

# Using Heuristics, Syntax and a Local Dynamic Dictionary for Protein Name Tagging

Gunnar Eriksson, Kristofer Franzén, Fredrik Olsson  
Swedish Institute of Computer Science  
Box 1263, SE-164 29 Kista, Sweden

Lars Asker, Per Lidén  
Virtual Genetics Laboratory AB  
SE-171 77 Stockholm, Sweden

## 1 Introduction

This paper presents work on a method to detect names of proteins in running text.

The detection and categorization of named entities, such as names of people, organisations and places, in classical MUC-style information extraction tasks (Borthwick *et al.*, 1998) might be regarded a solved problem. But names of proteins present a slightly different challenge because of their variant structural characteristics and the specifics of the text domains in which they appear. This certainly holds true for other biological substances, and probably for many other kinds of terminology as well.

We will present the different steps involved in our approach to this problem, and show how combinations of them influence recall and precision.

## 2 Background

The roles and functions of proteins are important study objects in many areas in the life sciences, as well as for the pharmaceutical industry. In view of the vast amount of scientific text produced in these areas, it would be useful to have methods for automatic structuring and extraction of information found therein.

One common reason for developing methods for automatic detection of protein names in text, has been the desire to build systems for automatic extraction of interactions between proteins (Blaschke *et al.*, 1999; Thomas *et al.*, 2000). However, the detection of protein names is in itself useful in, e.g., applications to support

browsing, searching and linking in abstracts from the National Library of Medicine's MEDLINE database.

## 3 Protein Names

Despite the lack of common standards and fixed nomenclatures, protein names exhibit several regularities that can be exploited in order to identify never-before-seen instances. Primarily, protein names are almost always descriptive in some way. Protein characteristics such as function (e.g. *growth hormone*), localization or cellular origin (such as *HIV-1 envelope glycoprotein gp120*), physical properties (*salivary acidic protein-1*), similarities to other proteins (*Rho-like protein*) are commonly reflected in the name. Names are also constructed using a combination or abbreviation of the above.

In this study we define a protein as a single biological entity composed of one or more amino acid chains. Protein fragments or protein families are not included in this definition. Furthermore, since names of genes and the names of their protein products are used equivocally we make no attempt to distinguish between them.

## 4 Method

Arguably, information extraction tasks are always a trade off between recall and precision, and depending on the application, one may want to focus on one or the other. When trying to extract protein interactions from MEDLINE abstracts, Thomas *et al.* (2000) claim that pre-

cision is more important, since the amount of text (11 million abstracts) suggests that if an interaction is not correctly detected in one abstract it is likely to be found in another. de Bruijn and Martin (2000), on the other hand, aim at a high recall and a fair precision, arguing that filtering techniques can be used to separate plausible hits from dubious ones.

In our case, the first application at hand was a browsing support system, to link protein names in MEDLINE abstracts to entries in SWISS-PROT (Bairoch and Apweiler, 2000). Since the intended user is likely to be a domain expert able to judge if a hyperlink actually refers to a protein, we could accept some false links. On the other hand, too many words erroneously marked as proteins would give the user sore eyes and little confidence in the system. The current algorithm initially strives for high recall with the consequence of poor precision. Later modules in the pipelined system use filtering techniques to boost precision, and a local dictionary is eventually applied to increase recall. The algorithm can be described as consisting of the following six steps, of which the first two and part of the third are an implementation of some of the heuristic steps in the algorithm described by Fukuda *et al.* (1998).

#### 4.1 Tagging feature terms

Feature terms are words that describe the function or characteristics of a protein, e.g., *receptor* and *enzyme*. We currently tag words as feature terms if we find them in our list of about 47 such words.

#### 4.2 Tagging core terms

Core terms are either words ending in *-ase* and *-in*, or strings with characteristics typical of protein names, i.e., strings containing instances of upper case letters or numbers, found in names of proteins like *HsMad2* and *U3-55k*. Two general filters is applied to these terms to avoid overgeneration: Words consisting of  $\geq 50\%$  non-word characters, and measuring units are discarded

as core terms .

#### 4.3 Applying filters and knowledge bases

To remedy the low precision obtained in the previous step, a set of filters is applied to get rid of false hits. Some filters use regular expression patterns of word suffixes to rule out, e.g., names of chemical substances. Other filters use patterns of whole words/expressions to filter out, e.g., personal names and other parts in bibliographical references, chemical formulas, arithmetic expressions, and amino acid sequences. A third group of pattern matching filters remove the core term annotation on words unlikely to function as core terms: Words,  $\geq 6$  characters long consisting solely of upper case letters, or consisting of upper case letters and more than one hyphen are discarded.

Short core terms ( $\leq 3$  characters) get special treatment. Only those found in our short-protein-name knowledge base drawn from SWISS-PROT are considered core terms. All the others are tagged as potential core terms to be used later in the protein name identification process. Core terms resembling regular proper names are treated the same way.

Furthermore, as all protein names do not conform to the patterns above, words are dubbed core terms if they are found in a list of established protein names like *interferon*.

#### 4.4 Finding noun phrases

This step takes advantage of the Functional Dependency Grammar (FDG) parser from Conexor Oy (Tapanainen and Järvinen, 1997). For every noun phrase, we identify its preceding lexical modifiers. These minimal noun phrases, the noun phrase without any subordinate NP, are considered potential locations for protein names.

#### 4.5 Identifying protein names

In this step we select as protein names all minimal noun phrases that contain a core term, special combinations of feature terms, or special

combinations of feature terms and words tagged as potential core terms.

#### 4.6 Applying a local dynamic dictionary

All combinations of relevant terms used to identify protein names in the previous step are stored in a local dictionary as regular expressions. This dictionary is used in an additional tagging pass of the document to make possible fuzzy matching of proteins in noun phrases undetected or misinterpreted by the parser.

### 5 Evaluation

Tagging of protein names in running text is cumbersome even for human domain experts, and evaluation of a protein tagger requires a tagged corpus. For an exhaustive discussion on the problems of building annotated corpora for the molecular-biology domain, and results on inter-annotator agreement, cf., Tateisi *et al.* (2000). At this point we can present results of our system applied to a corpus of 99 MEDLINE abstracts containing 1745 protein names tagged by our domain experts.

The aim of the current evaluation, which is performed on data used for reference during development, is to see how much each combination of the steps described in 4.3 and 4.6 contributes to the final result.

All four cases described below include the same way of tagging feature terms and core terms, employing the FDG parser to find minimal noun phrases, and mechanisms for identifying protein names.

The intent of using a local dynamic dictionary is to increase recall. Contrary to our intuition, Table 1 illustrates that precision did not seem to drop severely even though recall increased with 10.2% and 13.9% (first and second row in Table 1, respectively) when toggling the use of a local dynamic dictionary as regards the use of external filters and knowledge bases.

	NO LDD	LDD
NO FKB	$R = 88\%$ $P = 60\%$ $F = 72.01\%$	$R = 97\%$ $P = 59\%$ $F = 73.64\%$
FKB	$R = 79\%$ $P = 75\%$ $F = 77.60\%$	$R = 90\%$ $P = 74\%$ $F = 81.76\%$

Table 1: Results varying along Local Dynamic Dictionary (LDD) and Filters and Knowledge Bases (FKB), given in recall ( $R$ ), precision ( $P$ ), and F-score ( $F$ ).

### 6 Further Work

In the next couple of weeks we will be able to present results from running our system on a previously unseen corpus annotated according to our protein definition and on the GENIA corpus (Tateisi *et al.*, 2000), as well as a comparison between our system and the KEX system (Fukuda *et al.*, 1998).

### References

- Bairoch, A. and Apweiler, R. 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucl. Acids. Res.*, **28**:45–48.
- Blaschke, C., Andrade, M. A., Ouzounis, C., and Valencia, A. 1999. Automatic extraction of biological information from scientific text: protein—protein interactions. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*, pp. 60–67, Heidelberg, Germany, August 6–10.
- Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. 1998. NYU: Description of the MENE Named Entity System as used in MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, VA, USA, April 29 - May 1.

- Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T. 1998. Toward Information Extraction: Identifying Protein Names from Biological Papers. In *Proceedings of the Pacific Symposium on Biocomputing (PSB'98)*, pp. 705–716, Maui, Hawaii, January 4-9.
- Tapanainen, P. and Järvinen, T. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pp. 64–71, Washington D.C., April. Association for Computational Linguistics.
- Tateisi, Y., Ohta, T., Collier, N., Nobata, C., and ichi Tsujii, J. 2000. Building an Annotated Corpus in the Molecular-Biology Domain. In *Proceedings of Workshop on Semantic Annotation and Intelligent Content*, Centre Universitaire, Luxembourg, August. ACL. The workshop was held in conjunction with the 18th International Conference on Computational Linguistics (COLING-2000).
- Thomas, J., Milward, D., Ouzounis, C., Pulman, S., and Carroll, M. 2000. Automatic Extraction of Protein Interactions from Scientific Abstracts. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2000)*, pp. 538–549, Oahu, Hawaii, January 4-9.
- de Bruijn, B. and Martin, J. 2000. Protein Name Tagging. Presented as a poster at the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB'00).