

PathwayFinder: Paving The Way Towards Automatic Pathway Extraction

Daming Yao¹, Jingbo Wang², Yanmei Lu³, Nathan Noble², Huandong Sun², Xiaoyan Zhu², Nan Lin⁴, Donald G. Payan⁴, Ming Li⁵, Kunbin Qu⁶

¹ School of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

^{2,5} Computer Science Department, University of California at Santa Barbara, Santa Barbara, California, CA 93106, USA

³ Genentech Inc, 1 DNA Way South San Francisco, CA 94080, USA

^{4,6} Rigel Pharmaceuticals Inc, 240 East Grand Avenue, South San Francisco, CA 94080, USA

Corresponding authors: 5 and 6.

Contact: dyao@uwaterloo.ca, mli@cs.ucsb.edu, kqu@rigel.com

Abstract

Automatically mining protein pathway information from the vast amount of published literature has been an increasing need from the pharmaceutical industry and biomedical research community. This task has been proved to be a formidable one. Many systems have been implemented, but few are practical. Some are too restricted and some are overly ambitious. ¹This paper presents the PathwayFinder system with two key innovations that give the system simultaneously generalization power and practical capabilities: (a) PathwayFinder is designed with appropriate level of users' involvement on information extraction, based on the authors' belief that totally automatic pathway retrieval is beyond the current technology; (b) A novel multi-agent architecture is built to support the need of user-computer interactions and domain extensions. As a result, PathwayFinder is flexible, easy to use, and extendable to be customized to other domains. We have successfully applied the PathwayFinder system to study the ubiquitin cascade pathway.

Keywords: Pathway extraction, multi-agent, user-involved extraction, natural language processing, ubiquitin pathway

1 Introduction

There are currently over 12 million PubMed citations dating back to the mid-1960's from the PubMed service at NCBI. The number of published papers is growing at an exponential rate in the last several decades. In some intensively studied research areas, such as the ubiquitination pathway cascade, it becomes impossible for a single person to grasp all the concepts and ideas from the massive information.

Normally, the most important information within raw texts is the relationship among the entities, such as "protein A is activated by protein B" and "small molecule C can inhibit such a process". This kind of information is critical to the process of drug development: from early target identification, to target validation, to lead profiling, and finally to lead development and pre-clinical studies. It serves as the basis for experiment design, assay development, project planning and decision-making. Effectively extracting such relationships from bulky raw texts to structure them into a standard format and turn them into accumulated knowledge is the key to supporting an efficient, rapid drug development process, which is the goal of our project.

Recently there have been a number of projects aimed at conducting Information Extraction (IE) automatically in the biomedical domain. For a detailed review, please refer to the **Background** section of this paper. In general, there are three categories of IE systems. The first one uses simple statistical methods, such as term cooccurrence, to identify protein and gene names. The second category applies Natural Language Processing (NLP) techniques, such as part-of-speech tagging and grammar parsers to handle complex sentences. And the third category applies more sophisticated natural language technologies that can handle anaphora as well as extracting a broader range of information.

One major weakness of the current three categories of IE systems is the insufficient involvement of users. More accurately, although some systems provide ways to interact with users, the objective has always been over-ambitious to achieve fully automated IE systems without users' intervention, which is not practical for the following reasons:

1. Currently, NLP is not accurate enough, especially in biology literature, due to the complexity of sentence structures;
2. The credibility of the results are not properly reflected, which decreases further utilization;
3. Some crucial information is not extracted, such as conditions of interactions;

4. The demands from biologists are varied and constantly changing, to which systems can not adapt promptly.

To our knowledge, none of the current pathway extraction systems is widely accepted by biologists. To fill this gap, we introduce the user-involved extraction in the PathwayFinder system with the belief that the extraction process is a collaborative task involving both the users and the extraction system. User-involvement features enable “training by using”, which means users can start to use PathwayFinder without training data. With accumulating of data, term cooccurrence and feature cooccurrence will refine the extraction parameters and keep the system running autonomously. They will also counteract the adverse effects of user-introduced-errors.

To support the user-involvement features, an innovative multi-agent architecture is introduced. Different types of agents are created to distinguish users and manage extraction tasks in sub-domains, which not only improves the performance of extraction, but also isolates the erroneous information brought in by particular users. The current versions of the PathwayFinder agents are implemented in Java, which makes them platform-independent.

1.1 Background

Among the relationships of biomedical entities, protein-protein interactions are the most basic ones for many biological processes and building blocks of cellular pathways. A large part of this information is embedded in a large amount of biology literature. For this reason, some systems have been developed to extract protein-protein interactions automatically, which can be classified into the following groups:

Online protein-protein extraction systems (Blaschke *et al.*, 1999, Ng and Wong, 1999, Wong, 2001). Similar to information retrieval systems, these systems require users to input keywords—one or several protein names, and retrieve related abstracts or papers from PubMed or other text sources according to the keywords. Then, the systems process the retrieved documents with the aid of preset internal patterns, extract the interactions and present them to the users. Online extraction systems are quite useful for users to search for particular targets. However, the pattern set used is small and pre-defined, which limits the performance.

Systems targeting a particular sub-domain (Ono *et al.*, 2001, Humphreys *et al.*, 2000, Rindflesch *et al.*, 2000). These systems normally have pre-conditions, such as a well-defined protein name dictionary or a pre-retrieved data set with the protein names. At the present stage, these systems need substantial efforts to achieve similar performance to that in their original domain, which hinders them from being practical applications, even with claimed high precision and recall rates.

Systems migrated from general IE systems (Humphreys *et al.*, 2000, Thomas *et al.*, 2000). The templates and rules for these systems are manually customized to fit biological sub-domains. Due to the fast development in biology and bioinformatic fields, substantial efforts are needed even for moderate performance.

General interaction extraction systems (Proux *et al.*, 2000, Leroy and Chen, 2002, Friedman *et al.*, 2001). Some systems emerging in recent years target general interaction extraction from biology literature. These systems normally have well-defined patterns and strong language processing ability. The present results are promising: Friedman *et al.* (Friedman *et al.*, 2001) claim to have 96% of precision and 63% of recall rates for a paper they tested.

A lot of efforts have also been put into individual components of the extraction procedure, such as document retrieving and clustering (Iliopoulos *et al.*, 2001, Marcotte *et al.*, 2001), protein interaction database construction (Sanchez *et al.*, 1999), protein name recognition (Fukuda *et al.*, 1998, Tanabe and Wilbur, 2002, Collier *et al.*, 2000, Hatzivassiloglou *et al.*, 2001), tagging (Brill, 1995), parsing (Park *et al.*, 2001, Leroy and Chen, 2002), verb and pattern discovery (Hatzivassiloglou and Weng 2002), extraction rule development (Krauthammer *et al.*, 2002, Oyama *et al.*, 2002), gene clustering (Chaussabel and Sher 2002) and pathway visualization (Ng and Wong, 1999, Wong, 2001).

To date, most of pathway extraction systems, except GeneWays (Friedman *et al.*, 2001), use fixed patterns related with only a small number of verbs (“interact”, “activate”, “bind”, “inhibit”, “associate”, etc.). GeneWays manually collects many more patterns with a larger verb set and classifies them into 14 semantic classes. Since a few patterns can not cover all scenarios, more effort needs to be made on pattern definition to increase the recall rate.

Statistical methods such as Hidden Markov Models and the Maximum Entropy algorithm, which yield good results in general information extraction, start to be used in different components of biological extraction systems, such as the protein name identification (Collier *et al.*, 2000) and document classification (Raychaudhuri *et al.* 2002). Currently, these methods need large training sets, and respond slowly to the changes.

More recently, a small number of experts are involved in some systems to bring in their knowledge and judgement (Donaldson *et al.* 2003, Leroy and Chen 2002, Libbus and Rindflesch 2002). This is a promising progress, since humans do better on some tasks, and proper combination of computer and human is one of the best ways to achieve a practical system. However, we believe that, with proper mechanisms to deal with erroneous information, more general users need to be involved in extraction procedure, since “users know better about what they want”. With this belief, we introduce the concept of “user-involved extraction” in PathwayFinder system, which is supported by an innovative multi-agent architecture.

2 Multi-Agent Architecture

User-involved extraction demands not only the flexibility of extraction procedure, but also the flexibility to fit the diverse requirements of individual users. Our multi-agent architecture satisfies the demands by separating the system into five types of components: manager agent (MA), extractor agent (EA), interface agent (IA), toolbox and databases, as shown in Figure 1.

In PathwayFinder system, each user is represented by an IA that collects the user's usage patterns to customize the interactions. Through IA, a user is essentially integrated into

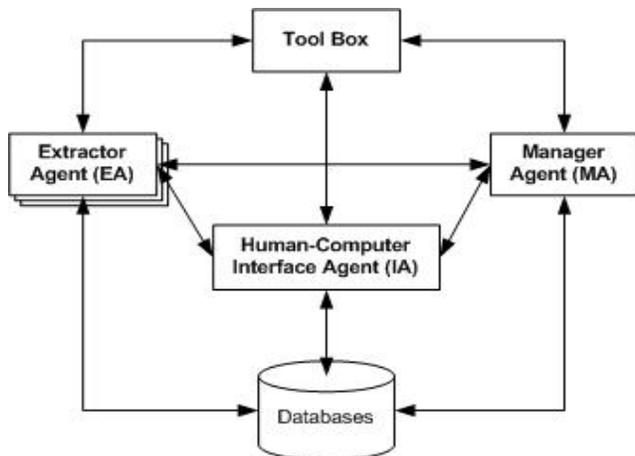


Figure 1: PathwayFinder system architecture

2.1 Manager Agent (MA)

MA provides two functions: one is to pre-process sentences, and the other is to organize EAs according to the sub-domain definitions.

Figure 2 shows the procedure of pre-processing, which is divided into six steps. Firstly, every selected sentence is tokenized into a word array. The result is inputted into the Brill's POS tagger (Brill, 1995), which determines the part-of-speech tag for each word. Next, the outputs from POS tagger are fed into the "Uniformization" step to generate the uniform word and uniform tag for each word. In the phrase combination step, PNPs are used to identify proteins in the sentence. Then, all the candidate patterns of the sentence are picked out by having each word looked up from the pattern database. These patterns will be given different rankings in different EAs according to their cooccurrence and users' feedback. In the last step, if the sentence contains one or zero protein or matches zero pattern, the pre-processor skips to the next sentence; otherwise, with the identified protein names, the sentence is directed to the proper EA according to sub-domain definition for further extraction. The MA also accepts requests from IAs, and directs them to proper EAs.

By properly dividing the domain into sub-domains, some hard-to-solve extraction errors can be avoided. Let us take

the extraction procedure with rich user-involved features. For example, a user can create interaction patterns through IA (see details in Section 3), and a confirmation of an extracted pathway from a user will be directed to the proper EA, which increases the ranking of the pattern used.

The extraction procedure is separated into two steps: pre-processing and extraction, which are handled by MA and EA respectively. The current PathwayFinder has only one MA, since pre-processing is similar in pathway extraction tasks. More MAs could be deployed for general extraction tasks. According to the sub-domain definition in MA, multiple EAs can be created to handle different targets.

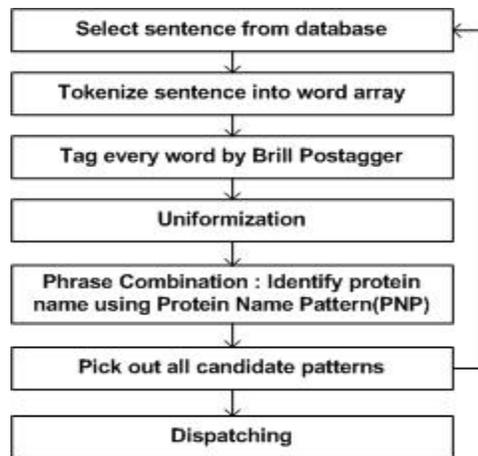


Figure 2: Pre-processing procedure

the following sentence by Ganoth *et al.* (2001) as an example.

"Human Cks1, but not other members of the family, reconstitutes ubiquitin ligation of p27 in a completely purified system, binds to Skp2 and greatly increases binding of T187-phosphorylated p27 to Skp2."

There are three interactions in this sentence: "Cks1 reconstitutes ubiquitin ligation of p27", "Cks1 binds to Skp2", and "binding of T187-phosphorylated p27 to Skp2". The part-of-speech tagging and the parsing results of the sentence are showed as follows (Please refer to http://monod.uwaterloo.ca/~dyao/download/PathwayFinder/pf_index.htm#down for the meaning of the tags.):

Part-of-speech tagging result: Human/NNP Cks1/NNP ./, but/CC not/RB other/JJ members/NNS of/IN the/DT family/NN ./, reconstitutes/NNS ubiquitin/VBP ligation/VBG of/IN p27/NNP in/IN a/DT completely/RB purified/VBN system/NN ./, binds/VBZ to/TO Skp2/NNP and/CC greatly/RB increases/VBZ binding/NN of/IN T187-phosphorylated/JJ p27/CD to/TO Skp2/NNP ./.

Parsing result: (S Human Protein (NP (NP , but not other members) (PP of (NP (NP the family) , (NP (NP (NP reconstitutes Protein ligation) (PP of (NP Protein))) (PP in (NP (NP a (ADJP (ADVP completely) purified) system) , (NP (NP binds) (PP to (NP (NP Protein) and greatly (NP (NP increases binding (PP of (NP Protein))) (PP to (NP Protein)))))))))))).)

The parsing result for the sentence is incorrect, even with all protein names identified. This kind of parsing error is common for parsers, and impossible to eliminate because of the complex sentence structures and the inherent ambiguity of nature languages.

Based on the given pre-processed results, non-agent version mistook the second interaction for “p27 binds to Skp2”, because p27 has a higher feature-ranking than Cks1, and pattern “PROTEIN bind to PROTEIN”, as a popular pattern, has a high score.

In our agent version, however, the second interaction is correctly extracted. The reason is that the pattern “PROTEIN bind to PROTEIN” has a low ranking in p27-agent since the pattern appears much less in it. And the pattern has a high ranking in Cks1-agent, which gives “Cks1 binds to Skp2” a higher score. The pattern “PROTEIN bind to PROTEIN” has a high ranking in phosphorylated-p27-agent, which extracts the third interaction correctly.

It is widely accepted in information extraction area that reducing the targeting domain will improve the extraction performance. It is one of the reasons that we apply multi-agent architecture to divide the targeting domain into smaller sub-domains.

The reason we put the pre-processing in MA is that the pre-processing for all sub-domains is similar in our system, while the rankings used in the extraction procedure for sub-domains are different. For significantly different sub-domains, “phrase combination” and “pattern picking” processes can be put in EAs for customized name recognition.

2.2 Extractor Agent (EA)

An EA conducts the extraction task for sentences in its own sub-domain, which is defined in MA, and maintains the related patterns and statistical data for the sub-domain. The implemented pattern-matching algorithm is similar to the Maximum Entropy Approach (Berger *et al.*, 1996). We have defined three types of features. The first is adjacent feature fa , which reflects the relations between uniformed tags. For example, tag pair “DT” and “NN”, which represent “determiner” and “noun” respectively, form an adjacent feature “DT-NN”, which reflects the probability of those two tags being in the same phrase. The second is grammar feature fg , which reflects the relations between a lower level tag and its parent tag in the grammar tree of a sentence. For example, “NP-NP” is a grammar feature which reflects the probability of the parent “NP”, which represents “noun phrases”, tracing down to a lower level “NP”. The third is pattern feature fp , which describes meaningful relations between elements in a sentence, such as “PROTEIN-activate-PROTEIN”. These features are pre-set, and can be adjusted according to their cooccurrence. Instead of the entropy, we calculate maximum feature scores of feature sets applicable to a sentence. The first calculated score is Adjacent-Pattern Feature score $A(p)$, which is the

maximum combination of all adjacent features and pattern features fit for the sentence. $A(p)$ makes use of the grammar information contained in the user-defined patterns, and measures the closeness between the sub-sentences and the patterns. It is introduced to compensate the deficiency of parsing caused by the complexity of biology literature. Another score is Grammar-Pattern Feature score $G(p)$, which is the maximum combination of all grammar features and pattern features fit for the sentence. It measures the matching between parsing results and the patterns. These scores are calculated as follows:

$$A(p) = \max\left(\prod_{f \in S_i} \max(fa, fp) \mid S_i \subset S\right)$$

$$G(p) = \max\left(\prod_{f \in S_j} \max(fg, fp) \mid S_j \subset S\right)$$

$$Score = k_1 \max(A(p), G(p)) + (1 - k_1)A(p)G(p)$$

where p is the pattern applied, f is the applicable feature, S is the set of all features in the sentence, S_i and S_j are the feature subsets, and k_1 is a constant, which is currently set to 0.75. For each “action” word, the qualified match with the highest combined score of $A(p)$ and $G(p)$ is selected as the extracted interaction.

In extraction procedure, the rankings of patterns and features are affected not only by users’ feedback, but also by cooccurrences of patterns and features in literature, which ensures the system work autonomously without users’ intervention.

2.3 Special Cases Processing

Besides the general grammar and pattern rules, there are other rules in the biological literature, which will affect the accuracy of extraction. Figure 3 shows some of the special cases handled by PathwayFinder.

Protein-Interaction interaction. In some cases, such as “..the protein HHR6B inhibits the interaction of cdc34 and ICP0..”, the protein HHR6B does not interact with another protein, but with an interaction. This kind of interaction is extracted by a multi-scan technique. Sentences containing at least one protein-protein interaction are scanned for the second time with extracted interactions substituted by new nodes labelled as “INTERACTION”. If this modified sentence matches any pathway patterns containing an “INTERACTION” tag, a nested interaction may be extracted. In this case, the “subject” or “object” of the extracted interaction is a link leading to the nested interaction. An example is given in Figure 3E.

Special proteins. Protein state and activity modification, such as mutants and protein inhibitors, are also identifiable. Special identifying processes are added during pre-processing. For example, “MEK inhibitor” is identified as “[x inhibit MEK]” instead of “MEK” and “COPI mutants” is identified as “COPI mutant” protein, instead of “COPI” protein.

Negative words. PathwayFinder identifies negative words within the scope of the interaction by searching through an editable negative word list, and removes those results. For example, in the following sentence, no interaction is extracted because of the negative words “neither” and “nor”.

“Neither rce1-null nor yor291w-null mutations affected PIO or the phenotype of spf1- or ste24-null mutants.”

Slash (“/”). Slashes may have different meanings in different sentences. It can be a simplified form of an interaction, as shown in Figure 3A, or an “and/or” relation, as shown in Figure 3B. These two cases are successfully

distinguished by patterns, since the former case has the subject and object in one word, which contains a “/”.

“And” and “Or”. “And” and “Or” may indicate an interaction or a parallel relation between proteins. To distinguish them, we create a relation set for the proteins connected by each “and” or “or” in a sentence. If both the subject and the object of the extracted interaction are in the same relation set, then the relation set is ignored. Otherwise, the “and” or “or” indicates a parallel relation, and expands the related subject or object. Examples are given in Figure 3C and Figure 3D.

A) slash in an interaction

Sentence ID: 75

Sentence: “...it is possible that the RPN-11/F55A11.3 interaction is involved in ...”

Action word: “interaction”

Associated pattern(s): “ROTEIN/PROTEIN interaction”, “interaction between/of PROTEIN and/with PROTEIN”

Applied pattern: “ROTEIN/PROTEIN interaction”

Extracted Interaction: “RPN-11 -> interaction -> F55A11.3”

B) slash as “and”

Sentence ID: 127

Sentence: “To examine whether direct protein-protein interactions between CCTs and the COP/DET/FUS proteins are ...”

Action word: “interaction”

Associated pattern(s): “ROTEIN/PROTEIN interaction”, “interaction between/of PROTEIN and/with PROTEIN”

Applied pattern: “interaction between PROTEIN and PROTEIN”

Extracted Interaction: “CCT -> interaction -> COP”, “CCT -> interaction -> DET”, “CCT -> interaction -> FUS”

C) “and” in parallel structure

Sentence ID: 1623

Sentence: “The fUBR11-1367 and UBR11-1140f, which, respectively, contained and lacked the RAD6-binding site (Fig. 2A), bound to GST-CUP9 with similar affinities (Fig. 5B, lanes 4-6 vs. lanes 1-3).”

Action word: “bind”

Associated pattern(s): “PROTEIN bind to/with/on PROTEIN”, “PROTEIN bind PROTEIN”

Applied pattern: “PROTEIN bind to PROTEIN”

Extracted Interaction: “fUBR11-1357 -> bind -> GST-CUP9”, “UBR11-1140f -> bind -> GST-CUP9”

D) “and” in an interaction, and “or” in a parallel structure

Sentence ID: 258

Sentence: “To test for the potential direct interactions between COP10 and the COP9 signalosome or COP1, a yeast two-hybrid assay was performed.”

Action word: “interaction”

Associated pattern(s): “ROTEIN/PROTEIN interaction”, “interaction between/of PROTEIN and/with PROTEIN”

Applied pattern: “interaction between PROTEIN and PROTEIN”

Extracted Interaction: “COP10 -> interaction -> COP9”, “COP10 -> interaction -> COP1”

E) “PROTEIN-INTERACTION” interaction

Sentence ID: 2184

Sentence: “Covalent attachment of SUMO-1 to Mdm2 requires the activation of a heterodimeric Aos1-Uba2 enzyme (ubiquitin-activating enzyme (E1)) followed by ...”

Action word: “attachment”, “require”

Associated pattern(s): “attachment of ROTTEIN to PROTEIN”, “INTERACTION require PROTEIN”

Applied pattern: “attachment of ROTTEIN to PROTEIN”, “INTERACTION require PROTEIN”

Extracted Interaction: “SUMO-1 -> attachment -> Mdm2”, “{SUMO-1 -> attachment -> Mdm2} -> require -> E1

Figure 3: Extraction examples. In each example, the patterns loaded for extraction are called associated patterns. Among them, the patterns used to extract the interactions are called applied patterns.

2.4 Human-Computer Interface Agent (IA)

IA provides rich user-involved features to absorb users' domain knowledge into the extraction process, and presents the required results to users. As an important function of PathwayFinder system, it is described in detail in Section 3.

2.5 Toolbox

Some utilities are relatively independent of the extraction process, thus they are wrapped into separate components and put into a toolbox. Currently, there are three tools integrated in the PathwayFinder system:

Paper Crawler grabs abstracts and papers from PubMed or other sources and stores them in the paper database.

Language Processing Server (PFSCTools Server) provides tagging and parsing services. It integrates Brill's POS tagger (Brill, 1995) and Link Grammar parser (Sleator and Temperley, 1991) into one program. This server supports TCP/IP connections similar to a web server. However, the PFSCTools server utilizes the RPC (Remote Procedure Call) service, instead of the GET or POST service of HTTP. The main application can connect to the PFSCTools Server through a network for POS tagging and parsing services.

Benchmark Comparer compares the extracted results with the standard results, both of which are in XML format.

The components in the toolbox are not agents, since they do not have learning capability. To speed up the processing, we can have several instances of each tool. For example, multiple Language Processing Servers can run on different computers to deal with the requests from different agents. Furthermore, these tools are independent, and can be replaced later with substitutes for better performance.

2.6 Databases

PathwayFinder uses several databases. Collectively, they store papers in text format, protein names, patterns, extracted pathways, and agent-specific knowledge that includes special cases of words, proteins, and the collected usage data.

3 IA: User-Involved Design

An IA acts as a mediator between a user and EAs. It provides two types of functions: customization, which customizes the MA and EAs to absorb users' domain knowledge and improves the extraction performance; and interactive presentation, which provides user-friendly interfaces for users to access the extracted pathway information. IAs also isolate the adverse effect of some erroneous information brought in by a particular user from others through the credibility ranking, which is generated from other users' feedback and statistical data. Thus, IAs can facilitate the tailoring of the extraction procedure to users' customized purposes.

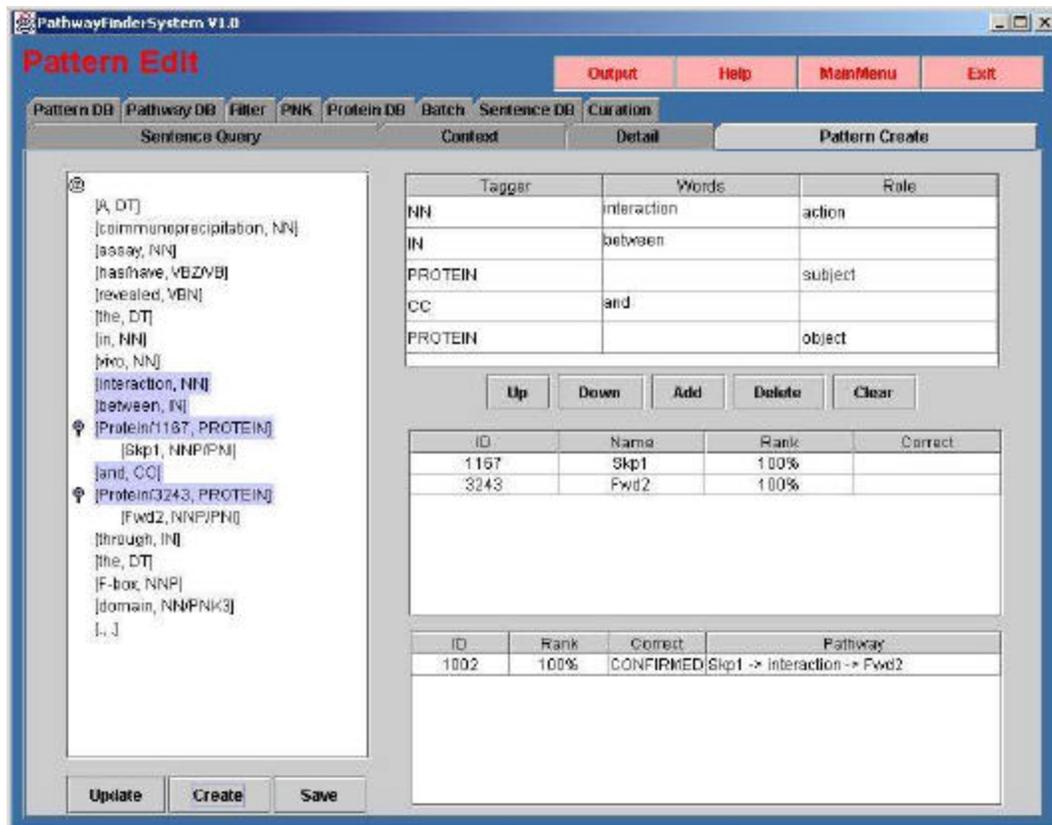


Figure 4: Demonstration of sample pattern creation

3.1 Customization

The variation of users' requirements is not limited to the queries they initiate, but includes the domain they define, the patterns they select, and the results they are interested in. For example, when a biologist queries on "CCT" in PathwayFinder, he may not be looking for a company named as CCT, or the definition of protein CCT, but the interactions between protein CCT and other proteins. The customization process not only helps users to tell the system what they want, but also integrates their domain knowledge into the system to improve the extraction capability. Following are three types of customizations supported by PathwayFinder system.

Pre-processing Customization In the PathwayFinder system, we introduce four new tags in the tagging process: "PNI" (Protein Name Identification), "PNK" (Protein Name Keyword), "PROTEIN" and "INTERACTION", to indicate simple or complex protein names. If a word has a "PNI" tag, it means that it is a candidate protein name; if a word has a "PNK" tag, then it is a piece of evidence that its neighboring words are protein names. After the phrase combination process, the recognized protein names are labeled as "PROTEIN", and some complex structures equivalent to proteins, such as "MEK inhibitor", are labeled as "INTERACTION". Users can browse through the "PNI" list to identify the true or false protein names, and edit the "PNK" list to indicate the new or special cases to improve the protein identification process. This knowledge is under constant development, and is not pre-fixed.

Pattern Customization A pattern contains all the key elements that can identify a protein interaction. In most other pattern-based information extraction systems, patterns are fixed, and can only be added or modified by the system developer. This is suitable for some special study cases, but can hardly adapt to users' frequently changing requirements.

The PathwayFinder system provides an open and flexible mode to create patterns combining users' domain knowledge. Users do not need to know a lot about NLP. They just indicate which words represent a protein-protein interaction in one sentence, and pick them out by a simple point-and-click method. As shown in Figure 4, key elements of a pattern are selected from the sentence by the user in the left window; the selected elements are transformed into a sample pattern in the upper-right window; and the pathways extracted from the sentence are listed in the lower-right window.

The pattern created by a user is called a "sample pattern", and represents an exact protein-protein interaction. A sample pattern is not directly used in the subsequent pattern matching, since many sample patterns have the same grammar constituents and there is considerable redundancy. We use general patterns to eliminate this redundancy. One general pattern is the summation of

many sample patterns that have the same grammar constituents. When a new sample pattern is added into the pattern database, it will merge into an old general pattern; or a new general pattern is created if there is not any suitable general pattern for merging the sample pattern.

As the issue of constructing a comprehensive pattern database is still one of the main problems in any pattern-based information extraction systems, we provide a gradual growth method to construct the pattern database in the PathwayFinder system.

Result Customization—Curation All the extracted results are initially labeled as "Auto". When users query or browse through the extracted interactions, they can label the results as "Confirmed" or "Denied" based on their domain knowledge. This piece of domain knowledge is also delivered to the related EA, and increases the ranking of the pattern that is used to extract the pathway. With one user's curation, all other users have the benefit of the confirmed results, which gives a high level of accuracy.

Another aspect of curation is knowledge accumulation. Users can merge their own knowledge into the pathway database. There are two ways to do it in PathwayFinder: for a small number of pathways, they can be imported one by one from the "Curation" Panel; and for large number of pathways or proteins, they can be put into text files, and loaded directly into PathwayFinder. The loaded proteins and pathways are given a ranking of 100%, which is reserved as user-confirmed pathways.

In the PathwayFinder system, the extraction process is separated from the query interface, and performed in the backend. Since users access the extracted results directly, not only can queries be processed much faster, but also further processing of the results, such as curation, is more easily accomplished.

3.2 Interactive Presentation

In PathwayFinder, results are presented to users in multiple ways to satisfy different levels of usage.

Pathway Query PathwayFinder provides two types of query functions:

1. Query on sentences, either on keyword or on sequence number. Since the extracted pathways are listed with the sentences, it is quite convenient to check whether there are any missing pathways;
2. Query on results. By inputting protein names, either one or multiple, users can get the related pathways from a diagram and browse to other interesting proteins by clicking on the protein names.

Two-Dimensional Diagram We create the query result diagram with the Minimum Distance Algorithm. Figure 5 is the query result diagram for E2.12, Daxx and Ubc9. The

5 Summary

The main contribution of the PathwayFinder is the significantly improved usability and extendibility provided by the user-involved extraction. We emphasize users' involvement in every aspect of our system. During pre-processing, users can specify the PNIs and PNKs to identify special forms of proteins; during the extraction, users can create, edit or delete patterns according to their domain knowledge. They can also confirm or deny the extracted results, which will not only increase the accuracy, but also affect pattern ranking in future extractions. Users can also track down the relationships between proteins on the diagram, which clearly states the credibility of the results, and jump directly to the related text source for further reference.

To support the user-involved extraction, we introduce an innovative multi-agent architecture. It provides not only the flexibility to satisfy users' constantly changing requirements, but also the flexibility of the system itself, which makes it much more extendible.

Compared with fully automatic extraction systems with fixed targets, PathwayFinder can adapt to any targets specified by users, and the shifting of targets is handled automatically by generating new agents.

User-involved extraction also lowers the requirements for the language processing unit. Since the user-defined patterns often contain the substructure of the relevant sub-sentences, we introduce the adjacent pattern feature to include this information into the language analysis result and compensate the deficiency of parsing. Although this method takes more storage and computation time, it is still feasible because of the cheaper and faster hardware.

User-introduced errors and the propagation of them are among the major problems of user-involved extraction. To handle the adverse effect of user-introduced errors, the statistical data is applied to adjust the pattern ranking, which will diminish the influence of improper patterns effectively. Besides, the design of IAs helps to isolate the erroneous information provided by particular users from others. Those problems will be further investigated as a future research topic.

The feedback of the system from domain users is positive. With rich user-involvement features, they can create their own patterns according to their domain knowledge without assistance from computer experts. The initially extracted results can act as an index to access related contexts. And with proper curation, the results can be used further to find functional pathways, or other potential relations between proteins.

6 References

Berger A.L., Pietra S., and Pietra V. (1996): A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Blaschke C., Andrade M., Ouzounis C., and Valencia A. (1999): Automatic extraction of biological

information from scientific text: Protein-protein interactions. In *Proceedings of the AAAI Conference on Intelligent Systems in Molecular Biology*, pages 60–67.

Brill E. (1995): Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Chaussabel D., Sher A. (2002): Mining microarray expression data by literature profiling. *Genome Biol* 3(10):Reserch0055.

Collier N., No C., and Tsujii J. (2000): Extracting the names of genes and gene products with a hidden markov model. In *Proc. COLING 2000*, pages 201–207.

Donaldson I., Martin J., De Bruijn B., Wolting C., Lay V., Tuekam B., Zhang S., Baskin B., Bader G.D., Michalickova K., Pawson T., Hogue C.W. (2003): PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics* 4(1):11.

Friedman C., Kra P., Yu H., Krauthammer M., and Rzhetsky A. (2001) Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 suppl. 1:74–82.

Fukuda K., Tsunoda T., Tamura A., and Takagi T. (1998): Toward information extraction: Identifying protein names from biological papers. In *Proceedings of Pacific Symposium on Biocomputing 3*, pages 705–716.

Ganoth D, Bornstein G, Ko TK, Larsen B, Tyers M, Pagano M, Hershko A. (2001): The cell-cycle regulatory protein Cks1 is required for SCF(Skp2)-mediated ubiquitinylation of p27. *Nat Cell Biol*. 3(3):321-324

Hatzivassiloglou V., Duboue P. A., and Rzhetsky A. (2001): Disambiguating proteins, genes, and rna in text: A machine learning approach. *Bioinformatics*, 17 no. 1:1–10.

Hatzivassiloglou V., Weng W. (2002): Learning anchor verbs for biological interaction patterns from published text articles. *Int J Med Inf* 67(1-3):19-32.

Hobbs J.R. (2002): Information extraction from biomedical text. *J Biomed Inform* 35(4):260 -264.

Humphreys K., Demetrios G., and Gaizauskas R. (2000): Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Proceedings of Pacific Symposium on Biocomputing*, pages 505–516, Hawaii.

Iliopoulos I., Enright A. J., and Ouzounis C. A. (2001): TEXTQUEST: Document clustering of MEDLINE abstracts for concept discovery in molecular biology,

- In Proceedings of Pacific Symposium on Biocomputing, pages 384-395, Hawaii.
- Krauthammer M., Kra P., Iossifov I., Gomez S.M., Hripcsak G., Hatzivassiloglou V., Friedman C., and Rzhetsky A. (2002): Of truth and pathways: chasing bits of information through myriads of articles. *Bioinformatics*, 18 suppl.1:249-257.
- Leroy G. and Chen H. (2002): Filling preposition-based templates to capture information from medical abstracts. In Proceedings of Pacific Symposium on Biocomputing 7, pages 350-361.
- Libbus B. and Rindflesch T.C. (2002): NLP-based information extraction for managing the molecular biology literature. *Proc AMIA Symp* 445-449.
- Marcotte E.M., Xenarios I., and Eisenberg D. (2001): Mining literature for protein-protein interactions. *Bioinformatics*, 17:359-363.
- Ng S. and Wong M. (1999): Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Informatics*, 10:104-112.
- Ono T., Hishigaki H., Tanigami A., and Takagi T. (2001): Automatic extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17 no.2:155-161.
- Oyama T., Kitano K., Satou K., and Ito T. (2002): Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18 no.5:705-714.
- Park J.C., Kim H.S., and Kim J.J. (2001): Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. In Proceedings of Pacific Symposium on Biocomputing, pages 396-407, Hawaii.
- Proux D, Rechenmann F, and Julliard L. (2000): A Pragmatic Information Extraction Strategy for Gathering Data on Genetic Interactions. In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, pages 279-285.
- Raychaudhuri S., Chang J.T., Sutphin P.D., Altman R.B. (2002): Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Res* 12(1):203-214.
- Rindflesch T.C., Tanabe L., Weinstein J.N., and Hunter L. (2000): Edgar: Extraction of drugs, genes and relations from the biomedical literature. In Proceedings of the Pacific Symposium on Biocomputing, pages 517-528, Hawaii.
- Sanchez C., Lachaize C., Janody F., Bellon B., Roder L., Euzenat J., Rechenmann F., and Jacq B. (1999): Grasping at molecular interactions and genetic networks in *drosophila melanogaster* using flynets, an internet database. *Nucleic Acids Res*, 27 no.1:89-94.
- Sleator D. and Temperley D. (1991): Parsing English with a Link Grammar, Carnegie Mellon University technical report, CMU-CS-91-196.
- Tanabe L. and Wilbur W. J. (2002): Tagging gene and protein names in biomedical text. *Bioinformatics*, 18 no.8:1124-1132.
- 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, pages 541-551, Hawaii.
- Wong L. (2001): PIES, a Protein Interaction Extraction System. In Proceedings of Pacific Symposium on Biocomputing, pages 520-531, Hawaii.