OntoGene in Biocreative II

Rinaldi, F; Kappeler, T; Kaljurand, K; Schneider, G; Klenner, M; Hess, M; von Allmen, J M; Romacker, M; Vachon, T
OntoGene in Biocreative II

Abstract

Research scientists and companies working in the domains of biomedicine and genomics are increasingly faced with the problem of efficiently locating, in the vast amount of published scientific results, the critical pieces of information that are needed in order to assess current and future research investment. In this paper we describe approaches taken within the scope of the second Biocreative competition in order to solve two aspects of this problem: the detection of novel protein interactions reported in scientific articles, and the detection of the experimental method that was used to confirm the interaction. Our approach is based on a high-recall protein annotation step, followed by two sharp disambiguation steps. The remaining proteins are then combined according to a number of lexico-syntactic filters, which deliver high-precision results, while maintaining a reasonable recall.
OntoGene in Biocreative II

Fabio Rinaldi1
Gerold Schneider1
Jean-Marc von Allmen2

Thomas Kappeler1
Manfred Klenner1
Martin Romacker2

Kaarel Kaljurand1
Michael Hess1
Therese Vachon2

1 Institute of Computational Linguistics, University of Zurich, Binzmühlestrasse 14, CH-8050 Zurich, Switzerland
{rinaldi,gschneid,klenner,kalju,hess}@ifi.unizh.ch

2 Novartis Pharma AG, Basel, Switzerland,
{martin.romacker,jean-marc.von_allmen,therese.vachon}@novartis.com

Abstract

Research scientists and companies working in the domains of biomedicine and genomics are increasingly faced with the problem of efficiently locating, in the vast amount of published scientific results, the critical pieces of information that are needed in order to assess current and future research investment.

In this paper we describe approaches taken within the scope of the second Biocreative competition in order to solve two aspects of this problem: the detection of novel protein interactions reported in scientific articles, and the detection of the experimental method that was used to confirm the interaction.

Our approach is based on a high-recall protein annotation step, followed by two sharp disambiguation steps. The remaining proteins are then combined according to a number of lexico-syntactic filters, which deliver high-precision results, while maintaining a reasonable recall.

1 Introduction

The increasing amount of published scientific results in the domains of biomedicine and genomics poses, to research scientists and companies alike, the problem of efficiently locating the most relevant pieces of information. The research community is therefore keen to adopt novel Text Mining solutions, which have the potential of supporting such discovery process [3]. While there is a broad consensus on the need for Text Mining, there is still a lot of controversy on which of the many possible approaches are most suited for each specific goal.

In this paper we describe experiments performed within the scope of the most recent BioCreAtIvE competition, using tools developed within the scope of the OntoGene project.2 BioCreAtIvE is ideally suited to create the conditions necessary for significant scientific advance in the area of Text Mining.

The OntoGene project aims at developing and refining (semi-)automatic methods for the discovery of interactions between biological entities from the scientific literature. The OntoGene approach is based on dependency-based linguistic analysis of scientific articles [6]. As witnessed by a number of recent publications [1, 2, 4], there is a growing interest in dependency-based representations for the purpose of biomedical Text Mining. One of the advantages of dependency based syntactic representations is that they can be mapped easily into a semantic representation, or, by application of simple transformations, can be used directly to match candidate answers with given queries, allowing easy identification of the arguments of complex relations [5].

In the rest of this paper we describe first the approach followed for subtask 3.2 (IPS). More specifically, section 2 presents our approach to detection of proteins in text, their annotation, and the various disambiguation steps that we have followed. In section 3 we describe how the possible interactions among proteins are generated and selected. Finally, section 4 describes the approach adopted for subtask 3.4 (IMS).

2 Identification and selection of Interactors

It is well known that protein names are highly ambiguous. Researchers working in specific sub-communities tend to develop their own nomenclature, resulting in multiple names for the same protein. Acronyms and abbreviations further complicate the picture. Simply being able to recognize a protein name as such is just a

1http://biocreative.sourceforge.net/
2http://www.ontogene.org/
starting point. The name needs then to be unambiguously qualified, by referring it to an entry into a standard protein database, such as UniProt.\textsuperscript{3}

In order for that to happen, disambiguation must happen at two levels: interspecies (i.e. to which specific organisms does the protein mention refer) and intraspecies (i.e. within a given organism, which specific protein is meant). For example, a protein mentioned in text as eIF4E could refer to a large number of different proteins. A search in the SwissProt section of UniProt (the manually curated section), delivers 46 possible matches. However if the term appears contextually with the mention of a specific organism, like in the sentence “The Cap-binding protein eIF4E promotes folding of a functional domain of yeast translation initiation factor eIF4G1”, then it is reasonable to assume that the name refers to a specific organism (yeast), thus restricting the possible matches in UniProt to the following two: EAP1\_YEAST (eIF4E-associated protein 1) and IF4E\_YEAST (Eukaryotic translation initiation factor 4E). For the task of protein annotation we have adopted a high-recall low-precision term annotation approach, followed by very strict disambiguation steps, which gradually increase precision (at some expense for recall).

UniProt contains for each protein a list of frequently used synonyms. We have built a database which maps the synonyms to the protein identifier. We have further enriched such list using morpho-syntactic rules that generate variants of the synonyms. Starting from a version of UniProt which contained 228670 protein identifiers\textsuperscript{4}, we extracted a list of 203061 unique protein names, and, after generation of the variants, obtained a DB of 698365 terms. Those terms are by necessity highly ambiguous: in average each term refers to 3 proteins, but there are also some terms referring to hundreds of proteins.

Because of the far from perfect HTML-to-text conversion of the articles, we decided early on to use only the abstracts, which we automatically downloaded, in plain text format, from PubMed.\textsuperscript{5} We work on the assumption that the authors will mention in the abstract the most relevant interactions that they discover (although in some cases this might not be true). The input abstracts are tokenized using a custom tokenizer. The stream of tokens is then passed through a DB lookup procedure which tries to determine the longest match possible. As a result of the process, tokens forming terms are grouped together, and their multiple possible values as proteins are associated to them. As an example, the term eIF4E gets 46 different values, such as:

- IF4E\_ASHGO, IF4E\_RAT, IF4E1\_SCHPO, IF4E\_BRAKE, ..., 4EBP2\_HUMAN, 4ET\_HUMAN

Although in a few cases the results described in the articles apply to multiple species, in the majority of cases the article focuses on one (or in some cases 2 or 3) organisms.\textsuperscript{6} Being able to determine with precision which is the organism used in the study leads therefore to a huge disambiguation effect.

For our experiments we have adopted a statistical approach based on the occurrences of the mentions of organisms in the various sections of the paper. Just like for proteins, we have stored in our DB a number of well-known synonyms for the organism (e.g. “murine” is an adjective referring typically to “mouse”).\textsuperscript{7} The relative frequency of species in the sections of the papers are combined linearly, with weights assigned through a learning procedure over a training corpus, and balanced by the known absolute frequency of species in biological research articles (whereby “human” by far outnumbers all other species). Mentions in the abstract tend to have a predominant role in the balanced statistics.

The algorithm delivers a ranked list of species for each article. Such a list is then used to drastically reduce the number of possible interpretations for each term. The first step of disambiguation (organism-based) will simply go through all values for a term, and select those that match the best ranked organism. If that fails to deliver any result, it will proceed with the next organism, according to the ranking, until an assignment is found, or a given threshold is reached.\textsuperscript{8}

Over the BioCreative training data (740 abstracts), the initial annotation step delivers 283556 distinct protein values (P: 0.0072; R: 0.7469).\textsuperscript{9} After the species-based disambiguation step this number is reduced to 45012 (P: 0.0308; R: 0.5763). The remaining ambiguity (intraspecie) needs to be solved by other means.

---

\textsuperscript{3}http://www.expasy.org/sprot/

\textsuperscript{4}We used the file “uniprot\_light\_table.txt”, delivered by the organizers at the beginning of September.

\textsuperscript{5}To be more precise, we analyzed only sentences contained in the abstracts for the detection of protein interactions, but additional information derived from the full articles was used for one aspect of the problem (organism-based disambiguation).

\textsuperscript{6}In the training data, there were 449 articles with interactions involving only 1 organism, 142 articles with 2, 26 articles with 3, 6 articles with 4, 3 articles with 5, 1 article with 6, and 1 article with 9 different organisms (only 628 articles, among those distributed as training data, contained curatable interactions).

\textsuperscript{7}Names and synonyms for organisms were automatically downloaded from NEWT (http://www.ebi.ac.uk/newt/). HTML pages were parsed using the Java-based open-source HTML parser NekoHTML (http://people.apache.org/~andyc/nelk/doc/html/).

\textsuperscript{8}Currently set at 3, i.e. if an assignment is not found in the 3 best ranked organisms, the term is NOT tagged as a protein.

\textsuperscript{9}All P/R/F figures reported in this paper, unless explicitly noted, refer to the training data. Due to lack of space and time, a detailed analysis of the results obtained on the test data was not possible. Such an analysis is being conducted and the results will be presented at the BioCreAtIvE workshop.
The Cap-binding protein eIF-4E promotes folding of a functional domain of yeast translation initiation factor eIF-4G. The association of eucaryotic translation initiation factor eIF-4G with the cap-binding protein eIF-4E establishes a critical link between the mRNA and the ribosome during translation initiation.

This association requires a conserved seven amino acid peptide within eIF-4G that binds to eIF-4E. Here we report that a 98-mer amino acid fragment of S. cerevisiae eIF-4G1 that contains this eIF-4E binding peptide undergoes an unfolded to folded transition upon binding to eIF-4E.

The folding of the eIF-4G1 domain was evidenced by the eIF-4E-dependent changes in its proteinase sensitivity and (1)H-(15) N HSQC NMR spectrum.

Analysis of a series of charge-to-alanine mutations throughout the essential 55.4-kDa core of yeast eIF-4G1 also revealed substitutions within this 98-amino acid region that led to reduced eIF-4E binding in vivo and in vitro.

These data suggest that the association of yeast eIF-4E with eIF-4G1 leads to the formation of a structured domain within eIF-4G1 that could serve as a specific site for interactions with other components of the translational apparatus.

They also suggest that the stability of the native eIF-4E - eIF-4G complex is determined by amino acid residues outside of the conserved seven-residue consensus sequence.

Figure 1: Example of annotated abstract. The tokens marked in red are those identified by the system as protein names, the tokens marked in blue are those identified as organism names, tokens marked in yellow are indicators for a relation, tokens marked in green might suggest the presence of a curatable relation. The green dot on the left of a sentence indicates that the system considers that sentence as potentially containing a “curatable” relation.

With the collaboration of a domain expert, a small set of rules has been developed, which reflects the typical naming conventions made by the authors. For example, the term MRGX, even if we know that it refers to a human protein, is still ambiguous among the following: MRGX1_HUMAN, MRGX2_HUMAN, MRGX3_HUMAN, MRGX4_HUMAN. However, it is a typical convention that, if no further qualifiers are adopted, the term will refer to the first case (MRGX1_HUMAN). Alternatively, where there is a group of proteins characterized by Greek letter suffixes (“-alpha”, “-beta”, etc.), the convention is that the unqualified name usually refers to the “-alpha” variant.10

By sequentially applying the variant rules suggested by the domain expert, the second disambiguation step typically selects one value for each term. Over our collection of 740 abstracts, this reduces the number of possible values to 6351 (P: 0.1311; R: 0.4974). As the figures reveal, one must accept a significant loss of recall at each disambiguation step, in order to reach a minimally satisfactory precision.

3 Identification and selection of Interactions

The training set contains 740 articles obtained from either the IntACT or MINT databases, together with the “gold standard”, i.e. the set of interactions that the curators have identified in each article as novel and relevant (3189 interactions in total). The average number of interactions per article is 4.31, however there are a few articles which contain unusually large number of interactions (the biggest number being 170). According to recommendations by the organizers, we dropped from the training set all articles containing more than 20 interactions. This left 719 articles, of which actually only 628 do contain interactions (for a total of 1900 interactions, average 3.07 interactions per article).

Once reasonable values have been reached in the task of detecting proteins, the next problem to be tackled is that of identifying their possible interactions. A naive approach would simply consist of generating all possible pairs of proteins mentioned in each single abstract. This results in a recall of almost 35%, however at the cost of an abysmal precision.11 Another simple approach consists in enforcing a maximal distance (in number of tokens) between any 2 mentions of the proteins. We have experimented with varying distances from 1 to 50 (without taking into account sentence boundaries), and found the best F-measure value at the distance of 9 (P: 0.0460; R: 0.1765; F: 0.0729).

The conceptually simpler (and more intuitive) approach of restricting the possible combinations to proteins within the same sentence, without requiring any maximal distance, delivers better results (P: 0.0494; R: 0.2077; F: 0.0798).

There are a few well-known exceptions, such as “immune interferon” (which is normally used to refer to “interferon-gamma”).

We decided against submitting such results, although this might have given us better scores for recall, because we think that results with precision inferior to 1% are in any case of little use.
The predicate `dep(TYPE,HEAD,DEPENDENT)` represents dependencies between heads of chunks, the predicate `prot(PROT)` identifies a protein.

\[
\begin{align*}
&\text{dep}(\text{subj},\text{bind},\text{Daxx}), \\
&\text{dep}(\text{pobj},\text{bind},\text{Mdm2}), \\
&\text{dep}(\text{conj},\text{Mdm2},\text{Hausp}), \\
&\text{dep}(\text{prep},\text{Mdm2},\text{to}), \\
&\text{prot}(\text{Daxx}), \\
&\text{prot}(\text{Mdm2}), \\
&\text{prot}(\text{Hausp}).
\end{align*}
\]

Figure 2: Example of dependency tree (internal representation on the left, graphical visualization on the right)

Still, while recall is relatively good (considering the limitations of the protein detection phase), precision appears too low for a practical application of the approach proposed. Therefore a further filtering phase is required to select among the proposed interactions only those really relevant. In this respect, two kinds of “false positives” need to be distinguished. On the one hand, there are pairs which correspond to interactions mentioned by the authors, but which are not relevant to the curation task, either because they are well-known interactions, or because they play a secondary role with the main interactions described. On the other hand, there are genuinely spurious protein pairs, which are not described by the authors as interacting, but are simply a product of the simplistic way in which the pairs are generated. The strategies to filter out the false positives need therefore to address both problems.

In the first case, the approach that we have followed is to try to identify in each abstract the sentences that describe the most relevant results according to the authors, and distinguish them from the sentences that describe background results, an example of which could be the following: “Previous studies have revealed a genetic interaction between DLG and another PDZ scaffolding protein, SCRIBBLE (SCRIB), during the establishment of cell polarity in developing epithelia.”

An example of a sentence that reports ‘curatable’ results is the following: “Here we report the isolation of a new DLG-interacting protein, GUK-holder, that interacts with the GUK domain of DLG and which is dynamically expressed during synaptic bouton budding.”

In order to distinguish between background and novel information, we adopted a machine learning approach based on a classifier\(^\text{12}\) which takes as training data the lemmatized version of sentences whose status has been determined on the basis of the gold standard. A sentence is considered positive if it contains at least one pair of proteins belonging to one of the gold standard interactions for the abstract to which the sentence belongs (see figure 1). After application of the ‘novelty’ filter the results that we obtained on the training data are the following: (P: 0.0945; R: 0.1992; F: 0.1282).

The second problem can be dealt with by taking into account the exact syntactic configuration in which the two proteins appear, i.e. does the context form a meaningful interaction? For example, in the sentence “\textit{Daxx simultaneously binds to Mdm2 and the deubiquitinase Hausp}” three possible interactions can be considered (the direction of the interaction is presently ignored):

1. \textit{Daxx} – \textit{Mdm2} \\
2. \textit{Daxx} – \textit{Hausp} \\
3. \textit{Mdm2} – \textit{Hausp}

However, on syntactic grounds (see figure 2), only the first 2 interactions are licensed, while the third is not justified. We have developed a series of lexico-syntactic filters, which are applied in a cascade to each proposed interaction. The filters make use of lexical, morphological and syntactic information delivered by a pipeline of NLP tools, including a novel dependency parser (for more details see \[5\]). For example, filters capturing the interactions shown in figure 2 are (using a simplified notation):

\[
\begin{align*}
\text{int}(X,Y) & \leftarrow \text{dep}(\text{subj},H,X), \text{dep}(\text{pobj},H,Y), \text{prot}(X), \text{prot}(Y). \\
\text{int}(X,Z) & \leftarrow \text{dep}(\text{subj},H,X), \text{dep}(\text{pobj},H,Y), \text{dep}(\text{conj},Y,Z), \text{prot}(X), \text{prot}(Z).
\end{align*}
\]

Only if at least one of such filters applies to the specific case, the interaction is further considered. The results that we obtain on the training data are (P: 0.5437; R: 0.1839; F: 0.2749). In order to enhance the usefulness and maintainability of the lexico-syntactic filters, a special type of visualization has been created (see figure 3) which shows for each sentence and each interaction potentially therein contained, which filter captures the given interaction.

\(^{12}\)We used the Rainbow tool (http://www.cs.cmu.edu/~mccallum/bow/rainbow/) and tested different methods, obtaining the best results with an SVM approach.
4 Identification of the Interaction Method

The original idea for this subtask was to compare two methodologies, pattern matching (supplemented by simple statistics) and machine learning. As the resources for this subtask were extremely limited and time was running short, this comparison had to be postponed, so only the results of the pattern matching approach were submitted. Pattern matching was done on a full-text version of the articles, as many abstracts don’t mention all methods, nor any hints for them. These are normally mentioned in the “Methods and Materials” section.

The first important decision for this pattern matching approach was that — considering the limited resources and time budget — patterns for most methods could not be written by hand. So we started with the method part of the PSI-MI ontology and took the official names, synonyms and exact synonyms of the methods given there as baseline. These patterns were then supplemented by patterns automatically derived from the baseline patterns by considering several well-known variations such as insertion of spaces and hyphens (everywhere), deletion of spaces or hyphens (between words), interpolation of words (between words), truncation of heads etc. In this phase, just as in the next one, recall improvement was the primary goal.

The selection of methods for which patterns should be written by hand was based on the frequency of the methods in the training data and the recall and precision of the automatically derived patterns. As just 5 methods account for two thirds of all file-method-pairs in the training data, these were carefully looked at by our team’s biologist, who tried to find additional hints in some of the papers where the methods were not found by the automatically derived patterns. The method ‘coimmunoprecipitation’ (MI:0019) and its hyponyms ‘anti tag coimmunoprecipitation’ (MI:0007) and ‘anti bait coimmunoprecipitation’ (MI:0006) were most successfully treated that way, because they are extremely frequent in the training data and at the same time seldom recognized by the automatically derived patterns. After identifying files as containing one of the coip methods, the most important problem was the very low precision for most hints with good recall (e.g. “antibod” predicts ‘anti bait coimmunoprecipitation’ (MI:0006) with recall 0.985 and precision 0.299) and the low recall for most hints with good precision (e.g. “flag-tagged” in combination with “precipitat” predicts ‘anti tag coimmunoprecipitation’ (MI:0007) with recall 0.434 and precision 0.543).

This could be overcome by a back-off algorithm, starting with the patterns with best precision (assigning their methods and excluding other coip methods), continuing with patterns with a lower precision (assigning their methods non-exclusively) and ending with a default (MI:0019).

Similar approaches for ‘pull down’ (MI:0096) led to much less improvement because the results for the automatically derived patterns were already rather good. This was even more so for the 5th method, ‘two
hybrid’ (0018), so the handcrafted patterns for this method were abandoned.

‘Imaging techniques’ (MI:0428) was selected for a handcrafted pattern because recall was very bad. It was improved significantly by deriving the new pattern from obsolete method names which have to be mapped to MI:0428 as they don’t figure in PSI-MI 2.5 any more. An improvement in precision for ‘biochemical’ (MI:0401) could be made by coupling the very imprecise pattern with other, more precise hints.

The pattern matching at this stage resulted in about 6.8 candidates per file with good recall (0.734) but bad precision (0.243). Obviously the number of candidates had to be reduced to a degree comparable to the training data. For this, every candidate (method) was given a weight influenced by its frequency in the training data and the precision and recall of the patterns used to detect it.

For the 3 runs to be submitted we decided on the following degrees of reduction: run 1, giving only the best candidate (and so the highest precision), was coupled with the results of the highest-precision-run for subtask 3.2. Run 2, giving the 3 best candidates (for best recall) was coupled with the results of the highest-recall-run for subtask 3.2 and run 3, giving the best F-measure by selecting up to 3 best candidates (additional condition was that candidate 2 and 3 reached a minimum in frequency and precision) was coupled again with the results of the highest-recall-run for subtask 3.2. As the interactants were identified in the abstracts only, whereas the methods were identified in the full text, no direct allocation of methods to specific interactant-pairs could be achieved. So we allocated every method for a file to all its interactant-pairs.

Pattern-matching just on the isolated “methods and materials” chapters of the articles without candidate-reduction had much higher precision than the unreduced pattern-matching of the full text, but after candidate-reduction the results for the full-text pattern-matching were slightly better.

5 Conclusions

This paper presents an approach, directed at the extraction of protein-protein interactions from biomedical literature, based on sequential filtering of candidate interactions (pairs of proteins in sentences). The filters make use of linguistic information derived from a pipeline of NLP tools, in particular including a dependency parser. Further, a pattern-based approach is capable of recognizing the most frequently used experimental methods with a high reliability. The results show that the proposed approach is competitive.

6 Acknowledgments

We thank the anonymous reviewer(s) for their helpful and insightful comments.

References

[1] Andrew B Clegg and Adrian J Shepherd. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8:24, 2007.

[2] Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx — Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[3] Martin Krallinger and Alfonso Valencia. Text-mining and information-retrieval services for molecular biology. *Genome Biology*, 6(7):224, 2005.

[4] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8:50, 2007.

[5] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Michael Hess, and Martin Romacker. An Environment for Relation Mining over Richly Annotated Corpora: the case of GENIA. *BMC Bioinformatics*, 7(Suppl 3):S3, 2006.

[6] Gerold Schneider. *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis, Institute of Computational Linguistics, University of Zurich, 2007.