector Classification." Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.