JAMIA Original Investigations

Research Paper ■

A Simple and Practical Dictionary-based Approach for Identification of Proteins in Medline Abstracts

SERGEI EGOROV, PHD, ANTON YURYEV, PHD, NIKOLAI DARASELIA, PHD

Abstract **Objective**: The aim of this study was to develop a practical and efficient protein identification system for biomedical corpora.

Design: The developed system, called ProtScan, utilizes a carefully constructed dictionary of mammalian proteins in conjunction with a specialized tokenization algorithm to identify and tag protein name occurrences in biomedical texts and also takes advantage of Medline "Name-of-Substance" (NOS) annotation. The dictionaries for ProtScan were constructed in a semi-automatic way from various public-domain sequence databases followed by an intensive expert curation step.

Measurements: The recall and precision of the system have been determined using 1,000 randomly selected and hand-tagged Medline abstracts.

Results: The developed system is capable of identifying protein occurrences in Medline abstracts with a 98% precision and 88% recall. It was also found to be capable of processing approximately 300 abstracts per second. Without utilization of NOS annotation, precision and recall were found to be 98.5% and 84%, respectively.

Conclusion: The developed system appears to be well suited for protein-based Medline indexing and can help to improve biomedical information retrieval. Further approaches to ProtScan's recall improvement also are discussed.

■ **J Am Med Inform Assoc.** 2004;11:174–178. DOI 10.1197/jamia.M1453.

Research in biology in the past decade has generated a large volume of protein function data, which are stored in the textual form in databases such as Medline. As a first step toward automatic extraction of this information, reliable methods of protein name identification are required. However, finding gene and protein names in natural language text is difficult. The lack of uniform nomenclature has resulted in discordant naming practices across different scientific communities.

Recently, a number of methods for identifying protein names in biomedical documents have been proposed. They vary in their degree of reliance on dictionaries, statistical- or knowledge-based approaches, and manual versus automatic rule generation. All methods can be roughly split into three categories: dictionary-based approaches, rule-based approaches, and machine-learning approaches, although some interesting mixed systems have been described.

Rule-based systems rely on a set of expert-derived rules, which usually combine surface clues (e.g., word alphanumerical composition, presence of special symbols, and capitalization) with word syntactic and semantic properties, to initiate, extend, and terminate the chains of sentence tokens. Some systems can also use small dictionaries of positive and negative terms to improve precision and recall. Examples of rule-based systems are presented in Narayanaswamy et al.[1] (precision 96%, recall 62%), Fukuda et al.[2] (precision 40%, recall 40%), and Franzen et al.[3] (precision 68%, recall 66%). Seki and Mostafa[4] used surface clues to anchor a protein name, but instead of syntactic features they used word first-order transition probabilities learned from annotated test corpora to extend the original match. The reported precision and recall rates are 60% and 66%, respectively.

Machine-learning approaches rely on the presence of an expert-annotated training corpus to automatically derive the identification rules by means of various statistical algorithms. The features used in machine-learning methods are mostly the

Affiliation of the authors: Ariadne Genomics, Inc., Rockville, MD.

Supported in part by NIH 1R43-GM067276. Some of the authors of the article are stockholders of Ariadne Genomics, Inc., a small private company whose business model is development and sales of bioinformatics software and, in particular, pathway analysis software. The described algorithm for identification of protein names is one component of the protein function information extraction pipeline, which, in turn, is part of PathwayAssist, the software product for pathway analysis (Bioinformatics. 2003;19:2155–7). The authors do not directly receive any royalties or fees from the sales of the described algorithm; however, the sales of the entire software product provide the significant portion of the company's income.

Correspondence and reprints: Nikolai Daraselia, PhD, Ariadne Genomics, Inc., 9700 Great Seneca Highway, Rockville, MD 20850; e-mail: <nikolai@ariadnegenomics.com>.

Received for publication: 09/08/03; accepted for publication: 01/11/04.

same as those in rule-based approaches—surface clues, parts of speech, and, sometimes, semantic word properties obtained from rough classification. Nobata et al.[5] used Bayesian classifier and decision tree algorithms to identify a noun phrase as a protein, based on its word composition. They report an F-score of 70% to 80% for protein detection. Collier et al.[6] used a first-order hidden Markov model (HMM) trained on annotated corpus to detect the protein names in text and report a 76% F-score. Kazama et al.[7] applied support vector machines to the same problem and achieved a 65% F-score.

An interesting combination of a machine-learning approach with hand-crafted rules is reported in Tanabe and Wilbur.[8] As a first step, the transformation-based part-of-speech tagger has been trained on the corpus of Medline sentences with hand-marked gene occurrences to induce the rules for tagging the text. Next, a complex set of manually derived contextual, morphologic, and dictionary-based post-processing rules have been applied. Reported precision and recall are 86% and 67%, respectively.

Dictionary-based approaches utilize a provided list of protein terms to identify protein occurrences in a text, usually by means of various substring matching techniques. Proux et al.[9] used a Drosophila protein dictionary derived from FlyBase for identification of proteins with 91% precision and 94% recall. However, they recognized only single-word protein names. They also reported that precision of the system dropped from 91% to 70% when transferred from a corpus of sentences from FlyBase to a more general set of Medline articles. An interesting combination of the dictionary-based approach with the Basic Local Alignment Search Tool (BLAST)-based identification algorithm has been proposed by Krauthammer et al.[10] The basic idea was to perform an approximate string match after converting both input text and a dictionary into the DNA sequence-like strings. The authors reported 79% recall and 72% precision.

Compared with rule-based approaches, dictionary-based protein identification systems are more accurate, and their performance is in direct correlation with the quality and completeness of the provided protein dictionaries. It usually is argued that development and maintenance of comprehensive protein name dictionaries are nontrivial tasks because new genes are constantly identified. However, both machine-learning and rule-based approaches also require significant amounts of expert work for creation of rules and manual tagging of the training corpus, respectively. In our opinion, creation of a practical information extraction system requires a dictionary-based approach for the following reasons. First, no machine-learning or rule-based system has yet achieved recognition accuracy sufficient for text-indexing or information-extraction tasks. Second, protein lists grow incomparably slower than protein function information. Third, the situation with terminology usage tends to improve. For instance, in human protein function domain, the standard HUGO (Gene Nomenclature Committee) nomenclature is used increasingly frequently instead of historical protein names. Comprehensive gene indexes also are being created for other completely sequenced organisms. However, the most important reason is that the goal of a protein function extraction is to link protein function information to the sequence and expression data. From this viewpoint a "potential protein name" identified in text is completely useless.

We have implemented a dictionary-based protein name identification system called ProtScan. It utilizes carefully constructed dictionaries of mammalian protein names to identify protein names in Medline abstracts.

Methods

The Dictionary

Our approach utilizes a combination of curated and non-curated (or "raw") protein name dictionaries for protein identification. The dictionaries were compiled on a basis of LocusLink database and additionally enriched by incorporating protein names, aliases, descriptions, and gene names from the linked GenBank, GoldenPath, and HUGO database entries. Main sources of useful names were GenBank gene name, HUGO description, HUGO symbol, HUGO aliases, GoldenPath gene name, and LocusLink official and preferred symbols and aliases. The resulting collection of protein "descriptors" contained, along with correct protein names, functional key words (e.g., "kinase"), clone names, and some completely irrelevant contaminant words and phrases. To improve the quality of this collection, the occurrence of each of the potential protein names in the 2003 Medline release was determined by the method described below, and erroneous names were removed manually from the top 20,000 entries sorted by occurrence. In the same curation process, some other names with high chance of false identification have been placed in a separate "raw" dictionary. The rest of the entries were automatically processed to:

- Remove records containing a single word with a length of 1 (e.g., "A," "C").
- Move records containing a single word with length 2 (e.g., "AS," "ET") to the "raw" dictionary.
- Move entries with length 3 or 4 not containing at least one digit (e.g., "AHH," "ATDC") to the "raw" dictionary.
- Remove purely numerical entries (e.g., "3742643").
- Remove entries consisting only of measures (e.g., "23 kDa protein").

The resulting curated dictionary then was verified for uniqueness of names by the procedure used in the name detection algorithm, and about 800 of colliding protein names were analyzed manually and either removed or resolved. The most frequent sources of colliding names were found to be multigene families. In some cases, the generic family name was included as one of the aliases to each member of the family (for example, all four human alcohol dehydrogenases ALDH1-ALDH4 contained aliases "alcohol dehydrogenase" and "ALDH"). Such generic aliases simply have been removed. In other cases, the ambiguity resulted from the changes in protein family nomenclature (e.g., "CYP2C" is an alias of "CYP2C18" and is listed as an old name of "CYP2C17")—in such cases, the aliases were kept for the most recent HUGO-approved records only. Finally, there were some truly ambiguous gene names (for example, "CNR2" is an official symbol of "cannabinoid receptor 2" and is also listed as an alias for "protocadherin alpha 6")—those were removed from the dictionary.

The remaining 245,248 records describing 81,915 unique proteins constitute our final curate dictionary, whereas all the separated records (a total of 20,115) constitute a raw dictionary. Both dictionaries have a numerical identifier (we use LocusLink-based IDs) associated with each name. At the end of the protein recognition procedure, this identifier is inserted as a markup in the original text. The combined efforts spent on creation and curation of the dictionary comprise about 80 person-hours.

Tokenization Algorithm

Overcoming traditional problems of dictionary-based approaches required us to investigate variations in protein name spelling so that they may be factored out in the matching process. We identified several classes of variations that can be taken into account by the automatic pattern matching process:

- Special characters such as hyphen, slash, and brackets are used as separators in different combinations by different authors.
- Parts of the names can be spelled in upper case by some authors and lower case by others.
- There is a variability in ways to separate name constituents: they can be put together or separated by white space or punctuation.

Our approach is to ignore these variations and use a single specialized tokenization process for target text and dictionary entries. Tokenization converts the input text into a sequence of tokens; tokens are made from the longest sequences of characters belonging to the same class. The ProtScan considers each punctuation character as belonging to a separate class; all letters belong to alphabetical class and all digits to numerical class. White space is treated as a token separator and is not considered a token. Numerical and punctuation sequences are converted into tokens with no special processing; alphabetical sequences are first converted to lower case and then searched for prefixes and suffixes made of English spelling of Greek letters ("alpha," "beta," "gamma," etc.). If such prefixes or suffixes are identified, they are stripped off and treated as separate tokens.

An example of the tokenization process is shown below.

Sentence: Here we show that ASFV IAP1 is also able to activate the transcription factor NF-kappaB.

Tokens: [here] [we] [show] [that] [asfv] [iap] [1] [is] [also] [able] [to] [activate] [the] [transcription] [factor] [nf] [-] [kappa] [b]

The described tokenization procedure is followed by a simple and efficient subsequence search applied to token sequences.

Identification of Protein Names

Protein names are identified by a variation of string search algorithm. Both curated and raw protein-name dictionaries contain names of proteins to be identified together with the corresponding IDs. A single protein usually has multiple entries in the dictionary; all these entries are labeled with the same ID, corresponding to the described protein.

When the dictionary is loaded, all entries are tokenized by the algorithm described above. When all tokens are identified, they are filtered by removing tokens belonging to a special small dictionary of *excluded* words given in bold typeface in square brackets [**′ ( ) + , - . / : cdna chain class clone disease family form gene isoform member molecule mrna of polypeptide precursor preproprotein**]. This dictionary includes punctuation tokens and frequently used words that usually do not change the identity of the protein being described and, thus, have no use for recognition process.

The remaining tokens are collected into a *token sequence* that serves as a "cleaned up" representation of the original entry. The ProtScan uses this concatenated token sequence as a key in a hash table, storing the corresponding ID as the value. If more than one ID is associated with the same token list, it is marked as *ambiguous*, and the list of IDs is stored as the corresponding value. As it was described, the curated dictionary has been edited so as not to contain any ambiguous entries; ambiguous entries present in the raw dictionary are used only in Name-of-Substance (NOS)-specific processing described below.

The raw and curated dictionaries are loaded into the same hash table. When both dictionaries contain entries with the same token list, the entry from the curated dictionary takes precedence; such a conflict is not considered an ambiguity. IDs coming from the raw dictionary are marked so that the source of an entry can be identified if needed.

In addition to the combined internal dictionary in a hash table form, the loading process creates a separate token set that consists of all unique tokens ever used in dictionary token sequences. This set of "protein words" allows ProtScan to effectively determine whether the given token in text can be a part of a protein name.

When both dictionaries are loaded, the ProtScan proceeds by processing the input abstracts. The target text (Medline abstract) is processed one sentence at a time. Breaking an abstract into individual sentences is the first step performed by ProtScan. It is done using a simple algorithm that looks for a dot followed by an upper-case letter while balancing the nested brackets and parentheses. To implement the original idea of "relaxed" matching, Medline abstracts and dictionary entries are processed by the same tokenizer and are scanned for the presence of uninterrupted token sequences, consisting only of tokens from the set of "protein words" (defined above) together with excluded words. In each case, the longest possible sequences are considered. Each such sequence, containing at least one token from the "protein words" set, is a potential protein name (or multiple names). There is no need to consider tokens outside these sequences; limiting the search to fragments of sentences improves the performance of the ProtScan, making the sequence search algorithm linear in the length of the sentence (it is still quadratic in the length of each token sequence, but they are usually short). In the example given below, all tokens included into assembled token sequences are shown in boldface type, and within each sequence, tokens not belonging to the excluded dictionary are underlined:

> "**Poly (A**) **polymerase**, the **enzyme** responsible **for poly** (**A)** addition to **primary** transcripts, contains **multiple** consensus phosphorylation sites **for p 34 (cdc 2)** ."

When the token sequence is assembled, it is required that the corresponding original character sequence is not immediately preceded by a word or a number with no separating white space and does not end in a word or a number not immediately followed by a punctuation sign (, ; . ?). Next, each token sequence is passed through a validation step to check if it satisfies the following constraints:

- comma (,) is not allowed as first or last token.
- comma is allowed between single quote (′) and a number.

- comma is allowed between a word (alphabetical token) of length > 1 and "a."
- comma is allowed between two alphabetical tokens/ numbers, second of which is not "a."
- comma is not allowed in other cases.
- slash (/) is only allowed between + and −.
- period (.) should not be followed by white space.

Each qualifying token sequence is searched for the presence of dictionary entries by trying all of its subsequences from long to short and from left to right. Each subsequence is tested by calculating its hash value in a way similar to the one used at dictionary loading: tokens belonging to the excluded words dictionary are ignored and the remaining "meaningful" tokens (shown as underlined in the above given example) are concatenated in a single string, which then is looked up in the dictionary hash table by calculating its hash value and scanning through the respecting hash table entries. If the lookup results in positive identification, the subsequence is marked up, and the rest of the tokens to its right are searched for more matches. In the presented exemplary sentence, two identified subsequences are "**Poly (A) polymerase**" and "**p 34 (cdc 2)**."

The result of dictionary lookup is one or more protein IDs that may have come from any of the two original dictionaries. At this moment, the algorithm behaves differently for raw and curated entries.

NOS-independent Identification

The protein name identification is based primarily on the use of the curated dictionary. If a token sequence from the input sentence is mapped to "curated" ID, this ID is used in the output. There is no need to consider alternatives, because the curated dictionary is free of ambiguities, and curated entries supersede any raw entries with the same hash key. The resulting protein ID is inserted into the output of the ProtScan as markup of the original text.

NOS-dependent Processing

Raw dictionaries contain ambiguous entries; if the dictionary lookup of the token substring returns one or more "raw" IDs (i.e., IDs coming from the "raw" entries), these IDs should be validated and disambiguated. Note that in this situation no "curated" IDs can be present in the mix because of the way dictionaries are loaded. To do this, the ProtScan relies on the NOS fields annotating Medline abstracts because they are frequently present in Medline abstracts, have a good quality, and refer to the proteins mentioned in the corresponding abstracts and thus provide the context needed to improve matching of the noncurated part of the dictionary. Each NOS field associated with the given abstract is looked up in the protein dictionary hash table in the usual manner; all the IDs, found from all the associated NOS fields, are collected in a single *validated ID list*, associated with the abstract. We use both protein dictionary entries at this step because this identification is for context purposes only. This ID list is used as a filter to disambiguate the results of identification performed on a basis of the raw dictionary.

The disambiguation proceeds as follows. First, all identified IDs for a given token substring that came from the raw dictionary and are not present in the *validated ID list* are discarded. Second, the remaining IDs are counted and if more than one ID remains, the disambiguation failed, and no results will be returned. If in the end we have exactly one protein ID, we consider it validated by NOS. The resulting ID is inserted into the output of the ProtScan as markup of the original text.

Local Abbreviations

The ProtScan also takes advantage of local abbreviations and alternative names sometimes provided in the text along with a full protein name. These abbreviations can also result in a protein match even if they are not in the dictionary. Similar to the previously proposed approaches (Schwartz and Hearst[11] and Chang et al.[12]), the ProtScan looks for *abbreviation definitions* in parentheses immediately following an identified protein name. To be considered an abbreviation, the contents of the parentheses should satisfy the following constraints: it should be a sequence of 2...5 characters, starting with an upper case letter and consisting of upper case letters, digits, and dashes (-), not ending in a dash. In the following example: *"ID{10914=Poly(A) polymerase} (**PAP**), the enzyme responsible for poly(A) addition to primary transcripts, contains multiple consensus phosphorylation sites for ID{983=p34(cdc2)}"*, identified protein names are tagged with numerical protein identifier (**ID{id=** ... **}**), whereas the identified abbreviation (PAP) is shown in boldface type.

When an abbreviation definition is recognized, it is added to the list of local abbreviations associated with the current abstract. The ID for the abbreviation is taken from the markup preceding the definition. All local abbreviations are used only in the scope of an abstract in which they were found.

The last pass of the ProtScan goes through unmarked parts of each sentence and looks for *literal* references to local abbreviations. This pass uses character-by-character case-sensitive matching and does not ignore punctuation or white space; the match should be delimited by punctuation or white space on both ends.

When an abbreviation use is identified, its associated protein ID is inserted into the output of the ProtScan as markup of the original text.

Evaluation

The ProtScan was evaluated by running it against a gold standard, which was created by manually labeling protein names in 1,000 randomly selected Medline abstracts. The gold standard was created by an expert in molecular biology (AY). In evaluation, two system parameters were determined—precision (percentage of correctly recognized protein names out of all manually labeled names) and recall (percentage of correctly recognized names out of all occurrences identified by the ProtScan). Notably, some protein names incorporate names of other proteins (for example, "p53-interacting protein"). The protein occurrence was considered to be correctly identified only if both its left and right boundaries were determined correctly by ProtScan, and it was assigned a correct ID.

Results

The described system is implemented in C programming language and optimized for speed. It is capable of processing approximately 300 abstracts per second on a 600-MHz Pentium III machine with 512M of memory.

The gold standard data contained 1,914 protein names manually labeled in 1,000 randomly selected Medline abstracts. The distribution of number of protein names per abstract was as follows: 688 abstracts contained no proteins, 215 abstracts contained one to five proteins, 53 abstracts contained six to ten proteins, 38 abstracts contained 11 to 20 proteins, and six abstracts contained more than 20 proteins. In our performance test experiments, ProtScan has identified 1,730 protein name occurrences, of which 1,696 were identical to the manually labeled names. This corresponds to 88.6% recall and 98% precision. The 2% of incorrectly identified names corresponded to either partially matched protein names or nonprotein abbreviations incorrectly recognized as protein names. We also have estimated the ProtScan's performance on the more general set of documents (which were assumed to be different from Medline abstracts in lacking NOS annotation) by running the system against the same gold standard described above but with empty raw dictionary. In this experiment, ProtScan identified 1,632 protein names, of which 1,608 were identical to the manually labeled ones (84% precision, 98.5% recall). However, when the content of the raw and curated dictionaries was combined into a single dictionary, which was used instead of the original "curated" one, nearly twice as many protein occurrences were identified—recall increased to 93%, whereas precision went down to 51%. This is not surprising, because the raw dictionary is mostly populated with short names that cannot unambiguously identify individual proteins without proper context information. However, disabling the local abbreviation identification algorithm resulted in only insignificant reduction of recall (85.2%) and did not affect precision when compared with the original NOS-dependent experiment.

Discussion

The manual inspection of the 218 missed protein names has found three main reasons for protein identification failure. The first and most obvious one (responsible for 22, or 10.1% of all missing names) is the absence of the protein name in the curated dictionary or its presence in a raw dictionary when no appropriate NOS annotation was present in Medline abstracts. Second, in 65 cases of 218 (29.8%), "collective" protein names were used in text instead of names of individual protein family members or instead of names of individual subunits in multisubunit proteins. For example, there are four human alcohol dehydrogenase genes, whereas in Medline, alcohol dehydrogenase is frequently cited without mentioning a particular gene. Similarly, some abstracts may cite "caspases" as an aggregate name for this multigene family. The third and most frequent reason for failure (131, or 60.1% of all missed names) was attributed to a widely used method of citing a group of related proteins by separating the common name subpart ("caspase 5, -8, and -9," "ERK1/2," or "IL1 and IL2 receptor**s**"). The latter example also results in an incorrect recognition of IL1 as the mentioned protein, whereas the IL1 receptor is actually described. To overcome these identification problems, we are developing a "partial-match" algorithm that will match a token sequence not to an individual protein but to the list of (possibly related) protein IDs. We are also testing an approach of using the match from the curated dictionary as a validation of the raw dictionary match in addition to the NOS-based validation. This can increase the recall of the system, because the NOS annotation of the Medline abstract usually covers only proteins "central" to the article and not other proteins mentioned in relation with them.

Conclusion

We have built and evaluated a dictionary-based mammalian protein identification system that has a 98% precision and 88% recall when applied to Medline abstracts. On a more general set of biomedical documents (lacking NOS annotation), the precision and recall of the system were estimated to be 98.5% and 84%, respectively. Disabling the local abbreviation identification algorithm in the NOS-dependent protein identification resulted in 85.2% recall and 98% precision, whereas combined raw and curated dictionaries, used instead of curated, resulted in 93% recall and 51% precision. It also seems that after initial efforts required for protein name dictionary construction, its maintenance and updating will be a much easier task requiring significantly less human intervention. In addition, automatic tools can be developed to aid identification of new protein name candidates in Medline, which then can be subjected to expert curation. We believe that in its current form, ProtScan can be used for protein-based Medline indexing and can help to improve biomedical information retrieval.

References ■

1. Narayanaswamy M, Ravikumar KE, Vijay-Shanker K. A biological named entity recognizer. Proc Pacific Symp Biocomput. 2003;8:427–38.
2. Fukuda K, Tsunode T, Tamura A, Takagi T. Toward information extraction: identifying protein names from biological papers. Pac Symp Biocomput. 1998;3:707–8.
3. Franzen K, Eriksson G, Olsson F, Asker L, Linden P, Coster J. Protein names and how to find them. Int J Med Inf. 2002;67: 49–61.
4. Seki K, Mostafa J. A Probabilistic Model for Identifying Protein Names and Their Name Boundaries. Stanford, CA: IEEE Computer Society Bioinformatics Conference, 2003.
5. Nobata C, Collier N, Tsujii J. Automatic term identification and classification in biology texts. Proc Natural Language Pacific Rim Symposium. 1999:369–75.
6. Collier N, Nobata C, Tsujii J. Extracting the Names of Genes and Gene Products with a Hidden Markov Model. Proc Intl Conf Comput Linguistics. 2000;18:201–7.
7. Kazama J, Makino T, Ohta Y, Tsujii J. Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain. 2002:1–8.
8. Tanabe L, Wilbur J. Tagging gene and protein names in biomedical text. Bioinformatics. 2002;18:1124–32.
9. Proux D, Rechenmann F, Julliard L, Pillet VV, Jacq B. Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. Genome Inform. 1998;9:72–80.
10. Krauthammer M, Rzhetsky A, Morozov P, Friedman C. Using BLAST for identifying gene and protein names in journal articles. GENE. 2001;259:245–52.
11. Schwartz AS, Hearst MA. A simple algorithm for identifying abbreviation definitions in biomedical text. Pac Symp Biocomput. 2003;8:451–62.
12. Chang JT, Schtze H, Altman RB. Creating an online dictionary of abbreviations from Medline. J Am Med Inform Assoc. 2002;9:612–20.