CONAN: An Integrative System for Biomedical Literature Mining

Rainer Malik[1] and Arno Siebes[1]

Universiteit Utrecht, Institute for Information and Computing Sciences, PO Box 80.089, 3508TB Utrecht,The Netherlands,
rainer@cs.uu.nl

Abstract. The amount of information about the genome, transcriptome and proteome, forms a problem for the scientific community: how to find the right information in a reasonable amount of time. Most research aiming to solve this problem, however, concentrate on a certain organism or a very limited dataset. Complementary to those algorithms, we developed CONAN, a system which provides a full-scale approach, tailored to experimentalists, designed to combine several information extraction methods and connect the outcome of these methods to gather novel information. Its methods include tagging of gene/protein names, finding interaction and mutation data, tagging of biological concepts, linking to MeSH and Gene Ontology terms, which can all be found back by querying the system. We present a full-scale approach that will ultimately cover all of PubMed/MEDLINE. We show that this universality has no effect on quality: our system performs as well as existing systems.

1 Introduction

It is an often quoted fact that the number of articles in MEDLINE and PubMed is growing exponentially [1]. The problem for the scientist is that interesting and useful information, like interaction data and mutation data, could appear in papers they have not read. Therefore, important facts might get overlooked and the scientific work might be affected. To overcome these problems, many systems have been developed that search the literature automatically for the relevant information [2]. Most systems, however, focus only on a very specific aspect of literature, on a very limited dataset or on a certain organism.

Complementary to those systems, we want to address - as completely as possible - the problem of experimentalists to find certain information “hidden” in the abstracts of biomedical literature. We present here the first release of CONAN, a system which is as complete as possible, offering a wide range of information. This information is also combined to construct new information, e.g. the output of a protein name tagging method is used as input for a method which finds Protein-Protein-Interaction Data and as input to find protein synonyms. Our system can be regarded as the “right-hand” of a scientist: given a query, it hands the researcher back a set of essential results.

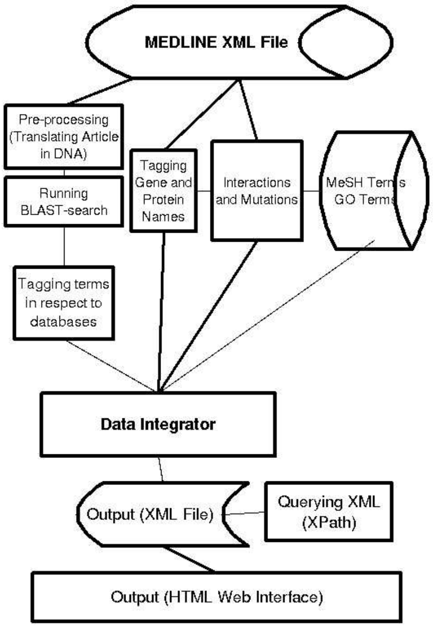
MEDLINE XML File
Pre-processing (Translating Article in DNA)
Running BLAST-search
Tagging terms in respect to databases
Tagging Gene and Protein Names
Interactions and Mutations
MeSH Terms GO Terms
Data Integrator
Output (XML File)
Querying XML (XPath)
Output (HTML Web Interface)

Fig. 1. Flow Diagram of CONAN

Our goal is not to find new algorithms, but to integrate interesting and important algorithms into one system. The system can be installed locally or accessed via a web-service.

The road map of this paper is as follows: In the next section, we describe the general architecture of CONAN and its components. In Section 3, we show the performance evaluation, and we discuss the results. In Section 4, we draw the conclusion and give future directions.

2 Approach

The general architecture is shown in Figure 1. It shows that MEDLINE XML Files, containing abstracts, serve as input for several processing steps, namely BLAST-searching and the tagging of Gene and Protein names. These Gene and Protein names serve as input for the detection of Protein-Protein-Interaction Data. Mutation Data is also extracted from the abstracts. MeSH- and Gene Ontology (GO)-terms serve as additional input, the data is combined and integrated in the Data Integrator-Step (see Section 2.4), before it gets stored in an XML-File. This XML-File can be queried directly by XPath-queries or via a Web-Interface, using pre-defined queries.

2.1 Input

MEDLINE The basis of all the tagging and information extraction are the MEDLINE files released by the National Library of Medicine. The database contains over 12 million citations dating back to the mid-1960's. PubMed is the system which provides access to bibliographic information that includes MEDLINE [3]. Via a licensing system, users are allowed to download MEDLINE files or are able to get the files wanted on tape sent to them by the NCBI. These files are in the MEDLINE XML format. For this experiment, only articles were taken into account which are completed, meaning that it is the final version of articles which have an abstract and were written in English.

MeSH The second source of information are the MeSH (Medical Subject Headings)-terms. MeSH (`http://www.nlm.nih.gov/mesh`) is the National Library of Medicine's controlled vocabulary thesaurus. It consists of sets of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity. MeSH terms are part of NCBI's MEDLINE distribution.

Gene Ontology The third source of information is the Gene Ontology (GO) Database [4] and, more specifically, the current annotation of Uniprot terms by GO (GOA) [5]. The concept of GO is to develop three structured ontologies, namely Biological Processes, Cellular Components and Molecular Functions. Protein names found in text have a certain synonym in UniProt [6]. These synonyms again are annotated by the EBI, assigning GO-terms to Uniprot-terms.

ENSEMBL As an additional source of information, for each gene/protein name found in text, the corresponding ENSEMBL-identifier is retrieved. ENSEMBL [7] provides complete and consistent annotation across the human genome as well as other genomes. ENSEMBL identifiers are used as cross-references to other identifiers and are therefore included in our system.

2.2 Output

We designed our system to give scientists a tool to help them find and process valuable information in abstracts. What we want the system to achieve is to combine several sources of input, thereby finding "novel" knowledge in biomedical literature and presenting this knowledge to the user. The "novel" information should be accessible as easily as the original information. The data available consists of information per abstract. The user can use several entry points to gather information. Specific PubMed ID numbers (PMIDs), Protein and Gene Names, Protein-Protein-Interactions, Gene Ontology (GO) identifiers, UniProt identifiers or EnSeMBL codes, but also specific biological concepts like Cell, Cell Component or Cell Type can be given to display all information extracted from a particular abstract. The user can query the database in two different ways, discussed in the next two sections. Dependent on which way the researcher used the system, several of those questions can be combined to one query.

XML-Output The basic output of CONAN is an XML file which holds all information about a certain abstract. For this XML file, a DTD (Document Type Definition) file was constructed which defines the legal building blocks of the XML document. This means that only specific types of data can be entered in the XML document. XML was chosen because it offers several benefits in regard to storing data, a major advantage being its platform-independency. The XML-file can be queried with several XML Query Languages (XQuery, XPath, XSLT). Using the query language, the researcher can combine several of the questions mentioned above.

HTML-Output A web-server has been set up which allows users to query our results. The web-server was built using HTML, the XML database has been made accessible for querying via a web-server by using the server-side scripting language Perl. The querying of the XML-file is done via XPath (`http://www.w3.org/TR/xpath`) which allows to refine the query while still being fast enough. The overall goal of this server is to generate reliable results of biological information in biomedical abstracts, given user-defined input terms. This web-server is an internet-based application.

There are two main query systems: Quick Search and Advanced Search.

- When starting a Quick-Search session, the user has the option to give two different types of data as input: the PMID of an article, or the name of a gene or protein. The logical operators (AND,OR) can be used to combine several terms to one search term. When giving a PMID, the result page shows all information found in the specific abstract and all “novel knowledge” found by combining the different data sources.
- In an Advanced Search, the input can be a Gene name, a Protein name, an arbitrary keyword,a PMID or a gene ontology number. When giving a protein name, there is a possibility to search for an interaction or mutation where this protein is involved.
- In neither of these search-methods, it is currently possible to combine several questions to one query.

In both Search Methods, the results itself are links to different pages again.

There is no graphical visualization of the results yet, but there are plans to make this happen in the future.

2.3 Algorithms

In this section, we show the algorithms used in this system. For details about these methods, please refer to the original publications and to Table 1.

Table 1. Overview of Methods and Algorithms used (derived from original publications)

Name	Measure
BLAST-Searching	Recall: 78.8%
	Precision: 71.7%
Gene/Protein-Tagging	Prediction correct: 77%
Mutation-Tagging	Sensitivity: 87.1%
	Specificity: 87.9%
Protein Tagging	Precision: 75%
	Recall: 76%

Table 2. Sample of Databases used in keyword search

Database	No. of Terms included
Gene	737801
Protein	41733
Organic Chemical	38258
Disease or Syndrome	36999
Therapeutic or Preventive Procedure	8328
Neoplastic Process	7791
Species	14121
Body Part, Organ or Organ Component	6555
Cell Component	818
Cell Function	456

BLAST-Searching The first method implemented is a BLAST-searching method first published by Krauthammer et al. [8]. It uses the BLAST-algorithm [9] to discover relevant biological information in text. In the original method, only gene and protein names were used.

In our approach, the original method was improved to extract even more information from an abstract. The UMLS Metathesaurus [10] is used by us to retrieve lists of biological relevant terms. These terms were split up into different databases, specified by their so-called Semantic Type. An example of this would be the terms "Cell Differentation" and "Endocytosis" which have the Semantic Type "Cell Function" assigned. The system includes now 90 different databases (Table 2 shows a selection of some databases used). The terms in the databases are translated to DNA using a specific translation table which can be found in the original publication. The abstract itself is translated following the same schema. To adjust the search process,the BLAST-parameters e-value and Word Size had to be adjusted (see Table 3). After fine-tuning these parameters, the BLAST-algorithm can be used to extract biological concepts from text in a quick manner.

Gene-Tagging The second method implemented is a Gene/Protein-tagging method called AbGene first published by Tanabe [11]. It uses a combination of

Table 3. Parameters used in BLAST-search

Term length (characters)	e-value	Word Size	mismatch penalty
3	$1e^{-15}$	12	6
4-5	$1e^{-15}$	16	6
6-10	$1e^{-20}$	20	3
11-20	$1e^{-25}$	40	3
20 or more	$1e^{-25}$	80	3

statistical and knowledge-based strategies. It does not make a distinction between Gene and Protein Names. This method incorporates automatically generated rules from a transformation-based part-of-speech tagger, and manually generated rules from morphological clues, low frequency trigrams, indicator terms, suffixes and part-of-speech information. The original source code was taken and altered slightly to suit the needs of CONAN.

Mutation-Tagging The third method implemented is based on the MuText application [12]. It takes several regular expressions to detect mutations mentioned in an abstract.The pattern usually starts with one amino-acid in one- or three-letter-code, followed by a number and another amino-acid abbreviation. The result of these regular expressions are verified or falsified by the surrounding elements in the text. While there is still a chance of false positives (e.g. the pattern of one-letter/number/one-letter is frequently found in names of cell lines), comparing it to data obtained by the other methods lowers the false positive ratio. As an example, if the BLAST-searching method finds the words "mutation" or "mutagenesis" in the abstract, there is a high probability that the mutation found is really a mutation.

Protein-Tagging Another method to find protein names and information about these proteins in the text comes from the so called NLProt method [13, 14]. It automatically extracts protein names from the literature and links those to associated entries in a sequence database. It uses rule-based filtering and Support Vector Machines (SVMs) to tag protein names in PubMed abstracts. It also gives the corresponding UniProt entries of the protein names as well as the organism this protein belongs to.

Interaction-Finding The last method implemented is used to extract interaction data from text. The basis of this method are again regular expressions as used in the PreBIND and BIND system [15, 16]. Some regular expressions have been deleted by us from the set due to redundant results. There are several categories of possible interactions:

- positive interactions
- negative interactions (inhibitions)

- positive/negative complex building / subunit / association
- positive binding
- negative binding
- activation
- (de-)phosphorylation
- (co)precipitation
- conjugation
- mutation

It is very important to notice that not only positive interactions are found by this method.

2.4 Data Integrator

Having different methods to obtain data, the main focus is to combine this data to gather new information. There are two major ways in CONAN to join data.

- The first way is to combine data before storing it in the Output-XML-file. This is done in the case of Interaction Data. The output of the NLProt-method serves as input for the Interaction-Tagging method. The list of protein names found by NLProt is passed through to the Regular Expressions which give Protein-Protein-Interactions as a result. Additionally, the output of the NLProt method is used to find the related UniProt, ENSEMBL and GO-identifiers.

  Another example of this type of integration of data is the validation of those protein names found by NLProt. NLProt offers a reliability-score (ranging from 0 to 1), assigning a score to each protein-name found in text. Given this score, a list is constructed which holds each occurrence of a protein name in text and its score. The list is compared to the list of protein names found by the BLAST-searching-method and the list of Gene- and Protein-names found by the Gene/Protein-Tagging-method. If certain terms are found by the other methods as well, the reliability score increases (+0.1 for each additional method) and it decreases, if the name is not found by the other methods (-0.1 for each method). If a protein-term is not found by NLProt, but by the BLAST-searching method or the Gene/Protein-tagging method,a score of +0.1 per method is also added.

  The methods can be compared because they all give the exact position of the term in the text. In this way, the number of false-positives and false-negatives is minimized.
- The second way is to combine data at the time when the Output-XML-file is queried. This is done with the mutation data. As explained above, to filter out false positives, the co-occurrence of the mutation and a term related to “mutation” is looked for. This mutation-related term has to be a result of the BLAST-searching method, also stored in the Output-XML-file.

Specific XPath-Queries have been created to verify this co-occurrence. When searching for interactions between proteins, the XML file is again searched through by an XPath Query.
Let N_p be the number of abstracts where the interaction of two proteins is shown to be positive, N_n the number of abstracts where it is shown to be negative (e.g. inhibition). N_a is the total number of abstracts found where two specific proteins are found to participate in an interaction. The number and type of interaction is computed by $W_t = N_p/N_a$, $W_f = N_n/N_a$. If $W_t > W_f$, it is more likely that the interaction is positive than negative.

Although the probability that an interaction is positive is more likely if $W_t > W_f$, it is not impossible that the interaction is negative (e.g. inhibition) or that, under certain circumstances, both kinds of interactions can occur. Due to this fact, all results are displayed with their respective weights (W_t, W_f).

3 Results and Discussion

In this section we show how we evaluated the system in terms of performance and discuss the results.

3.1 Performance Evaluation

Overview The big problem in most literature-mining systems is the large amount of false-positives in their results. In CONAN, the sheer amount of data rules out a large number of these false positives. When looking at different abstracts about the same topic, we assume that the underlying information should be the same. This means that, if one false-positive result is obtained by computing one abstract and several other true-positive results are acquired, the false-positive is overruled by the true-positives. A perfect example are interaction data in literature. Automatic extraction of interaction data is difficult, especially dealing with false positive results. If abstract A shows that proteins P1 and P2 inhibit each other, abstracts B, C and D on the other hand show that proteins P1 and P2 interact positively, the probability that abstract A is wrong is quite high. The same strategy can be applied for abstracts which have the right information, but this information is extracted wrongly (false-positive). In CONAN, such a false-positive remains in the data, but is overruled by true-positives.

Experiment Design All experiments that are shown, were conducted on an Intel Pentium 4, 2 GHz, 512MB RAM, running SuSE Linux 8.3.

In a first test of the reliability, stability and speed of the system, the latest 100,000 articles published on PubMed were processed. These files were medline04n0576.xml - medline04n0594.xml, including articles from every field. The whole collection of articles is approximately 745 Megabytes big. The computation of the details of all those articles took about two weeks. The resulting

XML files have been merged and used as a basis of analysis. The result was that CONAN is stable enough and fast enough to cope even with a large amount of data. For the final version, quicker and larger machines will be used to cover the whole of MEDLINE.

Test Set Construction Unfortunately, there are no benchmark databases against which the accuracy of CONAN could be measured. With no well-annotated dataset at the moment, in order to analyze interaction data, a dataset of 1,765 abstracts has been created, all of these containing one or more interactions. This was done because a high percentage of MEDLINE abstracts does not contain interactions at all and we only wanted articles including interactions in our set. These 1,765 abstracts are a combination of available lists of PMIDs from BIND [16] and DIP [17], ensuring that the abstracts contained at least one interaction. As these test-datasets are not or only partially annotated, there was a need to annotate a test set by hand, resulting in a more precise annotation. One hundred of those 1,765 abstracts have been selected completely at random by us to ensure that no organism or protein family is overrepresented and those interactions have been manually annotated. This annotation was done by one person only, resulting in no inter-annotator difficulties.

The 100 abstracts is a quite small number compared to the 100,000 abstracts originally processed by our system, but the effort of manually annotating 100,000 abstracts or even 1,765 abstracts would have been too high. This set of 100 manually annotated abstracts is considered by us of being the test-set. Not only interactions were manually annotated in this set, but also biologically interesting keywords (see Section BLAST Searching) and Gene/Protein names (see Section Protein Tagging) have been annotated.

This test-set of 100 abstracts was run by CONAN to give us data to evaluate. We used this test-set of 100 abstracts for all three following evaluations (see Sections BLAST Searching, Protein Tagging and Interactions), thus giving a coherent evaluation of the whole method.

BLAST Searching Firstly, the BLAST searching algorithm was evaluated. Both Precision, defined as the fraction of retrieved relationships that are relevant, and Recall, defined as the fraction of relevant relationships retrieved, have been computed. When analyzing the BLAST-searching method, we see that we get recall and precision comparable to the original method, even though we are using more and much more extensive dictionaries, containing not only Protein and Gene names, as in the original publication, but also more “real language”-terms, like the terms in the “Therapeutic Procedure”-database. Those “real language”-terms cannot be evaluated, because there is overlap between terms and there are no sharp term-boundaries, so we concentrated on Gene and Protein Names. We used the manually-annotated test-set of 100 abstracts (see Section Test Set Construction). By being very strict in our cut-offs, especially with the e-value, we get a higher precision than in the original publication, namely 80%, the recall of our implementation of the method still is 71%.

Protein Tagging When analyzing the different Protein-name-tagging methods, we see that the Data-Integrator-Step boosts the performance of those methods. We again used the same test-set of 100 manually annotated articles (see Section Test Set Construction) as in the evaluation of the BLAST searching. The original NLProt method shows a Precision of 75% and a Recall of 76%, the Protein/Gene-Tagging methods shows a correctness of 77%. Manual annotation showed 480 protein names in the abstracts, whereas CONAN found 504 protein names. When integrating the Protein-Tagging-data with the data found by the BLAST-search and the Gene/Protein-Tagging method, as described in Section 2.4 , we see an increase of Precision to 80.9% (408/504) and of Recall to 85%(408/480), when evaluating Protein names.

The same articles were used as in the evaluation of the Protein-Interaction-Tagging. It has to be said that most articles in this set are Yeast (Saccharomyces cerevisiae)-related articles and Protein-Tagging-Methods usually perform better on Yeast-articles than on articles related to Drosophila, a fact that is also mentioned in [13]. This result is also supported by the good result of the Interaction-Tagging method (see Section Interactions), because the tagging of interaction data is highly dependent on the tagging of Protein Names in text.

Interactions Finally, the protein-protein interactions were evaluated. In this analysis, no distinction was made between different groups of interactions. Positive interactions have been counted as well as negative interactions (e.g. inhibitions). In the 100 manually-annotated abstracts, a total of 427 interactions are documented. Those 427 interactions were manually annotated. CONAN found 477 interactions in total, compared to the 427 interactions which were annotated manually in the abstracts, this yields a number of 50 or more false positives. Analyzing those abstracts achieved a precision of 81.55% (389/477) and a recall of 91.10% (389/427).

Here we see that, by using our system, we get very good results in a fast and easy way, detecting almost all available interactions mentioned in the abstracts.

The main reason why CONAN did not detect the remaining 38 interactions is the failure to recognize certain generalized protein names. Although we use three different methods for tagging Gene/Protein names, we still get false-positives and false-negatives. This is why the regular expressions used for finding interactions cannot give back all results.

But, does it scale ? In Section 3.1, we describe the design of the experiment. We ran the latest 100,000 articles of PubMed with our system, using a single-processor machine. This calculation took 17 days. After indexing the output files with the freely-available XML-indexing software Gnosis (`http://gnosis.cx`), we determined the speed of the queries used in the system. A simple query of those 100,000 articles takes 90 seconds on the same single-processor machine. There are already plans for using a multi-processor, high-memory cluster for this system which will definitely improve speed in processing the articles and querying the results, respectively.

This indicates that CONAN can handle big amounts of data in a fast and reliable way, showing reproducible results. On a reasonably powerful machine, the method can be applied to the whole of MEDLINE, giving the user a tool which shows him all available information.

4 Conclusion

Our goal is to address, as completely as possible, the problem of experimentalists to find certain information "hidden" in the abstracts of biomedical literature. Most systems available at the moment either focus on a very small dataset, a specific organism or specific information (e.g. only interaction data, only Gene Ontology identifiers). We constructed CONAN, which is as complete as possible, offering a wide range of information, from Gene and Protein Names to Mutation Data, Interaction Data and tagging of distinct biomedical entities The big plus of our approach is that we integrate different sources of information to build one system useful for experimentalists.

The results we are obtaining are encouraging. We are performing better or at least as good in the methods we are using (Protein-Interaction Data, Protein/Gene-Name-Tagging), compared to similar systems like Chilibot [18] and present much needed information, like Mutation Data or reliable Interaction Data. The approach presented here provides good methods for all the problems addressed, putting everything into a bigger perspective.

The system itself is currently under consideration of several biologists. Their experience will give us new insights for improving the system and finally putting it accessible to everyone.

Future directions include generating interaction networks from our data, and graphical representation of those networks. Time is required to cover all of MEDLINE, but the ultimate goal is to cover every article published.

Acknowledgements

This work was supported by the Dutch Ministry of Economic Affairs through the Innovation Oriented Research Program on Genomics, grant IGE01017.

References

1. Rebholz-Schuhmann, D., Kirsch, H., Couto, F.: Facts from text–is text mining ready to deliver? PLoS Biol. **3** (2005) e65

2. Krallinger, M., Valencia, A.: Text-mining and information-retrieval services for molecular biology. Genome Biol. **6** (2005) 224
3. Canese, K., Jentsch, J., Myers, C.: The NCBI Handbook. National Center for Biotechnology Information. (2003)
4. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet **25** (2000) 25–29
5. Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., Binns, D., Harte, N., Lopez, R., Apweiler, R.: The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. Nucleic Acids Res. **32** (2004) 262–266
6. Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, R., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N., Yeh, L.S.: UniProt: the Universal Protein knowledgebase. Nucleic Acids Res **32** (2004) D115–D119
7. Birney, E., Andrews, T.D., Bevan, B., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cuff, J., Curwen, V., Cutts, T., Down, T., Eyras, E., Fernandez-Suarez, X., Gzane, P., Gibbins, B., Gilbert, J., Hammond, M., Hotz, H., Iyer, V., Jekosch, K., Kahari, A., Kasprzyk, A., Keefe, D., Keenan, S., Lehvaslaiho, H., McVicker, G., Melsopp, C., Meidl, P., Mongin, E., Pettett, R., Potter, S., Proctor, G., Rae, M., Searle, S., Slater, G., Smedley, D., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Storey, R., Ureta-Vidal, A.: An Overview of Ensembl. Genome Res. **14** (2004) 925–928
8. Krauthammer, M., Rzhetsky, A., Morozov, P., Friedman, C.: Using BLAST for identifying gene and protein names in journal articles. Gene **259** (2000) 245–252
9. Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic local alignment search tool. J Mol Biol. **215** (1990) 403–410
10. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32** (2004) 267–270
11. Tanabe, L., Wilbur, W.: Tagging gene and protein names in biomedical text. Bioinformatics **18** (2002) 1124–1132
12. Horn, F., Lau, A., Cohen, F.: Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors. Bioinformatics. **20** (2004) 557–568
13. Mika, S., Rost, B.: Protein names precisely peeled off free text. Bioinformatics. **20** (2004) I241–I247
14. Mika, S., Rost, B.: NLProt: extracting protein names and sequences from papers. Nucleic Acids Res. **32** (2004) W634–W637
15. Donaldson, I., Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G., Michalickova, K., Pawson, T., Hogue, C.: PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics. **4** (2003) 11
16. Bader, G., Betel, D., Hogue, C.: BIND: the Biomolecular Interaction Network Database. Nucleic Acids Res. **31** (2003) 248–250
17. Xenarios, I., Salwinski, L., Duan, X., Higney, P., Kim, S., Eisenberg, D.: DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Res. **30** (2002) 303–305
18. Chen, H., Sharp, B.: Content-rich biological network constructed by mining PubMed abstracts. BMC Bioinformatics **5** (2004) 147