Protein function assignment using term-based support vector machines — BioCreAtIvE Task Two 2003

Simon B. Rice[§] Goran Nenadić[§*] Benjamin J. Stapley[§]

UMIST
PO Box 8
Manchester M60 1QD
United Kingdom

Abstract

In this paper we describe a machine-learning approach to BioCreAtIvE Task Two. The method uses support vector machines to assign GO codes to proteins, and is based on co-occurring terms that are automatically extracted from documents. For each GO code we learn a classifier from the training data. Target proteins (represented by term features from documents in which they appear) are tested against each of the classifiers, and the highest scoring GO codes are assigned. The same approach is employed for passage extraction: a paragraph (pertaining to a given protein) that is the highest scoring for a given GO code is selected. We describe different submissions and discuss the results of each subtask. The most promising results are for combined document retrieval and GO term assignment (Task 2.3) – 50% precision. The method appears to work best when a substantial set of relevant documents can be obtained for the protein and there is sufficient data to train accurate classifiers.

Introduction

Automatic extraction of information on protein function from free text is undoubtebly task of great relevance and utility to molecular biologists. We have previously developed methods for combined retrieval and classification of a protein's subcellular location using support vector machines (SVM) and a bag of words approach [4]. The method we employed in BioCreAtIvE Task Two were largly derived from this work. SVMs have been demonstrated to perform very well at document classification tasks and we construe the protein classification task as a modified form of this problem.

To our mind, Task 2.3 represents the most realistic and challenging problem amongst the three sub-tasks. Assignment of a passage to a protein-GO term pair (Task 2.1) is generally a problem that would only arise if the protein-GO term assignment was derived from non-text methods (e.g. homology searching), otherwise the assignment must have been made by manual analysis of relevant text previously. Task 2.2 is closer to a real-world problem (i.e. assignment of a GO term to a protein given a relevant document), but it is rare that a document can be gaurenteed to be relevant to both the protein of interest and the GO term to be assigned; this could not be known *a priori* without an expert annotator having read the document. Our reasoning was that if one could solve Task 2.3, tasks 2.2 and 2.1 could be approached as subtasks.

Methods

We employed machine learning together with a bag-of-words approach to assignment of Gene Ontology terms to proteins. More specifically, we train support vector machines on term vectors formed from document-protein pairs from the initially released training data. The method is based on the idea that words and terms that co-occur with the gene of interest are indicative of its function and that genes with similar co-occurences of terms will have related functions. As classification features, we use both single words and domain-specific multi-word terms automatically recognised and extracted from the corpus. We have previously shown that use of such terms as features for classification improves performance [2]

[§]Department of Biomolecular Sciences
[*]Department of Computation

(we used only terms other than protein names to be classified). We consider that descriptions of experimental procedures are unlikely to contain information on gene function and may introduce unnecessary noise. Therefore, we remove the *experimental*, *methods* and *reference* sections of each document but retain figure legends prior to processing.

In order to automatically generate terms as features, we use an enhanced version of the C/NC-value method [3]. The method combines linguistic term formation patterns and statistical analysis. Also, term variants conflation rules have been added in order to enhance the results of the statistical part and to link lexically different but terminologically equivalent term occurrences. We use orthographic and inflectional normalisation of terms, as well as a method for recognition and linking of acronyms that have been introduced in documents. Acronyms are acquired prior to the selection of the term candidates and are also mapped to their expanded forms, which are normalised in the same manner as other term candidates.

For weighting features, we use a form of inverse document frequency to the term vectors that takes account of the number of documents considered relevant to the protein-GO term pair. We also remove 300 stop words and low frequency words/terms as features.

In the training phase, for each GO term, positive learning examples are formed from all proteins with GO codes that match or are descendant from the GO term concerned. The negative training examples were generated by taking an equivalent number of positive examples from sibling GO terms and their children. This means that GO classifiers can only be trained for a sub-tree of the GO formed from root to GO terms occuring in the training data.

In the prediction/testing phase, target proteins (represented by term features from document(s) in which they appear) are tested against each of the classifiers, and the highest scoring GO codes are assigned to them. The similar approach is employed for detection of relevant passages: a set of paragraphs (as tagged in the SGML corpus) pertaining to a given protein is retrieved, and each paragraph is formed into a term vector and tested against the relevant GO classifier(s). The highest scoring passage is assigned as pertaining to the protein/GO annotation.

Task 2.1: Passage assignment to protein/GO term pairs

In order to assign a relevant passage from a given document, each passage is tested against the relevant GO classifier (as provided by assessors).

Submission One. If the GO term in the testing data does not occur amongst our classifiers, the evidence code and paragraph are left blank.

Submission Two. If the GO term in the testing data does not occur amongst our classifiers, the GO term classifier nearest in the GO term concerned (shortest path through ontology) is used and the highest scoring passage to this term is assigned as evidence.

Task 2.2: Prediction of GO codes and passage assignment

In order to predict relevant GO codes and select paragraphs, we test a term vector composed from the given document for each protein against every available GO classifier. Passage assignment was as in Task 2.1 Submission One.

Submission One. For this submission, we test a term against every GO classifier derived from the training data and assigned the top scoring GO terms to the protein. The number of assigned GO-terms is as required by the assessors.

Submission Two. This submission was similar to Submission One; however, we also trained classifiers derived from the test data in Task 2.1 in addition to those from the original training data. We reasoned that this might improve recall since our initial training data contained only a limited set of GO terms.

Task 2.3: Document retrieval, GO codes prediction and passage assignment

In order to retrieve relevent documents for a given protein, we employ an ad-hoc retrieval method. Using inverse document frequency term weighting, we score each document in the corpus against a query formed from all the words of the DE and GN fields of the Swiss-Prot/Trembl entries of each protein. If a document contains an exact phrasal match to a multiword term in the DE field, the weight contributes from this term is raised to a power proportional to the length of this match. Some manual tuning of various parameters is employed for this task based on subjective assessment of retrieved documents.

Submission One. Once documents are selected, we apply the same method to assign GO codes and extract passages as in Task 2.2 Submission One.

Submission Two. As Submission One, but including additional classifiers derived from Task 2.1 testing data (as in Task 2.2 Submission Two).

Results and Discussion

Task 2.1

Table One: Precision for Task 2.1

GO code	protein	Submission 1 524[a]		Submission 2 998[a]	
		pairs	prec.	pairs	prec.
high	high	59	11%	125	12%
general	high	28	5%	69	7%
high	general	19	4%	38	4%
Total		136	26%	232	23%

[a]Total number of predictions.

Our supervised machine learning approach means we are unable to make predictions for GO terms that occur in the testing set but not in the training set. 43% of the testing examples fall into this class and 50% of the GO terms in the testing data are absent from the training data; however, using the classifier of a near-neighbour GO term when no classifier for the actual GO term is available (Submission Two) appears to substantially improve recall without sacrificing precision (Table One).

Task 2.2

Table One: Precision for Task 2.2

GO code	protein	Submission 1 502[a]		Submission 2 485[a]	
		pairs	prec.	pairs	prec.
high	high	3	1%	16	3%
general	high	8	2%	26	5%
high	general	2	0%	2	0%
Total		13	3%	44	9%

[a]Total number of predictions.

Results for this task were dissapointing. We cannot tell whether this is because of a poor overlap between the training GO classes and those in the test data because the correct GO terms were not released at the time of writing. However, including classifiers derived from Task 2.1 improves precision and recall substantially and indicates out method might be more effective if more training data were available. Suprisingly, results for Task 2.3 are much better than for Task 2.2. We suspect this may be because our method requires a substantial quantity of relevant text for each protein to be effective. We discuss this in more depth below.

Task 2.3

Table One: Precision for Task 2.3

GO code	protein	Submission 1 36[a]		Submission 2 52[a]	
		pairs	prec.	pairs	prec.
high	high	11	31%	11	21%
general	high	7	19%	8	15%
high	general	0	0%	0	0%
Total		18	50%	19	36%

[a]Total number of predictions.

An assessment of performance here is difficult because of the limited quantity of testing data and the fact that the assessors only partially evaluated the results (5 of the 10 proteins). Results are somewhat encouraging: 50% of our assignments were considered to be highly relevant to the protein and generally or highly relevant to the GO code. For two cases (*BRCA1-associated RING domain protein* and *Synaptophysin*) we have near perfect precision, as all evaluated predictions in these cases were considered generally relevant or better.

An important facet of our system is that GO assignments are not derived from 'relevant' passages, but from relevant documents. This implies that our GO assignments may be largely accurate and with relatively good recall, but finding the relevant passage may be difficult. Almost by definition, our approach performs poorly in classifying passages as relevant to a particular GO term, since short passages contain sparse features that cannot give accurate assignments. From a biologist's point of view, short passages rarely give unambiougous assignment of function without domain knowledge gleaned from other sources; as an example, imagine an non-specialist attempting GO assignment from a short passage of dense and highly technical prose; such a task would be very difficult. Similarly, computational methods that can learn some domain knowledge from a corpus should perform well even when an explicit statement relating a protein to GO term is absent. This may be the reason that Task 2.3 results are better than those for Task 2.2.

The average number of documents retrieved for each protein in Task 2.3 was 10; thus, we frequently have a substantial body of relatively weak evidence for a GO assignment rather than a rare but explicit statment of a protein – function relationship. This implies we can often get the GO code correct but the passage may be non-relevant.

An extreme example of where we get the GO code spectacularly correct but the passage spectacularly wrong is for *BRCA1-associated RING domain protein 1* [BARD1] (Task 2.3). We accurately assign the GO term 'DNA-directed RNA polymerase II, holoenzyme' to the protein. But the following passage is deemed relevant:

> *...Fig. 6. ESI-TOF mass spectra for the binding of Zn^{2+} to the subunits of wild-type and mutant BC-112/BD-115 heterodimer complexes. Spectra shown are for wild-type (A), C39A (B), C64A (C), the sample shown in C + 40 μ M Zn^{2+} (D), the sample shown in C + 50 mM EDTA (E), and the sample shown in C + acetic acid (pH < 4.0) (F). The increased intensity of the BRCA1 subunit relative to BARD1 in panel A (asterisk) is primarily attributable to an excess of BRCA1 homodimer in this sample preparation. ...* [1]

As is obvious, this passage is not relevant to 'DNA-directed RNA polymerase II, holoenzyme'. However, BARD1 is thought to be part of a DNA-directed RNA polymerase II holoenzyme as stated in its Swiss-Prot entry.

The above example illustrates another problem with the 'one passage implies function' paradigm. Without having read Table III of the quoted document, it would be impossible to determine that BD-115 is a mutant form of BARD1. It is highly likely that this term is neologism unique to this document; thus, the accurate interpretation of this passage relies on analysis of the document as a whole. We believe this type of latent information is very common; for example a statement of interaction between two proteins implies cellular co-location and hence knowing the location of one of the pair is evidence for location of the other.

Conclusions

A machine learning approach to protein functional prediction from text using SVMs and features derived from multi-word terms and words can yield good performance if sufficient training data is available. Performance improves as the number of relevant documents to a particular protein increases.

Our method works poorly on short passages and/or single documents. We beleive this is because short passages often do not contain the necessary information to infer protein function without information from other sources. It will be interesting to see if methods that use fine-grained analysis - such as parsing - can accurately predict cases where we fail and *vice versa*.

References

[1] P.S. Brzovic, J.E. Meza, M-C. King, and R.E. Klevit. Brca1 ring domain cancer-predispoing mutations. *J. Biol. Chem.*, 276(44):41399–41406, 2001.

[2] G. Nenadic, S. Rice, I. Spasic, S. Ananiadou, and B. Stapley. Selecting features for text-based classification: from documents to terms. In *Natural Language Processing in Biomedicine, Association for Computational Linguistics 2003 Workshop*, 2003.

[3] Goran Nenadic, Irena Spasic, and Sophia Ananiadou. Terminology-driven mining of biomedical literature. *Bioinformatics*, 19(8):938–943, 2003.

[4] B. Stapley, L. Kelly, and M. Sternberg. Predicting the sub-cellular location of proteins using support vector machines. In *Pacific Symposium on Biocomputing*, pages 374 – 385, 2002.