

# Extracting Protein Function Information from MEDLINE Using a Full-Sentence Parser

Nikolai Daraselia    Sergei Egorov    Andrey Yazhuk  
Svetlana Novichkova    Anton Yuryev    Ilya Mazo

Ariadne Genomics, Inc, 9700 Great Seneca Hwy, Rockville, MD 20850.  
{nikolai,esl,yazhuk,svetlana,ayuryev,mazo}@ariadnegenomics.com

## ABSTRACT

The living cell is a complex machine that depends on the proper functioning of its numerous parts, including proteins. Understanding protein functions and how they modify and regulate each other is the next great challenge for life science researchers. The collective knowledge about protein functions and pathways is scattered throughout numerous publications in scientific journals. Bringing the relevant information together creates a bottleneck in the research and discovery process. The volume of such information grows exponentially which, in turn, renders manual curation impractical. As a viable alternative, automated literature processing tools could be employed to extract and organize biological data into a knowledge base, making it amenable to computational analysis and data mining. We present MedScan, a completely automated NLP-based information extraction system. We have used MedScan to extract about 280,000 mammalian proteins functional links from the entire 2003 release of MEDLINE in only 21 hours. The precision of the extracted information was found to be 91%. We have compared the extracted data with protein co-occurrence data and with the nine well-studied cellular signaling pathways and estimated the recovery rate of MedScan for the entirety of MEDLINE to be between 30% and 50%. Further improvement of the MedScan technology is discussed.

## 1. INTRODUCTION

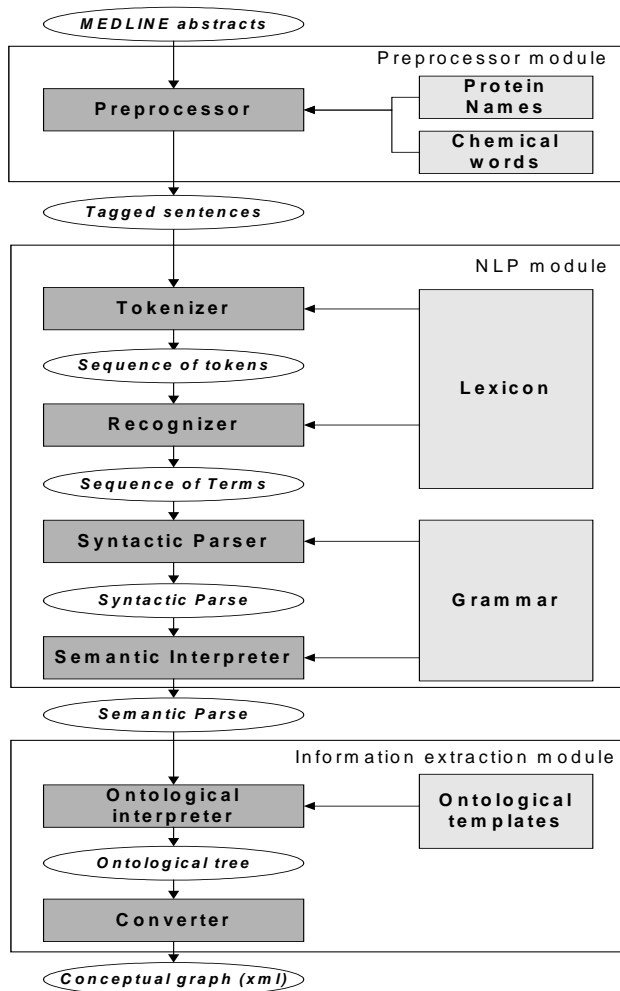
Information about protein function and cellular pathways is central to the system-level understanding of a living organism. Should such data be made easily available the problems that could be tackled include the following: quantitative simulation of complex cellular processes, identification of key elements in signal transduction, cross-talk between different pathways, mechanisms of gene co-regulation, and many others. There is a number of available databases covering different aspects of protein function, such as protein-protein interaction

DIP (<http://dip.doe-mbi.ucla.edu/>) and BIND (<http://www.bind.ca/>), regulatory gene networks (GeNet ([http://www.csa.ru/Inst/gorb\\_dep/inbios/genet/genet.htm](http://www.csa.ru/Inst/gorb_dep/inbios/genet/genet.htm))) and TRANSPATH (<http://193.175.244.148/>), or signaling pathways, CSNDB (<http://geo.nihs.go.jp/csndb/>) and SPAD (<http://www.grt.kyushu-u.ac.jp/eny-doc/>). However, since they are dependent on human experts, they rarely store more than a few thousand of the best-known protein relationships and do not contain the most recently discovered facts and experimental details. There is an urgent need for an automatic system capable of accurate extracting protein function information from literature. Only a few systems aimed at solving this task have been recently proposed. They range in approaches from simple statistical methods to advanced natural language processing (NLP) techniques.

The simplest way to extract protein relations from literature is to detect the co-occurrence of protein names in a text [12]. However, by its nature, the name co-occurrence detection yields very little or no information about the type of a described relation and, therefore, the co-occurrence data may be misleading. More sophisticated information extraction approaches rely on the matching of pre-specified templates (patterns) or rules, such as precedence/following rules of specific words. The underlying assumption is that sentences conforming exactly to a pattern or a rule express the predefined relationship(s) between the sentence entities. In some cases, these rules and patterns are augmented with additional restrictions based on syntactic categories and word forms in order to achieve better matching precision. The pattern-based systems have been applied to extract protein-protein interaction [1,8,11] and pathway information [10].

More advanced systems utilizing shallow parsing techniques have been described to extract protein interactions [13], enzyme reactions, and protein structure information [5], or functional relations between proteins [9]. Unlike word-based pattern matchers, shallow parsers perform partial decomposition of a sentence structure. They identify certain phrasal components and extract local dependencies between them without reconstructing the structure of an entire sentence. The precision and recall rates reported for shallow parsing approaches are 50-80% and 30-70%, respectively. Interestingly, most of the described systems are designed to extract only one specific aspect of protein function information.

The most promising candidates for a practical information extraction system are ones based on full-sentence parsing as they deal with the structure of an entire sentence and therefore are potentially more accurate. An example of such a



**Figure 1. The components and processing steps of the MedScan system.**

system is GENIES [4], which utilizes a parser and a semantic grammar consisting of a large set of nested semantic patterns (incorporating some syntactic knowledge), reflecting most frequently used sentence structures. Unlike other systems, GENIES is capable of extracting a wide variety of different relations between biological molecules, as well as nested chains of relations. However, the downside of the semantic grammar-based systems like GENIES is that they may require complete redesign of the grammar in order to be tuned to a different domain.

We believe that a key to efficient information extraction is in a modular architecture that separates natural language processing and information extraction into different modules. The NLP module deals with the domain-independent sentence structure decomposition, while the information extraction module can be reconfigured towards different tasks. We have previously reported a context-free biomedical domain-oriented NLP engine that parses sentences from MEDLINE abstracts into a set of alternative semantic trees [7]. In this paper, we present MedScan: a complete information extraction system that interprets these semantic structures using a pathway-oriented ontology and extracts protein function information.

## 2. SYSTEM OVERVIEW

MedScan is a three-tier information extraction system based on a full sentence parsing approach. Conceptually, it contains three modules: i) a preprocessor aimed to identify and tag various biomedical domain-specific concepts; ii) an NLP engine constructing the set of alternative semantic sentence structures; and iii) an information extraction module acting as a domain-specific filter for these structures and extracting information in a form of conceptual graph. An overview of MedScan architecture is presented in Figure 1.

### 2.1 Identification of proteins and other domain-specific concepts

Our approach for protein identification utilizes a semi-automatically curated protein name dictionary, which was based on and compiled from the LocusLink database and additionally enriched by incorporating protein names, aliases, descriptions and gene names from the linked GenBank, GoldenPath, and HUGO database entries. The resulting collection of protein “descriptors” contained along with correct protein names, functional keywords (e.g., “kinase”) clone names, as well as some completely irrelevant contaminant words and phrases. To improve the quality of this collection, the occurrence of each of the potential protein name in the 2003 MEDLINE release was determined by the method described below, and erroneous names were manually removed from the top 20,000 entries sorted by occurrence.

The rest of the entries were automatically processed in order to:

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

The resulting protein name dictionary consists of 245,248 records describing 81,915 unique proteins each assigned a LocusLink identifier.

In order to ignore variations in the protein name spelling we 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 preprocessor considers each punctuation character as belonging to a separate class. All letters belong to the alphabetical class, and all digits to the 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 (e.g., ‘alpha,’ ‘beta,’ ‘gamma,’ etc.). If such prefixes or suffixes are identified, they are stripped off and treated as separate tokens. The described tokenization procedure is followed by simple and efficient subsequence search applied to token sequences.

Protein names are identified by a variation of a string search algorithm. To implement the original idea of “relaxed” protein name matching, MEDLINE abstracts and dictionary entries are processed by the same tokenizer, and tokens belonging to the small list of “excluded” words are ignored. Next, a sentence is scanned for the presence of uninterrupted token sequences, consisting only of tokens derived from the dictionary processing. 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 mark (,;:?). 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 character length > 1 and "a"
- Comma is allowed between two alphabetical tokens/numbers the 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. If the subsequence lookup in the dictionary results in positive identification, the subsequence is marked up with a corresponding ID and the rest of the tokens to its right are searched for more matches.

An approach towards identification of chemical names is somewhat different: we have chosen not use formal mapping of a chemical name to an ID in any formal nomenclature due to a large size of the chemical dictionaries and simply mark up a name of a chemical substance in a text. Therefore, we employ a simplified matching algorithm that disregards rules of chemical nomenclature and instead utilizes the list of chemical "root words" that was created using the 2001 edition of United Medical Language System (UMLS) Metathesaurus. A total of 309,160 UMLS concept strings with semantic types "Organic chemical" and "Pharmacologic substance" were tokenized by our preprocessor module. Numbers and punctuation marks were removed, and the resulting list of approximately 77,000 non-redundant alphabetical tokens was used for chemical name tagging. The algorithm tokenizes the input sentence using the same procedure as for protein name recognition and the resulting sequence of tokens is searched for the longest subsequence satisfying the following criteria:

- It starts with a numerical token, a Greek letter spelled out in English, or an alphabetical token belonging to the list of chemical name constituents (the sum of "adjective" and "noun" morphemes).
- It contains numerical tokens, Greek letters, alphabetical tokens belonging to the list of chemical name constituents, and special punctuation symbols: comma, single quote, plus, minus, and round parentheses.
- If it contains parentheses, they should be balanced.
- A comma should be surrounded by two numerical tokens.
- A single quote should follow a numerical token.

Identification of other domain-specific concepts (cellular objects, complexes, and cellular processes) is based on the separate dictionary of such entities and is done in the manner identical to the protein identification.

## 2.2 NLP module

The NLP component of MedScan is a biomedical domain-oriented NLP engine that processes sentences from MEDLINE abstracts and produces a set of semantic structures representing the meaning of each sentence [7]. It is based on a context-free grammar and a lexicon developed specifically for MEDLINE. Processing is done in two steps. First, a syntactic

parser constructs a set of alternative syntactic structures of an input sentence. Because syntactic knowledge is ambiguous in its nature, a single sentence usually yields many (sometimes up to more than 100,000) alternative parses. Next, the semantic processor transforms each of them into a corresponding semantic tree. This conversion is based entirely on syntactic knowledge--namely, information about the predicate structure (number and order of arguments), and information about "prepositional patterns" associated with some of the lexemes (mainly verbs and verb nominalizations). The constructed semantic trees consist of nodes each representing a single lexeme. The nodes are connected by labeled vertexes, which can be split into two major categories: *thematic role* vertexes and *attributive* vertexes. Thematic role vertexes reflect essential relations, are constructed from verb and noun compliments, and are labeled according to the generally accepted order of lexeme compliments ("slot0," "slot1," and "slot2"). Attributive vertexes correspond to subtler qualifying or aggregative relations in a sentence usually expressed by prepositional phrases, coordinating and subordinating clauses, and conjunctive phrases, and are labeled simply as attributes ("attr"). Besides the name of a lexeme, each node is also labeled with an ontological labeled representing the meaning (or sense) of a lexeme. These senses, along with the vertex (slot) labels, are critical at the subsequent step of ontological transformation. The branches rooting from lexemes with no assigned senses are truncated.

## 2.3 Information extraction module

Protein function information usually constitutes only a part (and sometimes a rather small part) of a sentence. It needs to be intelligently extracted, considering the context of the entire sentence. Using only a part of the context may lead to errors in extracted information; this problem most often shows up in sentences with negation, uncertain modality, or improper type of main sentence statement ("hypothesize," "investigate," "analyze," etc.). MedScan always considers complete sentences. Our experience shows that the resulting increase in recovery rate makes up for the drop in precision of extracted information.

In MedScan, information extraction is controlled by a set of explicit declarative rules that specify which parts of an input semantic tree should be taken into consideration and what information should be retrieved. Input semantic trees are formed from various kinds of *frames* containing named *slots* filled with values, which can be strings or other frames. The recursive nature of input data requires recursive processing; the MedScan information extraction mechanism follows the input tree structure in a top-down manner, applying a set of context-free and context-dependent *transformation rules*. Each rule is essentially a conversion specification giving an example of input frame with slots filled with specific values or containing "variables" (an input pattern), followed by an output tree with positions to be filled with information extracted into those variables (an output template). The rules define a tree transformation procedure, generating an output tree from an input tree. Rules are considered in the order that they were written. If a pattern part of a rule matches the input tree and the sub-trees extracted into variables can be successfully converted (via the same set of rules), the output template is filled with sub-tree conversion results, producing the output tree. If either the pattern match or a sub-tree conversion is unsuccessful, the next rule is considered; if all alternatives fail, the conversion fails and no output tree is produced.

MedScan rules are not necessarily mutually exclusive; in many cases, more than one rule can match a particular input tree (or sub-tree). Rules employing a deep structural match are usually placed first so they can be tried before more generic “catch-all” rules. In addition to the deep structural match (an input pattern requiring particular values for slots of nested frames), MedScan rule variables can be restricted so they take only certain frames as values. This provides a certain degree of context-sensitivity to the transformation language defined by the rules. Although context sensitivity increases the complexity of the rules and is able to slow down the transformation process, it is necessary in order to handle the complicated constructs of natural languages.

In addition to covering the needs of basic transformation, special care is taken to reduce the number of rules written explicitly. Since many semantic structures defining the shape of input trees share common characteristics, their respective transformation rules have many common parts. To facilitate code sharing, MedScan rule language employs *slot list rules* resembling the subroutines of procedural programming languages. Slot lists are not nodes of input trees; they are free collections of slots with the corresponding values that can be a part of a given frame node’s contents. Allowing such collections to be transformed by separate rules significantly reduces the number of rules needed to describe a complete transformation system for MedScan semantic trees. (The estimated rule size reduction is two orders of magnitude or more.)

There are two kinds of slot list rules, covering two kinds of slots in MedScan’s semantic frames: *required slot list rules* describe transformations of widely used slots such as “agent”,

whereas *optional slot rules* describe transformations of discretionary attribute-like slots. The difference is manifested in the rules’ behavior: while required slot rules fail and backtrack as regular frame rules, optional slot rules never fail; only the successful optional slot transformations are recorded in the output trees.

MedScan rules describe the transformation of singular frame trees. The actual input trees though contain numerous “forks” describing alternative and conjunctive lists resulting from the corresponding constructs in the natural language. Input trees represent such constructs in a “packed” form; the task of MedScan transformation engine is to keep these lists in packed form for as long as possible. For each packed node, the transformation “forks” into multiple sub-transformations, whose results are recombined, whenever possible, to produce packed output nodes. Alternative and conjunctive lists are handled differently. Possibilities for such “bulk” transformations are deduced from transformation rules at the initial rule processing stage by factoring out a context-free “sublanguage.” Later, the rule transformation engine keeps track of its input, reusing the results obtained on previous steps, instead of recalculating them for each alternative. In this way, the processing of many thousands of alternative parses can be done with low per-alternative cost. Effective handling of alternative parses is a crucial requirement for a practical NLP-based system—the number of alternative parses can easily reach 100,000. An overview of information extraction process is shown on Figure 2.

As the last step, the output trees coming out of the MedScan transformation engine are compared for uniqueness.

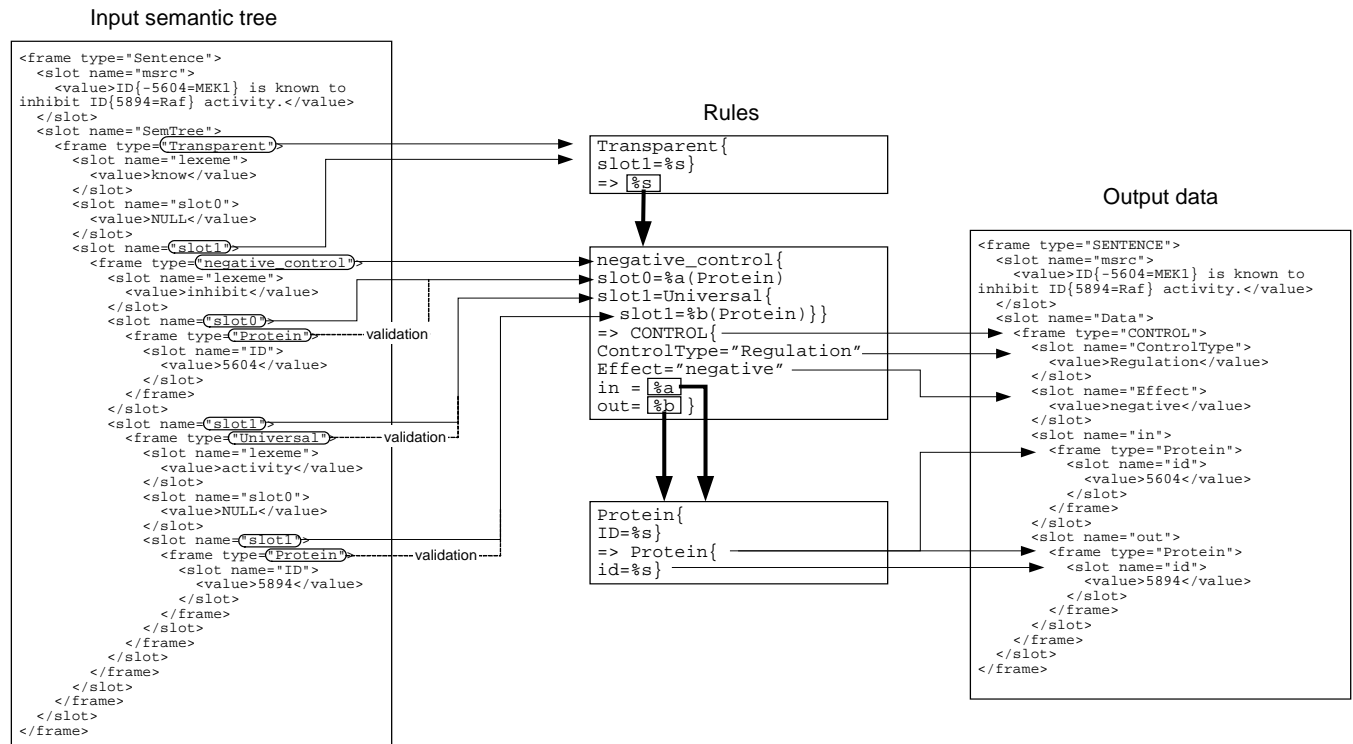


Figure 2. An overview of information extraction process.

This is necessary because the nature of the transformation process is many-to-one: alternatives that were different on input can produce equivalent outputs. Discarding duplicate results provides significant savings in space and time required for further processing.

The **ResNet converter**, the last module in the MedScan pipeline, converts the constructed output frame tree into the form of a generalized conceptual graph (ResNet), and transcribes it into XML format. The retrieved information in the final form is filtered for uniqueness, so that identical links are represented as a single link with multiple references to the MEDLINE sentences supporting it.

### 3. EXTRACTION OF PROTEIN FUNCTION INFORMATION FROM 2003 MEDLINE

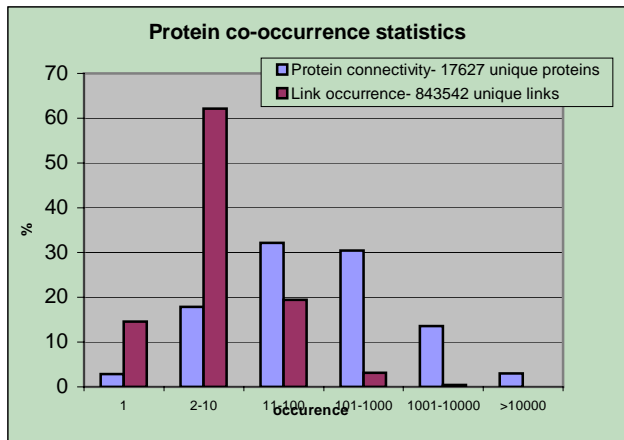
We have used MedScan to retrieve different types of regulatory links between proteins, cellular processes, and small molecules from the entire 2003 MEDLINE release. The information extracted included such phenomena as gene expression regulation, molecular transport, protein modification, regulation of cell processes, indirect causal relations between different entities, and other types of data conforming to a general notion of molecular networks and represented in the ontology developed specifically for protein and cellular function description.

The 2003 release of MEDLINE is about 50 GB and contains about 7.8 million abstracts, or roughly 81 million sentences. The scope of our experiments was limited to the mammalian protein function information extraction. Out of the entire MEDLINE database, 7.3 million were selected and tagged by the preprocessor and contained at least one notation of mammalian protein. Out of these sentences, 51% have been successfully parsed by the NLP engine of MedScan and produced 3.21 million “semantic forests,” which were further processed by the information extraction module. A total of approximately 280,000 million of unique functional relations were finally extracted. The entire processing took about 21 hours. The number of extracted relations of each type is presented in Table 1. The precision of the entire pipeline was determined by manual inspection of 988 randomly chosen extracted relations and was found to be 91%.

Next, we were interested in determining the coverage of MedScan, i.e., the proportion of all protein functional links described in MEDLINE, which are extracted by our system. It is customary for an information extraction system to determine a fact-by-fact recall rate, when the set of MEDLINE abstracts is

**Table 1. Types and abundance of links extracted from 2002 MEDLINE.**

Relation type	Count
Expression control	39,121
Binding	19,906
Prot. modification	6,053
Mol. Synthesis	12,915
Mol. Transport	21,642
Cell object control	530
Unknown regulation	177,703
Total	278,305



**Figure 3. The distribution of number of unique proteins co-occurring with each protein and each unique protein-protein co-occurrence frequency.**

presented to the human expert, who manually identifies and records all functional links described therein. Then, this set is compared to the set extracted by the automated system, and the proportion of the manually recorded links recovered by the system indicates the recall rate of the system. To eliminate the impact of the protein identification step, we have slightly modified this procedure: 6,000 randomly selected sentences, which have passed the preprocessing step, were presented to an expert familiar with our ontological model, who identified 1,560 different functional links between small molecules, proteins, cellular processes, and cellular components. The same sentences were then submitted to MedScan, and a total of 531 links were extracted. Of these links, 483 were found to be correct, which corresponds to the recall rate of 31%. However, because protein function information is redundant, such an approach hardly serves as a recovery measure of the system within the entirety of MEDLINE. Therefore, we did two more independent assessments of the system coverage: first, by comparing the extracted information with sentence-level protein co-occurrence statistics, and then by comparison of the extracted information with a set of nine well-studied “gold standard” signaling pathways.

The sentence level protein co-occurrences have been used as a statistically based method of extraction of functional relations between proteins without in-depth analysis of a sentence semantics [12]. The underlying assumption is that the more frequently protein names are cited in the same sentence, the more likely they are functionally related. Despite its simplicity, such an approach towards the detection of functional links between proteins has the highest recall among all information extraction methods, while the precision has been reported to be in the vicinity of 50% [3]. We have used the protein name co-occurrence statistics to evaluate the number of protein functional links in MEDLINE. The protein co-occurrence was obtained directly from the results of protein name identification step by generation of all possible links between all proteins identified within each sentence. A total of 17,627 unique proteins were identified in all 7.3 million sentences selected by the preprocessor; they have yielded 843,542 unique co-occurrence links. The distribution of the number of unique proteins associated with each of the 17,627 proteins is presented and citation frequency of all co-occurrence links is presented on Figure 3. According to the estimated 50% precision rate of the co-occurrence data, there is about 421,777

**Table 2. Comparison of the extracted protein function information with the nine well-studied “gold standard” pathways.**

Pathway	Links	Total extr. links	Pathway coverage	Co-occur. coverage	Redundant indirect links	Novel links	Erroneous links
JNK kinase pathway	85	318	43(50.6%)	74(87.1%)	151(47.8%)	12(3.8%)	21(6.6%)
Integrin pathway	101	352	54(53.5%)	80(79%)	171(48.6%)	17(4.8%)	25(7.1%)
Fas pathway	92	307	52(56.5%)	82(89.1%)	167(54.4%)	21(6.8%)	26(8.4%)
p38-MAPK pathway	64	180	31(48.3%)	54(84.4%)	101(56.1%)	11(10.1%)	15(5.6%)
Insulin pathway	65	212	34(52.3%)	51(78.5%)	122(57.5%)	6(2.8%)	21(9.9%)
Toll-like receptor	63	245	31(49.2%)	54(85.7%)	143(58.4%)	9(3.7%)	18(7.3%)
ERK pathway	54	201	28(51.8%)	51(94.4%)	97(48.3%)	8(4.0%)	19(9.5%)
TNF pathway	39	136	24(61.5%)	33(84.6%)	71(52.2%)	5(3.7%)	9(6.6%)
IL4 pathway	32	151	18(56.3%)	29(90.6%)	83(55.0%)	12(7.9%)	16(10.6%)

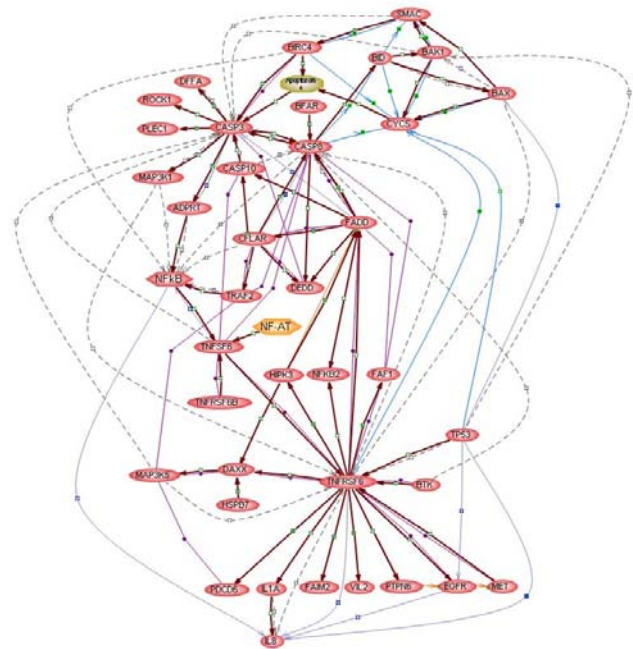
functional protein links described in MEDLINE. The 278,305 functional links extracted by MedScan correspond to 128,420 unique protein-protein pairs, and therefore the recall rate of the MedScan system in entire MEDLINE is 128,420/421,777=30.4%. The difference between number of unique protein-protein pairs and total number of extracted functional links is due to the redundancy of the extracted information, which is discussed below.

Next, we have compared extracted data with the set of “gold standard” well-known pathways. The pathway information was obtained from the signal transduction knowledge base available on the Web site of *Science* magazine ([www.sciencemag.org/](http://www.sciencemag.org/)). We have chosen the nine largest pathways (shown in Table 2). The analysis was aimed not only towards estimation of the MedScan recovery rate but also towards rough analysis of the nature of extracted data. Accordingly, we have evaluated the following criteria for each pathway:

- Total number of links between pathway entities (proteins, complexes, etc.) identified in MEDLINE 2003 by MedScan
- Number of links in reference pathways identified by MedScan
- Number of links in reference pathways identified at the sentence-level co-occurrence
- Number of indirect redundant links identified by MedScan. Redundant links are indirect links between any upstream and downstream elements of a pathway
- not connected by direct links in reference pathways (for pathway A->B->C, link A->C will be indirect and redundant).
- Number of novel non-redundant links between pathway entities not present in reference pathways
- Number of erroneously extracted links for each pathway

The results of this analysis are shown in Table 2. The average number of reference links for all pathways present in the extracted data is 52%, i.e., only about half of all pathway relations have been extracted from the release of the entire MEDLINE 2003. For comparison, we also present the data on protein co-occurrence coverage of each pathway– the proportion of all links in reference pathway where two linked pathway entities co-occur in at least one sentence in MEDLINE. As the results indicate, the average co-occurrence coverage for all nine pathways is about 87%. Interestingly, most of the recovered reference pathway links are represented by more than one link extracted by MedScan. This occurs because the identity of the extracted links is determined not only by the identity of the connected entities and direction of

the links, but also by the type of the link and, in some cases, by the sign of the link (positive vs. negative vs. unknown). (Refer to Table 1 for the types of extracted links.) For example, “phosphorylation” links in the reference pathway were usually mirrored not only by extracted “phosphorylation” but also by the protein-protein interaction relation, as well as “regulation” links. An average number of extracted links covering a reference link is 3.3. This observation suggests that the functional links extracted from MEDLINE are redundant and extracted information can be further compressed by merging (or removing) less specific links



**Figure 4. Analysis of the extracted data using Fas signaling pathway. The reference links are shown as solid bold dark-red links, while extracted links – as thinner links of different colors and line patterns (depending on a type of the links). Blue solid links denote molecular transport events, red solid links with small circles denote protein interaction events, grey solid links with small rectangles denote expression regulation events, and grey dashes links – general regulation events. The majority of redundant links are not shown to avoid picture crowding. If present, extracted links are shown next to the corresponding matched reference link.**

between each individual pair of connected proteins with more specific ones. Next, we discovered that, for each pathway, an average of 50% of extracted links not present in the reference pathway were indirect and redundant. Finally, only about 10% of extracted links were novel and non-redundant. An average 8.1% of all extracted links were erroneous, which agrees with the estimated precision of the system. Figure 4 illustrates the results of the analysis of extracted data using the Fas signaling pathway as a reference. Note that the majority of the indirect redundant links have been removed to simplify the picture and also all redundancy between multiple links connecting two individual entities have been manually eliminated leaving only the single most specific link.

Comparison of the extracted links with protein co-occurrence data and with reference pathways indicated that currently MedScan extracts about 30-50% of protein function information available in MEDLINE. We estimate that this recall rate is primarily due to the following two reasons: first, the coverage of MedScan grammar is about 51%, which means that information is extracted from only about half of all sentences. Second, our analysis revealed that about half of the protein function information in MEDLINE is expressed in a form of raw experimental data presentation and is not covered by our ontology. Interpretation of experimental data is not a trivial task--even for human experts--and extracting information from the description of experimental results extends beyond the natural language processing field into the domain of logical inference and logic programming. For example, the sentence, "Number of MDA-2 cells with apoptotic phenotype increased 50% after treatment with wortmannin" implies that the wortmannin somehow controls apoptosis, but this knowledge cannot be inferred from the sentence by NLP methods alone. Our observation is that functional information expressed as raw data is less reliable and requires caution in interpretation. However, it might still represent a considerable interest to the users of the technology. We therefore envision two major goals of further improvement of MedScan: the improvement of the NLP grammar and the enrichment of the ontological rules to include some of the information presented in a form of raw experimental data.

#### 4. ACKNOWLEDGMENTS

This work supported in part by NIH grants 1R43-GM067276 and 1RO1-GM068954

#### 5. REFERENCES

[1] Blaschke, C., Andrade, M.A., Ouzounis, C., and Valencia, A. (1999). Automatic extraction of biological information from scientific text: protein – protein interactions. *Ismb*: 60-67.

[2] Blaschke C., and Valencia, A. (2002) The frame-based module of the Suiseki information extraction system, *IEEE Intelligent Systems* 17: 14-20.

[3] Ding, J., Berleant, J., Nettleton, D., and Wurtele, E. (2002) Mining MEDLINE: Abstracts, Sentences, or Phrases? *Pac Symp. Biocomput.*

[4] 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: S74-S82.

[5] Humphreys, K., Demetriou, G., and Gaizauskas, R. (2000). Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. *Pac Symp. Biocomput.*: 505-516.

[6] Karp, P.D., Riley, M., Paley, S.M., Pelligrini-Toole, A., and Krummenacker, M. (1999). *Eco Cyc: Encyclopedia of Escherichia coli genes and metabolism*. *Nucleic Acid Res.* 27: 55-58.

[7] Novichkova, S., Egorov, S., and Daraselia, N. (2003) Medscan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, in press.

[8] Ono, T., Hishikagi, H., Tanigami, A, and Takagi, T. (2001). Automated extraction of information on protein – protein interactions from the biological literature. *Bioinformatics* 17: 155-161.

[9] Park, J.C., Kim, H.S., and Kim, J.J. (2001). Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. *Pac. Symp. Biocomput.* 6: 396-407.

[10] See-Kiong, N., and Wong, M. (1999). Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Informatics* 10: 104 – 112.

[11] Sekimizu, T., Park, H.S., and Tsujii, J. (1998). Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. *Genome informatics* 9: 62-71.

[12] Stephens, M., Palakal, S., Mukhopadhyay, S., and Raje, R. (2001). Detecting gene relations from MEDLINE abstracts. *Pac Symp Biocomput.*: 483 – 495.

[13] Thomas, J., Milward, D., Ouzounis, C.A., Pulman, S., and Carroll, M. (2000). Automatic extraction of protein interactions from scientific abstracts. *Pac. Symp. Biocomput.*: 541-552.

[14] Yakushiji, A., Tateisi, Y., Miyao, Y., and Tsujii, J. (2001). Event extraction from biomedical papers using a full parser. *Pac. Symp. Biocomput.* 6: 408-419.