BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 6 2012, pages 867–875
doi:10.1093/bioinformatics/bts042

Data and text mining

Advance Access publication January 27, 2012

Literature mining of host–pathogen interactions: comparing feature-based supervised learning and language-based approaches

Thanh Thieu[1], Sneha Joshi[2], Samantha Warren[1] and Dmitry Korkin[1,2,3,*]

[1]Department of Computer Science, [2]MU Informatics Institute and [3]Bond Life Science Center, University of Missouri, Columbia, MO 65211, USA.

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: In an infectious disease, the pathogen's strategy to enter the host organism and breach its immune defenses often involves interactions between the host and pathogen proteins. Currently, the experimental data on host–pathogen interactions (HPIs) are scattered across multiple databases, which are often specialized to target a specific disease or host organism. An accurate and efficient method for the automated extraction of HPIs from biomedical literature is crucial for creating a unified repository of HPI data.

Results: Here, we introduce and compare two new approaches to automatically detect whether the title or abstract of a PubMed publication contains HPI data, and extract the information about organisms and proteins involved in the interaction. The first approach is a feature-based supervised learning method using support vector machines (SVMs). The SVM models are trained on the features derived from the individual sentences. These features include names of the host/pathogen organisms and corresponding proteins or genes, keywords describing HPI-specific information, more general protein–protein interaction information, experimental methods and other statistical information. The language-based method employed a link grammar parser combined with semantic patterns derived from the training examples. The approaches have been trained and tested on manually curated HPI data. When compared to a naïve approach based on the existing protein–protein interaction literature mining method, our approaches demonstrated higher accuracy and recall in the classification task. The most accurate, feature-based, approach achieved 66–73% accuracy, depending on the test protocol.

Availability: Both approaches are available through PHILM web-server: http://korkinlab.org/philm.html

Contact: korkin@korkinlab.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on March 11, 2011; revised on November 14, 2011; accepted on January 18, 2012

1 INTRODUCTION

Infections are complex biological processes that target host organisms from virtually all kingdoms of life and involve a variety of microbial pathogens such as viruses, bacteria, fungi, protozoa, multicellular parasites and even proteins (Anderson and May, 1979; Mandell and Townsend, 1998). The pathogen's strategy to enter the host organism and breach the host's immune defenses often involves interactions between the host and pathogen proteins (Dyer *et al.*, 2010; Konig *et al.*, 2008). Systematic determination and analysis of host–pathogen interactions (HPIs) provides a challenging task for both experimental and computational approaches, and is critically dependent on previously obtained knowledge about these interactions. For example, several bioinformatics approaches apply the homology information to predict new HPIs, characterize the interaction structures or find conservation patterns across HPI networks (Davis *et al.*, 2007; Dyer *et al.*, 2007; Dyer *et al.*, 2010; Franzosa and Xia, 2011). Other approaches, either manual or reliant on the existing databases, collect the HPI molecular or genetic data into a centralized repository (Driscoll *et al.*, 2009; Kumar and Nanduri, 2010; Winnenburg *et al.*, 2008). However, a fully automated system for extracting molecular HPI data directly from the biomedical literature is yet to be built.

Rapid growth of published biomedical research has resulted in the development of a number of methods for biomedical literature mining during the last decade (Krallinger and Valencia, 2005; Rodriguez-Esteban, 2009). The methods dealing with the biomolecular information are generally divided into three categories based on the domain of biomedical knowledge they target: (i) automated protein or gene identification in a text (Mika and Rost, 2004; Seki and Mostafa, 2005; Tanabe *et al.*, 2005); (ii) literature-based functional annotation of genes and proteins (Chagoyen *et al.*, 2006); and (iii) extracting the information on the relationships between biological molecules, such as proteins and RNAs or genes (Hu *et al.*, 2005; Lee *et al.*, 2008; Shatkay *et al.*, 2007). The relationships detected by the methods from the third category range from a co-occurrence of the genes and proteins in a text (Hoffmann and Valencia, 2005) to detecting the protein–protein interactions (PPIs) (Blaschke and Valencia, 2001; Donaldson *et al.*, 2003; Marcotte *et al.*, 2001) and identification of signal transduction networks and metabolic pathways (Friedman *et al.*, 2001; Hoffmann *et al.*, 2005; Santos and Eggle, 2005).

Related to the problem of mining HPIs, the problem of extracting general PPIs from the text has received an increasing amount of attention from the community. A number of approaches have been recently proposed to extract PPIs from text. A basic approach determines an occurrence of a PPI by detecting the co-occurrence of names of the interacting proteins or genes in the same sentence (Stephens *et al.*, 2001). A more advanced approach relies on pattern matching to capture the semantic structure from the text phrases

*To whom correspondence should be addressed.

that may contain the protein interactions. These patterns can be defined either manually or by an automated method using dynamic programming (Corney *et al*., 2004; Hao *et al*., 2005; Huang *et al*., 2008; Leroy and Chen, 2002). Another approach employs a feature-based machine learning classifier (Donaldson *et al*., 2003). The last approach is based on the natural language processing (NLP) (Ahmed *et al*., 2005; Yang *et al*., 2009). The current state-of-the-art methods in NLP-based text mining use link grammar, a context-free grammar that relies on a dictionary of rules (linking requirements) to connect, or 'link', pairs of the related words (Sleator and Temperley, 1995). The link grammar-based methods have already been successfully applied to extract PPIs (Pyysalo *et al*., 2004).

The problem of PPI literature mining has been a recent focus of the BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology) initiative, a community-wide effort for evaluating biological text mining systems (Hirschman *et al*., 2005; Krallinger *et al*., 2008). The initiative defines three PPI mining subtasks: (i) classification of PPI-relevant documents; (ii) identification of sentences with PPIs; and (iii) identification of interacting protein pairs. The computational approaches developed for these subtasks range from machine learning methods for the first task to natural language processing methods for the second and third subtasks.

While similar to the problem of mining PPIs, the challenges of mining HPIs are further elevated by introducing an additional requirement for the extracted interaction to be shared exclusively between the proteins of host and pathogen organisms. In this article, we develop and compare two approaches for HPI text mining from the abstract or title of a PubMed publication. The first approach is a supervised learning feature-based approach that employs Support Vector Machines, while the second approach is a language-based approach that employs link grammars.

2 METHODS

Problem formulation: similar to the way the BioCreAtIvE initiative defines three types of PPI mining subtasks (Hirschman *et al*., 2005), we define and address the following three subtasks for the problem of HPI mining:

HPI Mining Task 1: given an *expanded* abstract, which includes both title and abstract of a biomedical publication, determine whether it is *HPI-relevant*, i.e. whether it contains the following HPI information: (i) host and pathogen proteins/genes interacting with each other and (ii) names of the corresponding host and pathogen organisms.

HPI Mining Task 2: given an expanded abstract containing HPI information, determine specific sentences that include this information.

HPI Mining Task 3: given an expanded HPI-relevant abstract, determine specific pairs of host and pathogen proteins/genes participating in the interactions and the corresponding organisms.

2.1 Feature-based approach

General description: the feature-based approach includes five basic stages (Fig. 1A). First, each abstract is preprocessed, and proteins/genes together with the organism names are detected. Second, for each abstract a feature vector is generated. Third, a supervised learning system is trained to identify HPI-relevant abstracts (Task 1) by providing the feature vectors generated from (i) manually curated abstracts that contain experimental evidence of interactions between the host and pathogen proteins as well as (ii) abstracts that are annotated as not HPI-relevant. Fourth, each positively classified abstract is processed to identify the specific sentences containing HPI information (Task 2), determine how certain this information is, and extract the protein/gene and organism information from the sentences (Task 3). Finally, the trained system is used on a new testing set of HPI-relevant and HPI-irrelevant abstracts to assess the approach.

Text preprocessing: an abstract is split into individual sentences by defining the title as a separate sentence and detecting sentence termination patterns throughout text of the abstract. A basic pattern of a period (.), followed by a space and a capitalized letter is directly used to distinguish sentences in a standard text. However, this approach has its limitations when applied to biomedical literature, due to the presence of periods in the names of proteins, abbreviations such as 'i.e.', 'e.g.', 'vs.', etc. Thus, in our approach such words are first identified using a dictionary; the periods are then replaced by spaces, and the above basic pattern is applied.

Next, each abstract is preprocessed to detect proteins/genes and the corresponding organism names using an entity tagging software NLProt (Mika and Rost, 2004). NLProt combines the dictionary search, rule-based detection and feature-based supervised learning to extract the names of proteins and genes and tag them using SWISS-PROT or TrEMBL identifiers (Boeckmann *et al*., 2003) as well as predict the most likely organisms associated with these proteins. It was reported to have a precision of 75% and a recall of 76% on detecting proteins/genes (Mika and Rost, 2004). In addition, our dictionary on host and pathogen organisms is used to search and tag the organism names in the abstract.

Support vector machines: the problem of determining an HPI-relevant abstract can be naturally formulated as a problem of supervised binary pattern classification. Given a training set of n abstracts, each represented as a vector of N numerical features, $\boldsymbol{x}^i = (x_1, x_2, \ldots, x_N)$, and their classification into one of the two classes of abstracts that either contain or do not contain HPI information, $\boldsymbol{y} \in \{-1,1\}$, a binary classifier of abstracts is trained based on these data. Once trained, the classifier can assign a class label from $\boldsymbol{y}$ for any new abstract $\boldsymbol{x}$. Here, we use support vector machines (SVMs) (Vapnik, 1998), a supervised learning approach, which is well established in bioinformatics and has been recently applied to address the HPI abstract classification problem (Yin *et al*., 2010). The linear and two widely used non-linear kernel functions are applied and compared: the polynomial kernel, $K^P(x.x') = (\langle x.x' \rangle + 1)^d$, where d is degree of the polynomial, and Gaussian radial basis function (RBF), $K^G(x.x') = \exp(-||x - x'||^2/c)$. In this work, *libsvm* software is used for SVM implementation (http://www.csie.ntu.edu.tw/~cjlin/libsvm).

Feature vectors: for each abstract a 12-dimensional feature vector, $x = (x_1, x_2, \ldots x_{12})$, is generated. The majority of features used for this method relies on the keywords related to the HPI topics. Features x_1 and x_2 reflect the presence of host and pathogen proteins/genes in the abstract. These features are determined based on the entity tagging obtained by NLProt (Mika and Rost, 2004). Each protein is then classified as a host or pathogen protein based on the source organisms provided either by NLProt or extracted directly from the abstract by searching our dictionary of host and pathogen organisms. Features x_3 and x_4 specify the number of occurrences for the host and pathogen organism names and are defined using the NLProt organism annotation as well as our dictionary of host and pathogen organisms. The dictionary was built using the set of organisms extracted from several databases (Driscoll *et al*., 2009; Kumar and Nanduri, 2010; Winnenburg *et al*., 2008) together with their synonyms and common names extracted from NCBI Taxonomy (http://www.ncbi.nlm.nih.gov/Taxonomy/), followed by adding generic keywords, such as' 'pathogen', 'host', 'plant', etc. Binary feature x_5 reflects the presence or absence of the general PPI keywords in the abstract and is obtained by scanning the extended abstract against the interaction keyword dictionary (Table 1). Features x_6 and x_7 describe additional statistics on PPI keyword occurrences and are defined as the percentage of interaction keywords with reference to the total number of words, and number of sentences containing the interaction keywords with reference to the total number of sentences in the abstract, correspondingly. Feature x_8 takes into consideration the *typicality* of each keyword in the

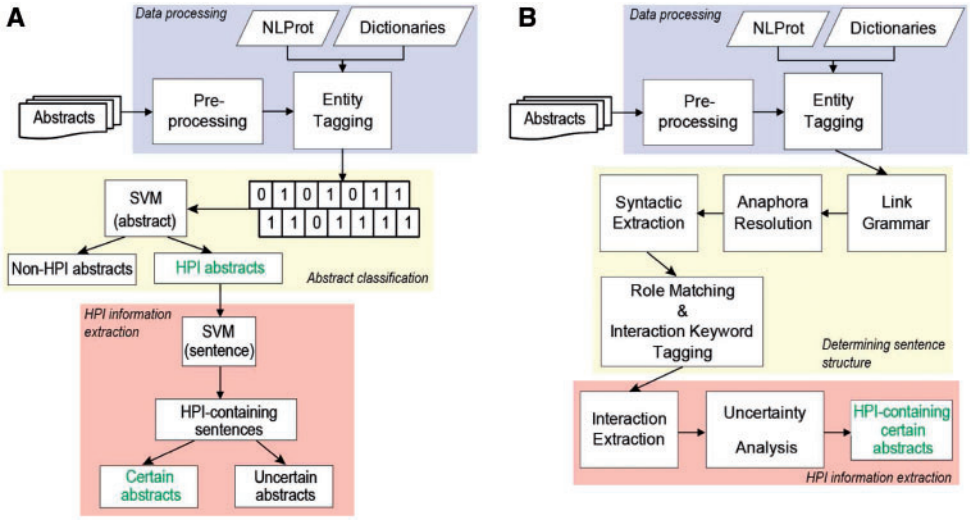

Fig. 1. Two literature mining approaches for classifying HPI-containing abstracts and extracting HPI data. (**A**) Feature-based supervised machine learning approach. (**B**) Language-based approach. Both approaches require data preprocessing, including entity tagging. The first approaches uses a set of manually annotated abstracts to train an SVM classifier, which then can be used to determine if an abstract contains HPI information followed by data extraction for the determined instance of HPI. In the language-based approach, the structure of each sentence is extracted using a link grammar parser. Then the set of structural patterns is used to determine if the sentence contains HPI information. The same patterns are used to extract specific information about HPI.

Table 1. Keyword dictionaries used in both approaches

Dictionary name	*N*	Examples
Interaction keywords	54	*Interact, associate, bind, complex*
Experimental keywords	28	*Yeast two-hybrid, chemical crosslinking*
Negation keywords	11	*Not, neither, inability, none, incapable*
HPI-specific keywords	17	*Virulence factor, effectors*
Host names	309	*Host, plant, human, Mus musculus*
Pathogen names	349	*Listeria monocytogenes, Hepatitis virus*
Uncertainty keywords	33	*May, suggest, assume*

N is the number of unique entries for each dictionary.

literature. The typicality of a keyword is defined as the percentage of abstracts in the training set containing the keyword. Feature x_8 is then defined as a sum of typicalities for all PPI keywords in the abstract. Since most papers describing a PPI also provide the experimental evidence supporting the interaction, an experimental keyword feature x_9 is defined as the number of experimental keywords in the abstract, detected by scanning the abstract against the experimental keyword dictionary (Table 1). Some abstracts report the absence of an interaction between a host and pathogen proteins. This information may be difficult to recognize by a feature-based approach, since the abstract would usually contain the information similar to a true HPI abstract with the exception of certain negation keywords present in a false HPI abstract (Table 1). Feature x_{10} accounts for such keywords and is defined as the percentage of negation keywords from the corresponding keyword dictionary with reference to to the total number of words in the abstract. Feature x_{11} allows for estimation as to whether a negation keyword is applied specifically to the information about a HPI in the abstract. The feature is defined as the number of words between interaction keyword and negation keyword in a sentence. Feature x_{12} accounts for the HPI-specific keywords, such as *virulence, effectors, etc*. and is calculated as a percentage of such keywords with reference to the total number of the abstract words.

Supervised training and classification using SVM: an SVM classifier is trained using HPI-relevant and HPI-irrelevant abstracts as positive and negative training sets, correspondingly. Once trained, the same classifier is used in our method twice: first, to classify whether an abstract is HPI-relevant, and secondly, if it is relevant, to determine which sentences of the abstract are most likely to contain the HPI-relevant data. For the latter, we generate a feature vector for each sentence and use it as an input to the SVM classifier.

The accuracy of an SVM-based classifier generally depends on a number of parameters that can be optimized during the training. In this work, two main parameters were optimized, C and γ. The error cost parameter, C, controls the trade-off between allowing training errors and forcing rigid margins. Together with γ, the parameters are selected by evaluating the accuracies of trained models using leave-one-out cross-validation. The values of C range from 2 to 20, while values for γ range from 2^{-10} to 2^{1}. The model of maximum accuracy is selected as a final model.

Handling information uncertainty: abstracts containing HPI information often report the interactions directly from the experimental data. However, there are cases where an HPI is suggested as a hypothesis. We refer to such cases as the *uncertain* HPI data. While distinguishing uncertain HPI data is not the focus of this work, a simple criterion to detect the uncertainty in the positively classified abstracts has been implemented: if at least one uncertainty keyword (Table 1) is found to directly precede the interaction keywords in at least one of the sentences classified by our SVM as containing HPI information, the abstract is considered to be uncertain.

2.2 Language-based approach

General description: the second approach relies on a language-based formalism. Specifically, it uses link grammars (Sleator and Temperley, 1995). Our approach is similar to the language-based systems that extract information on general PPIs. However, in HPI text mining, there are additional challenges that necessitate adding new modules to the computational pipeline, compared with a pipeline for extracting general PPIs. The main challenges include correctly associating the organism name for each protein, ensuring that the extracted interaction is an inter- and not intraspecies interaction, and combining the information about an HPI from multiple sentences.

Method organization: the HPI mining pipeline consists of the following eight steps (Fig. 1B): (i) text preprocessing; (ii) entity tagging, where proteins/genes and organism names are identified; (iii) grammar parsing, where the input text is parsed into dependency structures; (iv) anaphora resolution, where the references to pronouns are determined; (v) syntactic extraction, where a complex sentence is split into simple ones; (vi) role matching, where semantic roles are determined in each simple sentence; (vii) interaction keyword tagging; and (viii) extraction of the actual HPI information. In contrast to the feature-based approach, the language-based approach directly addresses Tasks 2 and 3 by finding the sentences containing HPI information and extracting the corresponding pairs of host and pathogen organisms and the interacting proteins/genes. Task 1 is addressed by classifying an abstract as HPI-relevant if there is at least one HPI with the *complete* information (i.e. host and pathogen protein/genes and the corresponding organisms) extracted from abstract's text.

Entity tagging: the entity tagging module identifies the named entities, including proteins/genes and the names of organisms associated with the proteins/genes. For the language-based approach, it is crucial that the source organism of a protein (which can be either a host or a pathogen) is correctly identified. As a result, an entity tagging module more detailed than in the feature-based approach entity is introduced. The module consists of three steps: protein/gene tagging using NLProt, host/pathogen organism dictionary-based matching and post-processing.

First, the NLProt tagger is applied to detect the proteins/genes. Next, using UniProt accession number (Bairoch *et al*., 2005), all synonyms of the same protein/gene are grouped into an object called a protein/gene entity by the module. The organisms predicted by NLProt are then identified as a host or a pathogen by selecting the longest match when searched against an expanded version of the organism dictionary (Section 2.1). The dictionary is expanded to include all organism synonyms and group them under NCBI Taxonomy IDs (Wheeler *et al*., 2006). Since all organisms in the abstract may not be identified by NLProt, the abstract is rescanned to find the remaining host and pathogen organisms.

While in the previous two steps detecting a protein/gene is independent of detecting the organism name, in the post-processing step the mutual context information is used to further improve the detection accuracy. Specifically, our system uses the phrase structure provided by link grammar to (i) find additional host/pathogen information that is not present in the dictionary and (ii) reassign a protein/gene to its correct organism, if needed. Due to incompleteness of the dictionary (not all organisms may be covered) as well as its ambiguity (the same organism can be both, a host and a pathogen), it is important to consider the contextual information when mining the data associated with an HPI. Specifically, if an organism name detected by NLProt is not found in our dictionary, the module nevertheless tries to assign its role as a host or pathogen by searching for generic keywords (such as 'host', 'pathogen', 'pathogenic', 'pathogenesis', *etc*.), in each phrase that contains the organism name. Using a similar procedure, the module associates the proteins/genes with the associated organism name by identifying the organisms name in the phrase that contains a protein/gene. The newly obtained information then replaces the current suggestions provided by NLProt. To associate a protein/gene to a corresponding organism, two search patterns are implemented:

P1. Organism name + protein name (e.g. '*Arabidopsis* RIN4 protein');

P2. Protein name + preposition + organism name (e.g. 'RXLX of human').

For instance, in the phrase 'the *Arabidopsis* RIN4 protein', NLProt associates RIN4 with a pathogenic organism, while the dictionary matches *Arabidopsis* as a host organism and the post-processing identifies this phrase as pattern P1. Therefore, *Arabidopsis* is assigned to be the organism that contains RIN4 protein, correctly assigning RIN4 as a host protein.

Link grammar parsing: link grammar is a context-free grammar with an implicit dependency property (Sleator and Temperley, 1995). It relies on a set of rules (linking requirements) that allow the linkage of pairs of related words. The link grammar defines a sentence as a sequence of words. For each sentence the links connecting the pairs of words are determined, such that the following three conditions apply: (i) the links do not cross (planarity);(ii) each word is connected to at least one other word by a link (connectivity); and (iii) the linking requirements for each word in the sentence are not violated (satisfaction). For example, the linkage for the sentence 'Avirulence protein B targets the *Arabidopsis* RIN4 protein' is shown in Figure 2. In total, the link grammar has 107 main links, each of which can derive many sublinks.

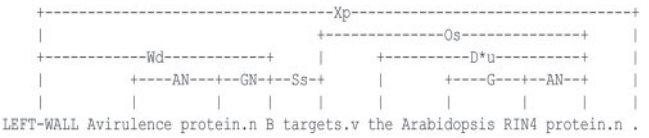

Fig. 2. An example of internal sentence structure extracted by a link grammar parser. Words in the sentence are labeled with the part-of-speech tags, such as .n (noun) and .v (verb). Xp, Os, D*u, G, An, Wd, Ss correspond to different types of connectors that are used to form links between the words.

Our system is based on the link grammar parser from the open source project AbiWord (http://www.abisource.com/projects/link-grammar/). Here, the original link grammar (Sleator and Temperley, 1995) is implemented and customized with additional features, including an adaptation of the parser to the biomedical sublanguage, BioLG (Pyysalo *et al*., 2006) and an English-language semantic dependency relationship extractor, RelEx (Fundel *et al*., 2007). To improve handling long sentences and reducing the context ambiguity by the link grammar, several of the best parses of each sentence are processed (the number of parses has been empirically set to 10).

A three-layer entity framework: next, a three-layer entity framework is constructed for the entity tagging module (Fig. 3). The bottom layer contains a set of real entities defined by a UniProt accession number (for proteins/genes) or an NCBI Taxonomy ID (for organism names). A real entity may contain several textual entities, (proteins/genes or organism names) that are obtained from multiple text sentences, if these names are synonyms. The middle layer consists of the set of all sentences in the abstract. At the middle layer, each textual entity is connected to a unique real entity. The top layer consists of the best link grammar parses selected for each sentence from the abstract. A single sentence can result in multiple link grammar parses; therefore, at the top level, one or several link grammar nodes are connected to a single textual entity. Once there are changes in the assignment of a host/pathogen role of an organism or an organism associated with a protein/gene, the entity framework allows the post-processing module to cascade the changes to the real entity, which is automatically reflected back to related textual entities.

Anaphora resolution: in this module, the semantic meaning of pronouns (e.g. 'it', 'they') and other language structures in a sentence is determined. The information on HPIs in an abstract often spans multiple sentences, with the names of organisms or proteins/genes often being replaced by pronouns. Therefore, it is important to have an accurate anaphora resolution module to extract the complete HPI information. Our system uses a RelEx anaphora resolution module, which employs Hobbs' pronoun resolution algorithm (Hobbs, 1978). For each anaphora, the module first produces a list of possible antecedents. Then, it resolves the first antecedent, which is assumed to be the most probable one. For instance, in the sentence 'The *Pseudomonas syringae* type III effector protein avirulence protein B (AvrB) is delivered into plant cells, where it targets the *Arabidopsis* RIN4 protein', 'it' is resolved by the anaphora resolution module as 'The *Pseudomonas syringae* type III effector protein avirulence protein B (AvrB)'.

Syntactic extraction: sentences in a biomedical text are often complex, consisting of two or more simple sentences, where a simple sentence consists of four components:

Subject (*S*)+Verb (*V*)+Object (*O*)+Modifying phrase of verb (*M*).

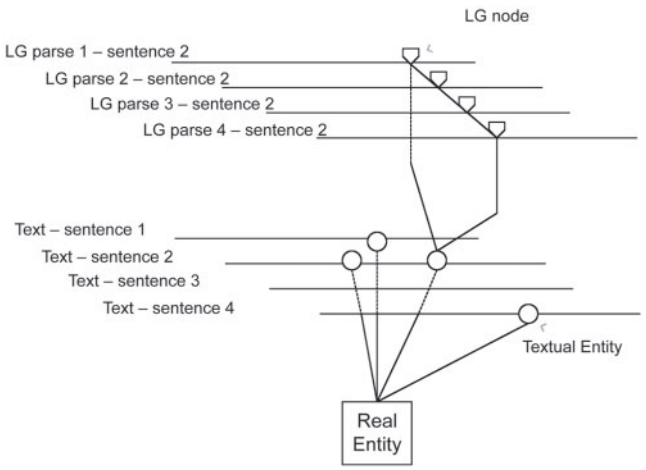

Fig. 3. A three-layer entity framework. The framework connects named entities at different levels of text representation and processing. The real entities at the bottom level correspond to UniProt accession numbers (for proteins/genes) or NCBI Taxonomy IDs (for organisms). The textual entities of the middle layer correspond to protein/gene or organism names. The entities at the top layer correspond to the link grammar parses.

Our system's syntactic extractor module, designed to detect the simple sentences, is built based on the idea of the automated extractor InTex (Ahmed *et al*., 2005). A sentence is scanned by the module to find all links of one of the four types: *S*, *RS*, *O* and *MV*. An *S*-link connects a subject to a verb (the subject should be located before the verb in the sentence), whereas an *RS*-link connects a verb to a subject (the subject should be located after the verb in the sentence). An *O*-link connects a verb to an object. The last link type, a *MV*-link, connects a verb to a modifying phrase. First, each sentence is scanned to find all *S*-links and *RS*-links. Each such link is defined to be the beginning of a simple sentence. Secondly, from a verb connected with an *S*-link or an *RS*-link, all possible verb phrases, adverb phrases or adjective phrases before and after the verb are determined, specifying the verb range. Thirdly, the module determines all objects by checking if there are any *O*-links for the verb in the verb range. Similarly, the module determines all modifying phrases of the verb by finding all *MV*-links. For example, after executing the syntactic extractor module, the sentence 'The Pseudomonas syringae type III effector protein avirulence protein B (AvrB) is delivered into plant cells, where it targets the Arabidopsis RIN4 protein' is split into two simple sentences: 'The Pseudomonas syringae type III effector protein avirulence protein B (AvrB) is delivered into plant cells' and 'The Pseudomonas syringae type III effector protein avirulence protein B (AvrB) targets the Arabidopsis RIN4 protein'.

Interaction keyword tagging: to tag interaction keywords in an abstract, our system first finds the word stems using a lexical database WordNet (Fellbaum, 1998). WordNet contains nouns, verbs, adjectives and adverbs grouped by semantic concepts, and uses a morphological function to infer the stem of an input word. Next, based on our manually curated dictionary of interaction keywords (Table 1), another dictionary of interaction keyword stems is created. The stem of each word in the parsed sentences is then searched against the dictionary of interactions keyword stems, thus reducing the search time. In the last example, there are two words whose stems are found among the HPI-specific keywords of the dictionary: 'delivered' (the stem is 'deliver') and 'targets' (the stem is 'target').

Role type matching: in this module, the role of each syntactic component (i.e. subject, verb, object and modifying phrase) of a simple sentence is determined. The role of a syntactic component specifies whether the component contains complete information about an HPI. In this work, three types of roles are considered: elementary, partial and complete. A component of elementary type can be a single host entity, a pathogen entity or an interaction keyword. A partial type component contains two distinct elementary type components. Finally, a syntactic component of complete type contains components of the three distinct elementary types: a host entity, a pathogen entity and an interaction keyword.

Table 2. Species hierarchy used in the language-based approach

Level 1	*Deuterostomia, Viridiplantae, Rhodophyta*
Level 2	*Acoelomata, Pseudocoelomata, Protostomia, Cnidaria, Ctenophora, Mesozoa, Placozoa, Porifera*
Level 3	*Bacteria, Choanoflagellida, Fungi, Fungi/Metazoa incertae sedis, Apusozoa, Centroheliozoa, Cryptophyta, Euglenozoa, Fornicata, Glaucocystophyceae, Haptophyceae, Heterolobosea, Jakobida, Katablepharidophyta, Malawimonadidae, Nucleariidae, Oxymonadida, Parabasalia, Rhizaria, stramenopiles*
Level 4	Viruses

In the language-based framework, an organism from a higher level can only be a host of an organism from a lower level.

Interaction extraction: next, each identified syntactic component is searched against a set of interaction patterns. First, components of the complete type are selected, since they contain complete information about a HPI occurring between two proteins/genes. Secondly, each of the remaining components of elementary or partial type is paired with one or several other components, until the set of paired components jointly contains complete information on HPI. An interaction pattern is defined in the following form: LS = RS. The left side (LS) is used to match the complete type from syntactic component(s), and the right side (RS) is used to extract the interaction information from each component. For example, a pattern

$$S<E>V<E>O<E>=P<S>I<V>H<O>$$

indicates that if a simple sentence includes three components, each of elementary type: (i) subject, (ii) verb and (iii) object, then the sentence contains (i) a pathogen entity in the subject; (ii) an interaction keyword in the verb; and (iii) a host entity in the object. In this work, the following seven patterns are considered: $S-V-O$, $S-O$, $S-V-M$, $S-M$, S, O *and* M (for abbreviations, see Section 2.3.7). We note that HPI information detected using the seven patterns may be incorrect, since these patterns are general and do not guarantee the semantic connection between the host entity, pathogen entity and interaction keyword. Therefore, in this module an additional heuristic template filter is used that is based on the empirical analysis of how the information about HPIs is usually expressed in a biomedical abstract. Currently, the filter includes three patterns, similar to the patterns employed by RelEx:

Pattern 1: A + interaction verb + B

Pattern 2: Interaction noun + 'between' + A + 'and' + B

Pattern 3: Interaction noun + 'of' + A + 'by' + B

where interaction verb is a interaction keyword in a verb form with or without prepositions, and interaction noun is an interaction keyword in the noun form. In addition to the pattern filter, a built-in hierarchy of species (Table 2) is used to correct the interactions containing swapped host and pathogen entities that were incorrectly assigned by the Entity Tagging module. The hierarchy is designed such that species in the higher hierarchical level cannot be assigned as a pathogen for the host species at the lower level. Our system builds the taxonomy of the detected host/pathogen entities on-the-fly and determines whether the host/pathogen entity assignment is correct by determining their hierarchy levels.

Uncertainty analysis: to determine whether an HPI-containing sentence carries uncertain information, the uncertainty analysis module scans the

sentence against two dictionaries: one containing negation keywords (e.g. 'does not', 'cannot') and another containing uncertainty keywords (e.g. 'possibly', 'may'). A negation/uncertainty keyword is considered to contribute to the HPI content if there is a link connecting the keyword with one of the three syntactic components.

Interaction normalization: often, the information about a specific HPI is presented in multiple sentences. The interaction normalization stage ensures that all sentences containing duplicate HPIs are detected. Two HPIs are defined as duplicate, if they contain the same quadruple of host and pathogen proteins/genes together with the names of their organisms. Since protein/gene and organism names are normalized into real entities during the Entity tagging stage, the module checks for duplication by referring to the real entity layer. Secondly, all certainty levels across different sentences describing the same HPI are summarized to find the final certainty level. For HPIs shared by multiple sentences, the following rule is applied: if there is a sentence containing negative information on an HPI, then the interaction is defined as negative; if there is a sentence about an HPI that was classified as uncertain, then HPI is the uncertain interaction; if none of the previous two condition are applied, then the HPI is certain.

2.3 Assessment

We next assess both protocols, comparing their performance with each other and with a naïve protocol that relies on a state-of-the-art literature mining method for extracting PPIs. Unfortunately, the only currently published HPI abstract classifier (Yin *et al*., 2010) was not available for the assessment.

Naïve protocol: in the naïve approach, an HPI-containing abstract is classified by (i) determining whether the abstract contains any PPIs; and (ii) determining whether it has at least one host and one pathogen organism. Our naïve protocol relies on the Protein Interaction information Extraction system (PIE) system, which utilizes the natural language processing and machine learning methods to determine specific sentences containing PPI (Kim *et al*., 2008). For each sentence, the protein names and interaction keywords are also extracted. The source organism for each interacting protein is then identified using NLProt (Mika and Rost, 2004). Finally, a PPI-containing sentence is considered to have HPI information if the two proteins come from two different organisms, one of which is classified as a host and another as a pathogen, according to our dictionary of host and pathogen names.

Assessment of approaches: the performance of all three approaches, including the naïve approach, on Task 1 is done using five basic measures. The accuracy, f_{AC}, is calculated as $f_{AC}=(N_{TP}+N_{TN})/N$, where N_{TP} and N_{TN} are the numbers of true positives and negatives, correspondingly, and N is the number of classified examples. The precision, f_{PR}, is calculated as $f_{PR}=N_{TP}/(N_{TP}+N_{FP})$, and the recall, f_{RE}, is calculated as $f_{RE}=N_{TP}/(N_{TP}+N_{FN})$, where N_{FP} and N_{FN} are the numbers of false positives and negatives, correspondingly. F-score is calculated as $F=2\frac{f_{PR}f_{RE}}{f_{PR}+f_{RE}}$. Finally, for the feature-based method, the area under receiver-operating characteristic curve (AUC) is calculated.

To assess the feature-based approach, three benchmarking protocols are used. For the first two protocols, the entire data set of 350 abstracts (see Section 3.1 for more detail) is randomly split into 262 (75%) abstracts for the training set and 88 (25%) abstracts for the testing set; each set consists of an equal fraction of positive and negative examples. In the first protocol, the SVM model training is done on the training set and the assessment is performed exclusively on the testing set (Supplementary Table S1). For the second protocol, a 10-fold cross-validation on the training set is used. Finally, in the third protocol, we explore whether increasing the training set will drastically influence the accuracy of the method. To do so, the training and testing sets are merged together and a leave-one-out cross-validation is performed on the entire set (we will refer to it as the expanded leave-one-out cross-validation). The language-based and naïve approaches are also evaluated on the testing set to compare their performance with the feature-based approach.

For Task 2, the predicted sentences are compared with the manually annotated sentences from the test set, and the overlap between the predicted and annotated sentences is calculated. Two types of HPI-relevant manually annotated sentences are defined. The first, complete, type contains the host and pathogen proteins/genes together with their organism names in one sentence. In the second, partial, type this information is split into multiple sentences. To fully evaluate each approach, two measures are introduced: (i) percentage of true positive sentences with reference to the total number of sentences extracted (prediction accuracy); (ii) percentage of the total number of predicted sentences with reference to the total number of positive sentences determined by manual curation (prediction coverage). Furthermore, each measure is calculated on the following four sets of sentences: (i) complete sentences detected in the HPI-relevant abstracts for which both organisms are extracted by the language-based approach; (ii) partial sentences (missing some HPI information) detected in the same abstracts; (iii) complete sentences detected in the HPI-relevant abstracts for which both proteins/genes are extracted by the language-based approach; and (iv) partial sentences detected in the same abstracts. In total, eight measures are calculated for each approach. Each measure is denoted as S_j^i, where j corresponds to one of the two measures and i corresponds to one of the four sets of sentences. For instance, S_2^B corresponds to the prediction coverage of the second set. Finally, we note that for this task and for Task 3, the micro-average protocol is employed (Krallinger *et al*., 2008): the average measures are calculated across all HPIs from the testing set, equally weighting the contribution of each interaction, even if an abstract contains multiple HPIs.

For Task 3, the performance of each approach is assessed based on the extracted information about the interacting host and pathogen proteins/genes and the names of their respective organisms in the HPI-relevant abstracts. Specifically, 12 different measures are calculated. The first six measures, h_{PR}^{PRT}, h_{RE}^{PRT}, h_{FS}^{PRT}, h_{PR}^{ORG}, h_{RE}^{ORG} and h_{FS}^{ORG} address the precision (PR), recall (RE) and F-score (FS), correspondingly, of detecting the pairs of interacting host and pathogen protein/genes as well as their organisms. The other six measures, g_{PR}^{PRT}, g_{RE}^{PRT}, g_{FS}^{PRT}, g_{PR}^{ORG}, g_{RE}^{ORG} and g_{FS}^{ORG} account for the partial detection of HPI information: they correspond to PR, RE and FS when at least one of the two proteins/genes or their organisms is detected.

3 RESULTS

3.1 Data collection

HPI data used for both training and testing sets were collected and manually curated from the MEDLINE/PubMed database (Supplementary Table S1). The data included 175 abstracts containing information about PPIs between 29 host and 77 pathogen organisms (positive set of examples) and 175 abstracts where no HPI information is found (negative set of examples). To obtain the positive data set, all abstracts containing the names of at least one host and one pathogen organism from our dictionary were identified. The names included generic keywords 'host' and 'pathogen', but excluded 'human' as a host organism. The search resulted in 88 abstracts. Next, an additional set of 87 human–pathogen interactions was added. These interactions were extracted from the pathogen interaction gateway (PIG), a database on human–pathogen interactions (Driscoll *et al*., 2009). Each abstract was then manually annotated, with the following information extracted from each positive example: host name, pathogen name, host protein/gene, pathogen protein/gene and certain/uncertain HPI, where the last annotation refers to the uncertainty of HPI information. Out of 46 members of the positive training set, eight were annotated as uncertain.

The negative data set was generated from the manual curation of a similar query as the one for the positive data set: human

Table 3. Evaluation of the feature-based classifier on Task 1

Protocol	f_{AC}	f_{PR}	f_{RE}	AUC	F-score
10-fold	72%	73%	71%	0.78	0.72
Test	66%	66%	65%	0.66	0.65
LOO	71%	72%	72%	0.78	0.71

10-fold refers to a 10-fold cross-validation protocol applied to the models that are trained on the set of 262 examples. Test refers to assessment on an independent test set. Finally, LOO denotes the expanded leave-one-out cross-validation protocol applied to the models trained on the entire data set of 352 examples. The assessment measures are shown for the radial kernel, which has the highest accuracy in the 10-fold cross validation, compared with to the other two kernels.

was included as a host organism together with others and the abstracts that were found to be HPI irrelevant were included to the negative set. The list of manually curated positive and negative sets of PubMed abstracts can be found at: http://korkinlab.org/datasets/philm/philm_data.html

3.2 Evaluation of feature-based, language-based and naïve approaches

The performances of the language-based and feature-based approaches were evaluated and compared with each other and with the naïve approach. For Task1, the naïve approach was applied to 88 abstracts from the test set, obtaining 53% accuracy, 100% precision and 6.8% recall, with an F-score of 0.13. We found that the method almost completely failed to detect the true positive abstracts; the contribution to the accuracy came primarily from the true negative hits, containing 44 (out of 44) abstracts from the negative testing set. The assessment of the language-based method on the test set resulted in 65% accuracy, 84% precision and 36% recall, with an F-score of 0.51. Given the NLProt annotation, the average running time of the system on a single abstract was 36.3 s on a 2.4 Ghz Intel workstation, where 99.95% of time was used by the link grammar parsing and Relex processing modules.

During 10-fold cross-validation of the feature-based method on Task 1, C and γ SVM parameters, as well as the polynomial degree in the polynomial kernel, were optimized. As a result, the most accurate SVM model was the model with the radial kernel of degree 3 and parameter values $C=2$ and $\gamma=0.001$ (Table 3), while the least accurate model, linear has only 2% worse accuracy. Overall, the performance of the three kernels during the leave-one-out and 10-fold cross-validation protocols was remarkably similar. The assessment results on the test set for the radial kernel were 4–11% worse, although the linear kernel model did significantly better than other kernels for this protocol (see Supplementary Table S1).

The SVM classifier worked significantly faster than the language-based approach: it took 0.003 s to classify 88 abstracts by an SVM classifier on a 2.66 Ghz Intel Xeon (Quad) workstation. Unfortunately, the high efficiency of this approach was offset by a significantly slower protein tagging stage by NLProt: it took ∼18 min on the same workstation to tag proteins in 262 abstracts from the data set. The protein tagging module was also among the main contributors to the running time of the language-based approach.

The assessment of all three approaches on Task 3 revealed that the language-based approach benefited significantly from the advanced processing of sentences when compared to the simple dictionary search approach employed in the feature-based and in the naïve approaches (Supplementary Table S2). For instance the precision, recall and F-score for extracting the interacting host and pathogen organisms were 36%, 18% and 0.24, correspondingly, whereas values of the same measures for the feature-based approach were 3%, 7% and 0.05, correspondingly. Moreover, the results showed that the naïve methods could not be used for Task 2: the values of the corresponding measures were all zeros. Thus, while the language-based approach was far from being a satisfactory method, it demonstrated perhaps the most promising direction for addressing Task 3.

The evaluation of the approaches on Task 2 also demonstrated a significant advantage of the language-based approach in pinpointing the HPI-relevant sentences (Supplementary Table S3). In particular, in the abstracts with the correctly identified pairs of proteins/genes the language-based method was able to determine complete sentences with 63% accuracy and 100% coverage, whereas the feature-based approach was less accurate (17%), while covering too many sentences (600%). Similar results were obtained for the abstracts where the host and pathogen organisms were correctly identified. However, the feature-based approach was able to accurately determine the sentences with the partial HPI information (Supplementary Table S3).

3.3 Case study summary

Finally, a detailed analysis of case studies on classifying PubMed abstracts and extracting HPI data (Supplementary Material) has revealed several common reasons behind the incorrect classification by each approach. Specifically, we found that incorrect protein tagging and species assignment were among the main reasons behind the failures of each approach, but not the only ones. Other important sources of errors were the link grammar parser and anaphora resolution method. For the feature-based approach, it was noticed that the information on the extracted organism names and on the extracted proteins/genes affected the classification of the HPI-relevant abstracts in a different way. Overall, the feature-based approach was less sensitive to the missing data than the language-based approach.

Both, feature-based and language-based, approaches were implemented in PHILM (Pathogen–Host Interaction Literature Mining) web server, accessible at http://korkinlab.org/philm.html. PHILM allows the user to classify an abstract and extract HPI information based on the abstract's text or its unique PubMed ID.

4 DISCUSSION

In this work, the problem of mining interactions occurring between proteins/genes of a pathogen organism and its host from the abstracts of biomedical publications was addressed. To do so, two new approaches were introduced: a feature-based approach, which relied on SVM methodology, and a language-based approach, which employed the link grammar.

Three basic HPI mining tasks were defined; for each task the approaches were evaluated and compared with each other. The performance of the two approaches was also compared with a naïve approach utilizing a publicly available state-of-the-art method for mining PPIs. The evaluation demonstrated that both approaches could mine the HPI information more accurately than the naïve

approach. In addition, the feature-based approach was found to be more accurate than the language-based approach when classifying the HPI-relevant abstracts, while the latter approach was more accurate when extracting the HPI information and pinpointing the HPI-relevant sentences. As expected, extracting the HPI information (Task 3) was by far the most difficult task, leaving room for further improvements. In contrast, identifying the HPI-relevant sentences (Task 2) could be accurately identified even with the current language-based approach. The results suggest that it would be possible to integrate both approaches into one system that will benefit from advantages of each approach.

With the rapid improvement of experimental methodology, we expect to see an increasing growth of the biomedical literature on HPIs in the next several years. Recently, several HPI databases were developed (Driscoll *et al*., 2009; Kumar and Nanduri, 2010) that collected and filtered the information on HPIs from the databases on general PPIs, such as IntAct (Aranda *et al*., 2010), DIP (Salwinski *et al*., 2004) and MINT (Ceol *et al*., 2010). The information provided by the HPI databases is valuable, since it could be an important source for training and testing sets used in computational methods. However, these databases rely on the PPI databases whose coverage is generally limited. For instance, out of 131 manually annotated HPI abstracts from the positive training set, 15 abstracts were found in MINT, eight abstracts in IntAct and only two abstracts in DIP. The most recent abstracts from the training set that were found in IntAct dated back as far as 2006. The importance of developing the HPI mining methods as a source of information complimentary to the existing integrative approaches was further emphasized by the obtained analysis.

A detailed analysis of case studies further supported our conclusion that increasing the accuracy of the name tagging system is critical to increasing the classification accuracy for both approaches, and is one of our next steps. Improving the link grammar parsing and anaphora resolution are the other future steps towards improving the accuracy of the language-based approach. Finally, we will integrate the feature-based and language-based approaches into a single system, and apply it to the task of finding the HPI-containing abstracts from the entire PubMed data set.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the helpful comments and suggestions from anonymous reviewers.

Funding: We acknowledge funding from University of Missouri (Mizzou Advantage to D.K.), National Science Foundation (DBI-0845196 to D.K.) and Department of Education (GAANN Fellowship to S.W.).

Conflict of Interest: none declared.

REFERENCES

Ahmed,S.T. *et al.* (2005) IntEx: a syntactic role driven protein-protein interaction extractor for bio-medical text. In: *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature. Ontologies and Databases: Mining Biological Semantics*. Association for Computational Linguistics, Detroit, pp. 54–61.

Anderson,R.M. and May,R.M. (1979) Population biology of infectious diseases: Part I. *Nature*, **280**, 361–367.

Aranda,B. *et al.* (2010) The IntAct molecular interaction database in 2010. *Nucleic Acids Res.*, **38**, D525–D531.

Bairoch,A. *et al.* (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Res.*, **33**, D154–D159.

Blaschke,C. and Valencia,A. (2001) The potential use of SUISEKI as a protein interaction discovery tool. *Genome Inform.*, 123–134.

Boeckmann,B. *et al.* (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, **31**, 365.

Ceol,A. *et al.* (2010) MINT, the molecular interaction database: 2009 update. *Nucleic Acids Res.*, **38**, D532–D539.

Chagoyen,M. *et al.* (2006) Discovering semantic features in the literature: a foundation for building functional associations. *BMC Bioinformatics*, **7**, 41.

Corney,D.P. *et al.* (2004) BioRAT: extracting biological information from full-length papers. *Bioinformatics*, **20**, 3206–3213.

Davis,F.P. *et al.* (2007) Host pathogen protein interactions predicted by comparative modeling. *Protein Sci.*, **16**, 2585–2596.

Donaldson,I. *et al.* (2003) PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, **4**, 11.

Driscoll,T. *et al.* (2009) PIG–the pathogen interaction gateway. *Nucleic Acids Res.*, **37**, D647–D650.

Dyer,M.D. *et al.* (2007) Computational prediction of host-pathogen protein-protein interactions. *Bioinformatics*, **23**, i159–i166.

Dyer,M.D. *et al.* (2010) The human-bacterial pathogen protein interaction networks of Bacillus anthracis, Francisella tularensis, and Yersinia pestis. *PLoS ONE*, **5**, e12089.

Fellbaum,C. (1998) *WordNet : an Electronic Lexical Database*. Language, speech, and communication. MIT Press, Cambridge, USA.

Franzosa,E.A. and Xia,Y. (2011) Structural principles within the human-virus protein-protein interaction network. *Proc. Natl Acad. Sci.*., **108**: 10538–10543.

Friedman,C. *et al.* (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, **17**, S74.

Fundel,K. *et al*. (2007) RelEx–relation extraction using dependency parse trees. *Bioinformatics*, **23**, 365.

Hao,Y. *et al.* (2005) Discovering patterns to extract protein-protein interactions from the literature: Part II. *Bioinformatics*, **21**, 3294–3300.

Hirschman,L. *et al.* (2005) Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, **6** (Suppl 1), S1.

Hobbs,J. (1978) Resolving pronoun references. *Lingua*, **44**, 311–338.

Hoffmann,R. *et al.* (2005) Text mining for metabolic pathways, signaling cascades, and protein networks. *Sci. STKE*, **2005**, pe21.

Hoffmann,R. and Valencia,A. (2005) Implementing the iHOP concept for navigation of biomedical literature. *Bioinformatics*, **21** (Suppl 2), ii252–ii258.

Hu,Z. *et al.* (2005) Literature mining and database annotation of protein phosphorylation using a rule-based system. *Bioinformatics*, **21**, 2759.

Huang,M. *et al.* (2008) Mining physical protein-protein interactions from the literature. *Genome Biol.*, **9**, S12.

Kim,S. *et al.* (2008) PIE: an online prediction system for protein-protein interactions from text. *Nucleic Acids Res.*, **36**, W411.

Konig,R. *et al.* (2008) Global analysis of host-pathogen interactions that regulate early-stage HIV-1 replication. *Cell*, **135**, 49–60.

Krallinger,M. *et al.* (2008) Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biol.*, **9** (Suppl 2), S4.

Krallinger,M. *et al.* (2008) Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge. *Genome Biol.*, **9** (Suppl 2), S1.

Krallinger,M. and Valencia,A. (2005) Text-mining and information-retrieval services for molecular biology. *Genome Biol.*, **6**, 224.

Kumar,R. and Nanduri,B. (2010) HPIDB-a unified resource for host-pathogen interactions. *BMC Bioinformatics.* **11**, S16.

Lee,H. *et al.* (2008) E3Miner: a text mining tool for ubiquitin-protein ligases. *Nucleic Acids Res.*, **36**, W416.

Leroy,G. and Chen,H. (2002) Filling preposition-based templates to capture information from medical abstracts. *Pac. Symp. Biocomput.*, **2002**, 350–361.

Mandell,G.L. and Townsend,G.C. (1998) New and emerging infectious diseases. *Trans. Am. Clin. Climatol. Assoc.*, **109**, 205–216; discussion 216–207.

Marcotte,E.M. *et al*. (2001) Mining literature for protein-protein interactions. *Bioinformatics*, **17**, 359–363.

Mika,S. and Rost,B. (2004) Protein names precisely peeled off free text. *Bioinformatics*, **20**, i241.

Pyysalo,S. *et al.* (2004) Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions. In: *International Workshop on Natural Language Processing in Biomedicine and its Applications* (*JNLPBA*). Association for Computational Linguistics, pp. 15–21.

Pyysalo,S. *et al.* (2006) Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, **7**, S2.

Rodriguez-Esteban,R. (2009) Biomedical text mining and its applications. *PLoS Comput. Biol.*, **5**, e1000597.

Salwinski,L. *et al.* (2004) The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res.*, **32**, D449–D451.

Santos,C. and Eggle,D. (2005) Wnt pathway curation using automated natural language processing: combining statistical methods with partial and full parse for knowledge extraction. *Bioinformatics*, **21**, 1653.

Seki,K. and Mostafa,J. (2005) A hybrid approach to protein name identification in biomedical texts. *Inform. Process. Manag.*, **41**, 723–743.

Shatkay,H. *et al.* (2007) SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics*, **23**, 1410.

Sleator,D. and Temperley,D. (1995) Parsing English with a Link Grammar. In: *Third International Workshop on Parsing Technologies*. ACL/SIGPARSE, p. 91.

Stephens,M. *et al.* (2001) Detecting gene relations from Medline abstracts. *Pac. Symp. Biocomput.*, **2001**, 483–495.

Tanabe,L. *et al.* (2005) GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, **6**, S3.

Vapnik,V.N. (1998) Statistical learning theory. In *Adaptive and Learning Systems for Signal Processing, Communications, and Control*. Wiley, New York, NY, USA.

Wheeler,D. *et al.* (2006) Database resources of the National Center for Biotechnology information. *Nucleic Acids Res.*, **34**, D173–D180.

Winnenburg,R. *et al.* (2008) PHI-base update: additions to the pathogen host interaction database. *Nucleic Acids Res.*, **36**, D572.

Yang,Z. *et al.* (2009) BioPPIExtractor: a protein-protein interaction extraction system for biomedical literature. *Expert Syst. Appl.*, **36**, 2228–2233.

Yin,L. *et al.* (2010) Document classification for mining host pathogen protein-protein interactions. *Artif. Intell. Med.*, **49**, 155–160.