Badal *et al. BMC Bioinformatics* (2018) 19:84
https://doi.org/10.1186/s12859-018-2079-4

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

# Natural language processing in text mining for structural modeling of protein complexes

Varsha D. Badal, Petras J. Kundrotas* and Ilya A. Vakser*

## Abstract

**Background:** Structural modeling of protein-protein interactions produces a large number of putative configurations of the protein complexes. Identification of the near-native models among them is a serious challenge. Publicly available results of biomedical research may provide constraints on the binding mode, which can be essential for the docking. Our text-mining (TM) tool, which extracts binding site residues from the PubMed abstracts, was successfully applied to protein docking (Badal et al., PLoS Comput Biol, 2015; 11: e1004630). Still, many extracted residues were not relevant to the docking.

**Results:** We present an extension of the TM tool, which utilizes natural language processing (NLP) for analyzing the context of the residue occurrence. The procedure was tested using generic and specialized dictionaries. The results showed that the keyword dictionaries designed for identification of protein interactions are not adequate for the TM prediction of the binding mode. However, our dictionary designed to distinguish keywords relevant to the protein binding sites led to considerable improvement in the TM performance. We investigated the utility of several methods of context analysis, based on dissection of the sentence parse trees. The machine learning-based NLP filtered the pool of the mined residues significantly more efficiently than the rule-based NLP. Constraints generated by NLP were tested in docking of unbound proteins from the DOCKGROUND X-ray benchmark set 4. The output of the global low-resolution docking scan was post-processed, separately, by constraints from the basic TM, constraints re-ranked by NLP, and the reference constraints. The quality of a match was assessed by the interface root-mean-square deviation. The results showed significant improvement of the docking output when using the constraints generated by the advanced TM with NLP.

**Conclusions:** The basic TM procedure for extracting protein-protein binding site residues from the PubMed abstracts was significantly advanced by the deep parsing (NLP techniques for contextual analysis) in purging of the initial pool of the extracted residues. Benchmarking showed a substantial increase of the docking success rate based on the constraints generated by the advanced TM with NLP.

**Keywords:** Protein interactions, Binding site prediction, Protein docking, Dependency parser, Rule-based system, Supervised learning

## Background

Protein-protein interactions (PPI) play a key role in various biological processes. An adequate characterization of the molecular mechanisms of these processes requires 3D structures of the protein-protein complexes. Due to the limitations of the experimental techniques, most structures have to be modeled by either free or template-based docking [1]. Both docking paradigms produce a large pool of putative models, and selecting the correct one is a non-trivial task, performed by scoring procedures [2]. Often knowledge of a few binding site residues is enough for successful docking [3].

In recent years, the number of biomedical publications, including PPI-relevant fields, has been growing fast [4]. Thus, automated text mining (TM) tools utilizing online availability of indexed scientific literature (e.g. PubMed https://www.ncbi.nlm.nih.gov/ pubmed) are becoming increasingly important, employing Natural Language Processing (NLP) algorithms to purge non-relevant information from the initial

* Correspondence: pkundro@ku.edu; vakser@ku.edu
Center for Computational Biology and Department of Molecular Biosciences, The University of Kansas, Lawrence, Kansas 66047, USA

pool of extracted knowledge. TM + NLP techniques are widely used in biological text mining [5–18], particularly for the extraction and analysis of information on PPI networks [19–34] and for the prediction of small molecules binding sites [35, 36].

Recently, we developed a basic TM tool that extracts information on protein binding site residues from the PubMed abstracts. The docking success rate significantly increased when the mined residues were used as constraints [37]. However, the results also showed that many residues mentioned in the abstracts are not relevant to the protein binding. Examples of such residues include those originating from studies of small molecule binding, or from papers on stability of the individual proteins. Filtering the extracted residues by the shallow parsing (bag-of-words) Support Vector Machines (SVM) was shown to be insufficient. In this paper, we present an advancement of our basic TM procedure based on the deep parsing (NLP techniques for contextual analysis of the abstract sentences) for purging of the initial pool of the extracted residues.

## Methods

### Outline of the text-mining protocol

The TM procedure was tested on 579 protein-protein complexes (bound X-ray structures purged at 30% sequence identity level) from the DOCKGROUND resource (http://dockground.compbio.ku.edu) [38]. The basic stage of the procedure consists of two major steps: information retrieval and information extraction [37] (Fig. 1). The abstracts are retrieved from PubMed using NCBI E-utilities tool (http://www.ncbi.nlm.nih. gov/books/NBK25501) requiring that either the names of both proteins (AND-query) or the name of one protein in a complex (OR-query) are present in the abstract. The text of the retrieved abstracts is then processed for the residue names. The structures of the individual proteins are used to filter the pool of the extracted residues by: (*i*) correspondence of the name and the number of the extracted residues to those in the Protein Data Bank (PDB) file, and (*ii*) presence of the extracted residue on the surface of the protein. Several NLP-based approaches (semantic similarity to generic and specialized keywords, parse tree analysis with or without SVM enhancement) were further applied for additional filtering of the extracted residues from the abstracts retrieved by the OR-queries. Performance of the TM protocol for a particular PPI, for which $N$ residue-containing abstracts were retrieved, is evaluated as

$$P_{TM} = \frac{\sum_{i=1}^{N} N_i^{\text{int}}}{\sum_{i=1}^{N} \left(N_i^{\text{int}} + N_i^{\text{non}}\right)}, \tag{1}$$

where $N_i^{\text{int}}$ and $N_i^{\text{non}}$ are the number of the interface and the non-interface residues, correspondingly, mentioned in abstract $i$ for this PPI, not filtered out by a specific algorithm (if all residues in an abstract are purged, then this abstract is excluded from the $P_{\text{TM}}$ calculations). It is

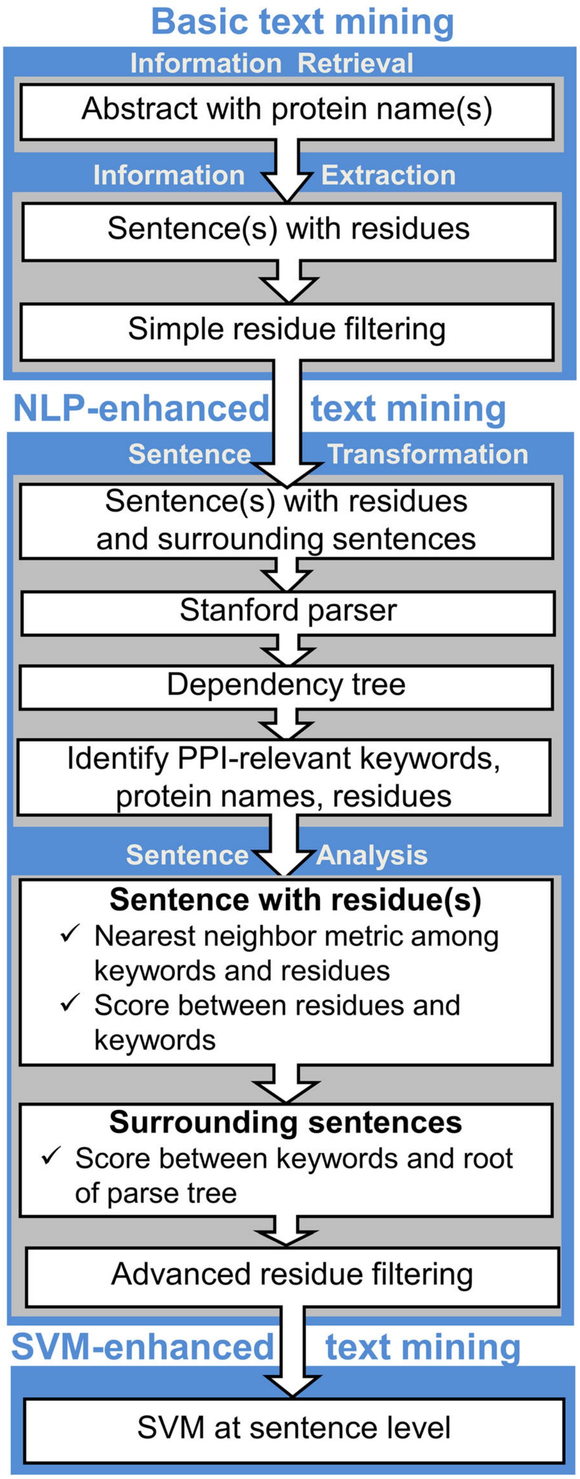
Basic text mining
Information Retrieval
Abstract with protein name(s)
Information Extraction
Sentence(s) with residues
Simple residue filtering
NLP-enhanced text mining
Sentence Transformation
Sentence(s) with residues and surrounding sentences
Stanford parser
Dependency tree
Identify PPI-relevant keywords, protein names, residues
Sentence Analysis
Sentence with residue(s)
✓ Nearest neighbor metric among keywords and residues
✓ Score between residues and keywords
Surrounding sentences
✓ Score between keywords and root of parse tree
Advanced residue filtering
SVM-enhanced text mining
SVM at sentence level

**Fig. 1** Flowchart of NLP-enhanced text mining system. Scoring of surrounding sentences is shown for Method 3 (see text)

convenient to compare the performance of two algorithms for residue filtering in terms of

$$\Delta N(P_{TM}) \quad = N_{tar}^{X_1}(P_{TM}) - N_{tar}^{X_2}(P_{TM}), \tag{2}$$

where $N_{tar}^{X_1}(P_{TM})$ and $N_{tar}^{X_2}(P_{TM})$ are the number of targets with $P_{\mathrm{TM}}$ value yielded by algorithms $X_1$ and $X_2$, respectively. The $N(0)$ and $N(1)$ values capture the general shape of the $P_{\mathrm{TM}}$ distribution. Thus, the effectiveness of an algorithm can be judged by its ability to reduce $N(0)$ (all false positives) and increase $N(1)$ (all true positives). In this study, advanced residue filtering algorithms are applied to the pool of residues extracted by the OR-queries with the basic residue filtering, thus $X_2$ will hereafter refer to this algorithm. The negative values of $\Delta N(0)$ and the positive values of $\Delta N(1)$ indicate successful purging of irrelevant residues from the mined abstracts.

### Selection of keywords

Generic keywords semantically closest to PPI-specific concept keywords (see Results) were found using Perl module QueryData.pm. The other Perl modules lesk.pm, lin.pm and path.pm were used to calculate similarity scores introduced by Lesk [39, 40], Lin [41] and Path [42, 43], correspondingly, between the token (words) in a residue-containing sentence and the generic keywords. These Perl modules, provided by the WordNet [44, 45] (http://wordnet.princeton.edu), were downloaded from http://search.cpan.org. The score thresholds for the residue filtering were set as 20, 0.2, and 0.11, for the Lesk, Lin and Path scores, respectively.

The keywords relevant to the PPI binding site (PPI + ive words), and the keywords that may represent the fact of interaction only (PPI-ive words) (Table 3) were selected from manual analysis of the parse trees for 500 sentences from 208 abstracts on studies of 32 protein complexes.

### Scoring of residue-containing and context sentences

The parse tree of a sentence was built by the Perl module of the Stanford parser [46, 47] (http://nlp.stanford.edu/software/index.shtml) downloaded from http://search.cpan.org. The score of a residue in the sentence was calculated as

$$S_{\mathrm{X}} \quad = \sum_i \frac{1}{d_{\mathrm{X}i}^{+}} - \sum_j \frac{1}{d_{\mathrm{X}j}^{-}}, \tag{3}$$

where $d_{\mathrm{X}i}^{+}$ and $d_{\mathrm{X}j}^{-}$ are parse-tree distances between a residue and PPI + ive word $i$ and PPI-ive word $j$ in that sentence, respectively. Distances were calculated by edge counting in the parse tree. An example of a parse tree of residue-containing sentence with two interface residues having score 0.7 is shown in Additional file 1: (Figure S1).

An add-on value to the main $S_{\mathrm{X}}$ score (Eq. 3) from the context sentences (sentences immediately preceding and following the residue-containing sentence) was calculated either as simple presence or absence of keywords in these sentences, or as a score, similar to the $S_{\mathrm{X}}$ score, but between the keywords and the root of the sentence on the parse tree.

### SVM model

The features vector for the SVM model was constructed from the $S_{\mathrm{X}}$ score(s) of the residue-containing sentence and the keyword scores of the context sentences (see above). In addition, the scores accounting for the presence of protein names in the sentence

$$S_{\mathrm{prot}} \quad = \begin{cases} 0, \text{if no protein names in the sentence} \\ 1, \text{if only name of one protein in the sentence} \\ 2, \text{if name of both proteins in the sentence} \end{cases} \tag{4}$$

were also included, separately for the residue-containing, preceding, and following sentences. The SVM model was trained and validated (in 50/50 random split) on a subset of 1921 positive (with the interface residue) and 3865 negative (non-interface residue only) sentences using program SVMLight with linear, polynomial and RBF kernels [48–50]. The sentences were chosen in the order of abstract appearance in the TM results.

The SVM performance was evaluated in usual terms of precision $P$, recall $R$, accuracy $A$, and $F$-score [51].

$$\begin{aligned} P \quad &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad R \quad = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ A \quad &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FN}+\mathrm{TN}+\mathrm{FP}}, \quad F \quad = 2\frac{P \times R}{P+R}, \end{aligned} \tag{5}$$

where TP, FP, TN, and FN are, correspondingly, the number of correctly identified interface residues, incorrectly identified interface residues, correctly identified non-interface residues, and incorrectly identified non-interface residues in the validation set. The results (Additional file 1: Figure S2-S7) showed that the best performance was achieved using RBF kernel with gamma 16. Thus, this model was incorporated in the TM protocol (Fig. 1).

### Text mining constraints in docking protocol

TM constraints were incorporated in the docking protocol and the docking success rates assessed by benchmarking. Basic TM tool [37] with OR-queries was used to mine residues for 395 complexes from the Dockground unbound benchmark set 4. The set consists of the unbound crystallographically determined protein structures and corresponding co-crystallized complexes (bound structures). Binary combinations of OR and AND queries were generated [37]. The original publication on the crystallographically determined complex was left out, according to PMID in the PDB file. Because of the frequent discrepancy in the residue numbering and the chain IDs in the bound and the unbound structures, the residues were matched to the ones in the bound protein. The residues were ranked for each interacting protein using a confidence score. The confidence range was between 1 (low) and 10 (high). The AND-query residues were given preference over the OR-query ones for the basic TM protocol, according to our ranking scheme [37]. The confidence score was calculated as

$$f(R) = \min\left(10, \sum_{i=1}^{N_R} a_i\right), \tag{6}$$

where $N_R$ is the number of abstracts, mentioning residue $R$, $a_i = 1$, if abstract $i$ was retrieved by the OR-query only, and $a_i = 2$, if the abstract was retrieved by the AND-query. For each protein, the top five residues were used as constraints in GRAMM docking [52]. The constraints were utilized by adding an extra weight to the docking score if the identified residue was at the predicted interface. The maximum value of 10 reflects the difference between the low confidence ($f = 1$) and the high confidence ($f = 10$) constraints, while alleviating the effect of possible residue overrepresentation in published abstracts (very high $f$ values).

For the NLP score, the confidence ranking scheme was modified such that the range is preserved between 1 and 10 and the AND-query residues are given higher precedence than the OR-query residues. The NLP was used for re-ranking within each category as

$$f(R) = \left\{\begin{array}{l} 10,\ if\ \ for\ \ some\ \ i,\ \ a_i\ \ is\ \ retrieved\ \ in\ \ AND\ \ query\ \ and\ \ passes\ \ NLP \\ 8,\ \ if\ \ a_i\ \ is\ \ retrieved\ \ in\ \ AND\ \ query \\ 6,\ \ if\ \ any\ \ a_i\ \ retrieved\ \ in\ \ OR\ \ query\ \ passes\ \ NLP \\ \ \max(5,\ \ count\ \ of\ \ abstracts\ \ containing\ \ R) \end{array}\right\}, \tag{7}$$

The residues at the co-crystallized interface were used as reference. Such residues were determined by 6 Å atom-atom distance across the interface. The reference residue pairs were ranked according to the $C^{\alpha}$ - $C^{\alpha}$ distance. The top three residue-residue pairs were used in docking with the highest confidence score 10, to determine the maximum possible success rate for the protein set.

## Results and discussion

### Generic and specialized dictionaries

The simplest approach to examining the context of a residue mentioned in the abstracts would be to access the semantic similarity of words (token) in the residue-containing sentence to a generic but at the same time PPI-relevant concept. For the purpose of this study, such concept was chosen to be "binding site" as the one describing the physical contact between the two entities (proteins). We designated the words "touch" and "site" as the most semantically similar words relevant to this concept (binding site) to be used in WordNet [44, 45] (generic English lexical database with words grouped into sets of cognitive synonyms), which does not contain any knowledge-domain specific vocabularies [53]. Thus, we calculated similarity scores (see Methods) between these two words and all the words of the residue-containing sentence(s) in the abstracts retrieved by the OR-query. If a score exceeded a certain threshold, all residues in the sentence were considered to be the interface ones. Otherwise they were removed from the pool of the mined residues. The calculations were performed using three different algorithms for the similarity score. Similarity scores by Lesk and Path demonstrated only marginal improvement in the filtering of mined residues compared to the basic residue filtering (Table 1 and Fig. 2). Lin's score yielded considerably worse performance. Similarly poor performance of this score was reported previously, when it was applied to word prediction for nouns, verbs and across parts of speech [54]. In our opinion, this may be due to some degree of arbitrariness in the way the similar words are grouped under a common subsumer (most specific ancestor node), and how this subsumer fits into the overall hierarchy within the synset (set of cognitive synonyms). Thus, we concluded that generic vocabularies cannot be employed in the TM protocols for identifying PPI binding sites. This correlates with the conclusions of Sanchez et al. [55] that hierarchical structure of generic and

**Table 1** Overall text-mining performance with the residue filtering using semantic similarity of words in a residue-containing sentence to a generic concept in the WordNet vocabulary. For comparison, the results with basic residue filtering are also shown

| Query | Similarity measure | $L_{tot}^{a}$ | $L_{int}^{b}$ | Coverage (%)[c] | Success (%)[d] | Accuracy (%)[e] | $\Delta N(0)^{f}$ | $\Delta N(1)^{f}$ |
|---|---|---|---|---|---|---|---|---|
| AND | – | 128 | 108 | 22.1 | 18.7 | 84.4 | | |
| OR | – | 328 | 273 | 56.6 | 47.2 | 83.2 | | |
| OR | Lesk [39, 40] | 319 | 267 | 55.1 | 46.1 | 83.7 | -3 | −1 |
| OR | Lin [41] | 251 | 184 | 43.4 | 31.8 | 73.3 | + 8 | −8 |
| OR | Path [42, 43] | 316 | 265 | 54.6 | 45.8 | 83.9 | −3 | + 1 |

[a] Number of complexes for which TM protocol found at least one abstract with residues
[b] Number of complexes with at least one interface residue found in abstracts
[c] Ratio of $\mathbf{L_{tot}}$ and total number of complexes
[d] Ratio of $\mathbf{L_{int}}$ and total number of complexes
[e] Ratio of $\mathbf{L_{int}}$ and $\mathbf{L_{tot}}$
[f] Calculated by Eq. (2)

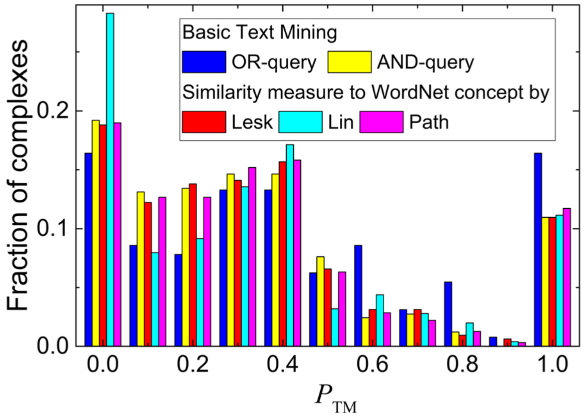

**Fig. 2** Performance of basic and advanced text mining protocols. Advanced filtering of the residues in the abstracts retrieved by the OR-queries was performed by calculating various similarity scores (see legend) between the words of residue-containing sentences and generic concept words from WordNet. The TM performance is calculated using Eq. (1). The distribution is normalized to the total number of complexes for which residues were extracted (third column in Table 1)

domain-specific vocabularies are different and thus, for example, MESH specific vocabulary [56] provides more accurate knowledge representation of medical concepts compared to the generic WordNet lexicon.

Next, we tested applicability of the 7 specialized dictionaries (Table 2) to filtering of the residues mined by the OR-queries. All these dictionaries were specifically designed for the mining of the literature on PPI identification and contain up to several hundred PPI-relevant keywords. Thus, there is no need to measure semantic similarity between words in the residue-containing sentence and words in these dictionaries, and it is just enough to spot these words in the sentences (maximum possible semantic similarity). If any keyword was spotted in a sentence, all residues mentioned in this sentence were considered as interface residues. The results (Table 2 and Fig. 3) indicated, however, that using all dictionaries did not yield significant improvement in the residue filtering. While some dictionaries (with $\Delta N(0) < 0$ in Table 2) succeeded in removing irrelevant information, there is a general tendency of removing relevant information as well (predominantly negative numbers of $\Delta N(1)$ in Table 2). Interestingly, the best performing dictionary by Schuhmann et al. [57] contains the smallest number of words.

All tested dictionaries were designed for the mining information on the existence of interaction. Thus, we also tested our own dictionary, designed specifically to distinguish keywords relevant and irrelevant to the protein-protein binding sites (see Methods). Despite the small amount of PPI-relevant words in the dictionary, the filtering of the mined residues based on this dictionary led to considerable improvement in the TM performance (the rightmost bars in Fig. 3 and the bottom row in Table 2). This suggests that even a limited amount of text provided by abstracts can be used to extract reliable PPI-relevant keywords.

### Analysis of sentence parse tree - deep parsing

In the dictionary look-up approach all residues in the sentence were treated either as interface or non-interface ones. The parse tree (hierarchical syntactic structure) of a sentence enables treating residues in the sentence differently depending on a local grammatical structure. Also, two adjacent words in a sentence can be far apart on the parse tree, and vice versa (distant words in a sentence can be close on the parse tree). This mitigates fluctuations in distances between keywords in “raw” sentences, caused by peculiarities in author’s writing style (some authors favor writing short concise sentences whereas others prefer long convoluted sentences). We adopted a simple approach based on the proximity of mined residue(s) to the PPI + ive and PPI-ive keywords (Table 3) on the parse tree, quantified in the score $S_X$ calculated by Eq. 3 the close proximity (in the grammatical sense) to the PPI + ive. The high positive value of the score implies that a residue is in keywords, making it plausible to suggest that this residue

**Table 2** Overall text-mining performance with the residue filtering based on spotting in the residue-containing sentences keyword(s) from specialized dictionaries

| Dictionary and reference | Number of PPI keywords | $L_{\mathrm{tot}}^{a}$ | $L_{\mathrm{int}}^{b}$ | Coverage (%)[c] | Success (%)[d] | Accuracy (%)[e] | $\Delta N(0)$[f] | $\Delta N(1)$[f] |
|---|---|---|---|---|---|---|---|---|
| Blaschke et al., [20] | 43 | 265 | 205 | 45.8 | 35.4 | 77.4 | 0 | −8 |
| Chowdhary et al., [58] | 191 | 284 | 233 | 49.1 | 40.2 | 82.0 | −7 | −4 |
| Hakenberg et al. [59] | 234 | 297 | 232 | 51.3 | 40.1 | 78.1 | 6 | −7 |
| Plake et al. [60] | 73 | 291 | 230 | 50.3 | 39.7 | 79.0 | 1 | −1 |
| Raja et al. [23] | 412 | 302 | 247 | 52.2 | 42.7 | 81.8 | 0 | −5 |
| Schuhmann et al. [57] | 64 | 212 | 152 | 36.6 | 26.3 | 71.7 | − 1 | 5 |
| Temkin et al. [21] | 174 | 283 | 223 | 48.9 | 38.5 | 78.8 | 0 | −9 |
| Own dictionary | 16 | 224 | 169 | 38.7 | 29.2 | 75.4 | −6 | 8 |

For definitions of columns 3–9, see footnotes to Table 1. Full content of in-house dictionary is in Table 3, but only PPI + ive part was used to calculate the data in this Table

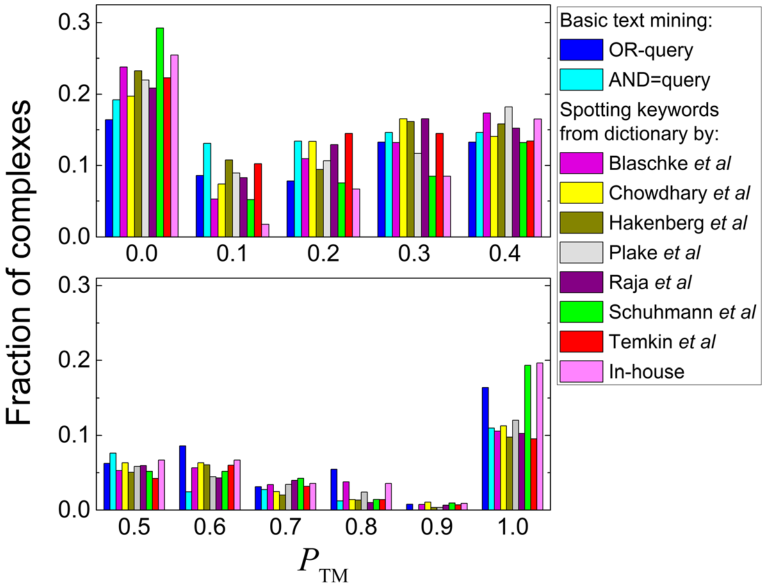

**Fig. 3** Performance of basic and advanced text mining protocols. Advanced filtering of the residues in the abstracts retrieved by the OR- queries was performed by spotting PPI-relevant keywords from various specialized dictionaries (see legend). The TM performance is calculated using Eq. (1). The distribution is normalized to the total number of complexes for which residues were extracted (third column in Table 2). Full content of the in-house dictionary is in Table 3, but only PPI + ive part was used to obtain results presented in this Figure. The data are shown in two panels for clarity

is related to the protein-protein binding site. Large negative $S_X$ values indicate closeness of the residue to the PPI-ive keywords, thus such residue is most likely outside the PPI interface. Note, that this approach is susceptible to quality and extent of the dictionary used. However, this problem will be mitigated as more relevant texts (including full-text articles) will be analyzed for finding new PPI + ive and PPI-ive keywords.

The interface residues tend to have $S_X > 0.25$ (Additional file 1: Figure S8). Thus, we used this value as a threshold to distinguish between interface and non-interface residues. Compared to the simple dictionary look-up (see above), even such simplified analysis of the parse tree, yielded significant improvement in the performance of our text-mining protocol (Method 1 in Table 4 and red bars in Fig. 4).

**Table 3** Manually generated dictionary used to distinguish relevant (PPI + ive) and irrelevant (PPI-ive) information on protein-protein binding sites. Only lemmas (stem words) are shown

| Category | Words |
|---|---|
| PPI + ive | bind, interfac, complex, hydrophob, recept, ligand, contact, recog, dock, groove, pocket, pouch, interact, crystal, latch, catal |
| PPi–ive | deamidation, IgM, IgG, dissociat, antibo, alloster, phosphory, nucleotide, polar, dCTP, dATP, dTTP, dUTP, dGTP, IgG1, IgG2, IgG3, IgG4, Fc, ubiquitin, neddylat, sumoyla, glycosylation, lipidation, carbonylation, nitrosylation, epitope, paratope, purine, pyrimidine, isomeriz, non-conserved, fucosylated, nonfucosylated, sialylation, galactosylation |

The main message of a sentence can propagate through the article text comprising several sentences around the master sentence (context) and therefore it would be logical to include context information in the residue filtering as well. However, there is no clear understanding how far away the message can spread, especially in such dense text as an abstract. Thus, we treated as context only sentences immediately preceding and following the residue-containing sentence. These sentences usually do not contain residues. Thus, we included context information either by simple spotting PPI + ive keywords in these sentences (Method 2) or by calculating $S_X$-like score of PPI + ive and

**Table 4** Overall text-mining performance with the residue filtering based on analysis of sentence parse tree

| Method of parse tree analysis | $L_{\text{tot}}$ | $L_{\text{int}}$ | Coverage (%) | Success (%) | Accuracy (%) | Δ*N*(0) | Δ*N*(1) |
|---|---|---|---|---|---|---|---|
| Method 1. Scoring of the residue-containing sentence only | 222 | 173 | 38.3 | 29.9 | 77.9 | −13 | + 10 |
| Method 2. Scoring of the residue-containing sentence and keyword spotting in the context sentences | 208 | 154 | 35.9 | 26.6 | 74.0 | −7 | + 3 |
| Method 3. SVM model with scores of the residue-containing and context sentences | 182 | 146 | 31.4 | 25.2 | 80.2 | −27 | + 21 |

Keywords used in the analysis were taken from our dictionary (Table 3). For definitions of columns 2–8, see footnotes to Table 1

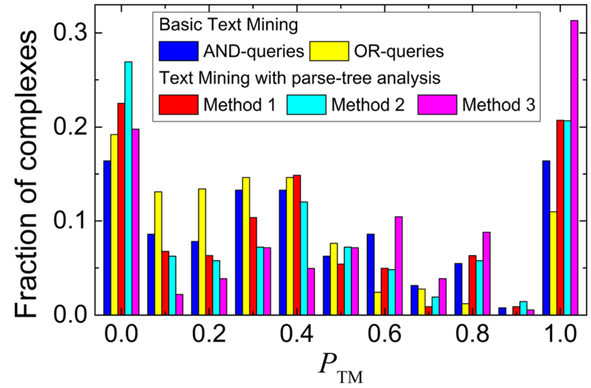

**Fig. 4** Performance of basic and advanced text mining protocols. Advanced filtering of the residues in the abstracts retrieved by the OR-queries was performed by different methods of analysis of the sentence parse trees (for method description see first column in Table 4) The TM performance was calculated using Eq. (1). The distribution is normalized to the total number of complexes for which residues were extracted (second column in Table 4)

PPI-ive words with respect to the sentence root (Method 3). In the former algorithm, a mined residue is treated as interface residues if its $S_X > 0.25$ and a PPI + ive keyword was spotted in the context sentences. The latter algorithm requires a more complicated approach as there is no clear distinction between the context-sentence scores for interface and non-interface residues. Thus, classification of the residues was performed by an SVM model with the optimal parameters (see Methods).

Inclusion of the context information by simple keyword spotting worsens the performance of the residue filtering (Method 2 in Table 4 and cyan bars in Fig. 4) as many interface residues are erroneously classified due to the absence of the keywords in the context sentences. Application of the SVM model, despite a relatively small number of its features, increased filtering performance dramatically, making SVM-based approach superior to all other methods investigated in this study. All three methods have comparable values of overall success and accuracy (Table 4). An example of successful filtering of non-interface residues is shown in Fig. 5 for the chains A and B of 2uyz. Out of five residues mined by the basic TM protocol, only one residue (Fig. 5, Glu67B) was at the complex interface ($P_{TM} = 0.20$). SVM model has filtered out all four non-interface residues, elevating TM performance to $P_{TM} = 1.00$ (details are available in Additional file 1: Table S1 and accompanying text).

Finally, to ensure that the results are not determined by over fitting the SVM model, we filtered residues on a reduced set of abstracts where all abstracts for a complex were excluded from the consideration if at least one abstract contained sentence(s) used for the training of the SVM model. Despite a significant drop in the coverage, the results on the reduced set (Additional file 1: Figure S9) did not differ much from the results obtained on the full set of abstracts.

## Docking using text-mining constraints

Constraints generated by NLP were tested in docking by GRAMM to model complexes of unbound proteins from the DOCKGROUND X-ray benchmark set 4 (see Methods). The set consists of 395 pairs of separately resolved unbound protein structures and their co-crystallized complexes. Each unbound complex was docked by GRAMM three times, using (1) constraints from the basic TM, (2) constraints re-ranked

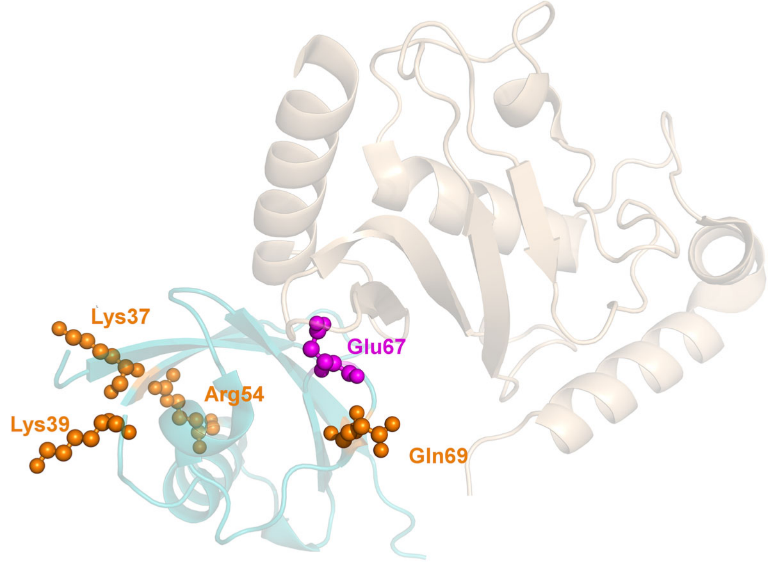

**Fig. 5** Successful filtering of mined residues by the SVM-based approach of the parse-tree analysis (Method 3 in Table 4). The structure is 2uyz chains A (wheat) and B (cyan). Residues mined by the basic TM protocol are highlighted. The ones filtered out by the advanced TM protocol are in orange

by NLP, and (3) the reference constraints. The output of the global low-resolution docking scan consisted of 20,000 matches, with no post-processing (except for the removal of redundant matches). The matches were scored by the sum of the f values (Eq. 7), if constraints were generated for the complex. If no constraints were generated, the score was zero. The quality of a match was assessed by $C^{\alpha}$ ligand interface root-mean-square deviation, i-RMSD (ligand and receptor are the smaller and the larger proteins in the complex, respectively), calculated between the interface of the docked unbound ligand and the corresponding atoms of the unbound ligand superimposed on the bound ligand in the co-crystallized complex. Success was defined as at least one model with i-RMSD ≤5 Å in top 10 predictions. The results (Fig. 6) show significant success rate increase in the docking output when using constraints generated by the advanced TM, from 27% in the case of the basic TM, to 47% in the case of the advanced TM with NLP.

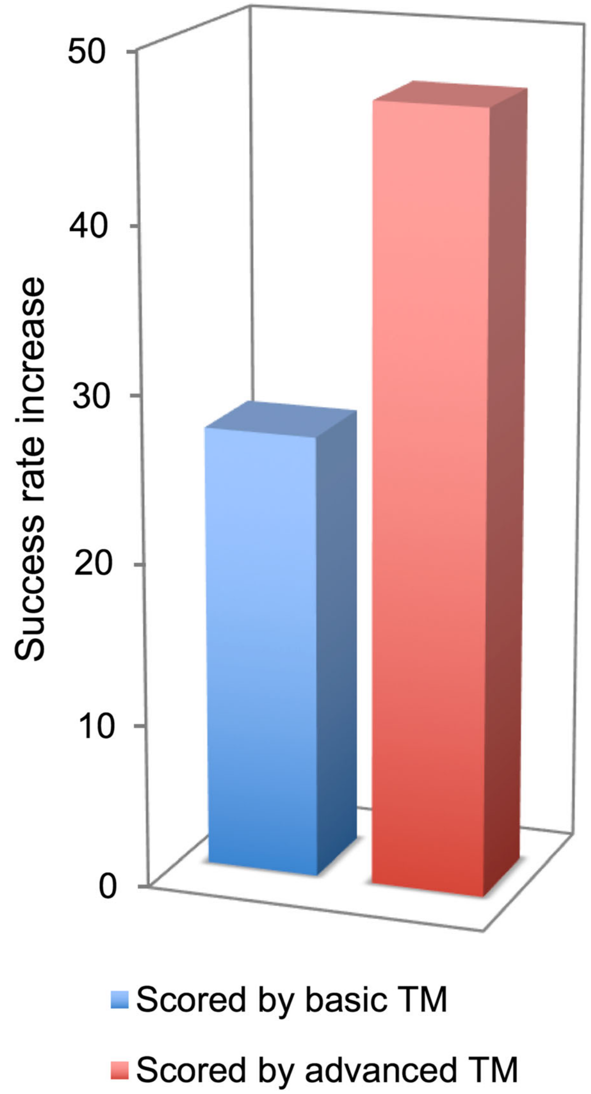

**Fig. 6** TM contribution to docking. The success rate increase of the rigid-body global docking scan by GRAMM using constraints generated by basic TM and the advanced TM with NLP

Since some authors might not include the required details in the abstracts of their papers, we plan to extend the automated analysis to the full-text articles, as well as to explore incorporation of the papers from bioaRxiv. This should increase of the size of the training sets for machine-learning models, and the number of available features, thus enabling the use of the deep learning methodologies for generation of the docking constraints. Such constraints could be potentially further improved by incorporating information automatically extracted from other publicly available PPI-related resources, leading to more accurate and reliable structural modeling of protein interactions.

## Conclusion

We explored how well the natural language processing techniques filter out non-interface residues extracted by the basic text mining protocol from the PubMed abstracts of papers on PPI. The results based on generic and specialized dictionaries showed that the dictionaries generated for the mining of information on whether two proteins interact, as well as generic English vocabularies are not capable of distinguishing relevant (interface) and irrelevant (non-interface) residues. Efficient filtering of irrelevant residues can be done only using a narrowly specialized dictionary, which comprises words relevant to PPI binding mode (binding site), combined with interpretation of the context in which residue was mentioned. Interestingly, the size of such specialized dictionary is not a critical factor for the protocol efficiency. We tested several methods of context analysis, based on dissection of the sentence parse trees. The best efficiency was achieved using machine-learning approaches for examining residue-containing and surrounding sentences (as opposed to the rule-based methods). Docking benchmarking showed a significant increase of the success rate with constraints generated by the advanced TM with NLP.

## Additional file

**Additional file 1:** Supporting information for the main manuscript, including Additional file 1: Figure S1-S9 and Table S1. (PDF 151 kb)

### Abbreviations

i-RMSD: Interface root-mean-square deviation; NLP: Natural language processing; PDB: Protein Data Bank; PPI: Protein-protein interactions; SVM: Support vector machines; TM: Text mining

### Acknowledgments

This study was supported by NIH grant R01GM074255 and NSF grants DBI1262621 and DBI1565107.

### Funding

This study was supported by NIH grant R01GM074255 and NSF grants DBI1262621 and DBI1565107.

### Availability of data and materials
The TM datasets used in this study are available from the corresponding authors on reasonable request.

### Authors' contributions
VDB designed and implemented the approach, generated the data and wrote the draft of the manuscript. PJK and IAV set the goals, analyzed the data, and wrote the manuscript. All authors read and approved the final manuscript.

### Ethics approval and consent to participate
Not applicable.

### Consent for publication
Not applicable.

### Competing interests
The authors declare that they have no competing interests.

Received: 24 October 2017 Accepted: 20 February 2018
Published online: 05 March 2018

### References
1. Vakser IA. Protein-protein docking: from interaction to interactome. Biophys J. 2014;107:1785–93.
2. Moal IH, Moretti R, Baker D, Fernandez-Recio J. Scoring functions for protein–protein interactions. Curr Opin Struc Biol. 2013;23:862–7.
3. de Vries SJ, van Dijk ADJ, Bonvin AMJJ. WHISCY: what information does surface conservation yield? Application to data-driven docking. Proteins. 2006;63:479–89.
4. Turinsky AL, Razick S, Turner B, Donaldson IM, Wodak SJ. Literature curation of protein interactions: Measuring agreement across major public databases. Database 2010; 2010:baq026.
5. Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics. 2001;17:S74–82.
6. Friedman C, Alderson PO, Austin JH, Cimino JJ, Johnson SB. A general natural-language text processor for clinical radiology. J Am Med Inf Assoc. 1994;1:161.
7. Fundel K, Kuffner R, Zimmer R. RelEx—relation extraction using dependency parse trees. Bioinformatics. 2007;23:365–71.
8. Califf ME, Mooney RJ. Relational learning of pattern-match rules for information extraction. In: Proc 16th Natl Conf Artificial Intelligence. Orlando: The AAAI Press, Menlo Park, California; 1999. 328.
9. Yakushiji A, Tateisi Y, Miyao Y, T. J. Event extraction from biomedical papers using a full parser. In: Proc Pacific Symp Biocomputing: 2001. World Scientific: 408–19.
10. Liu H, Keselj V, Blouin C, Verspoor K. Subgraph matching-based literature mining for biomedical relations and events. In: 2012 AAAI fall Symp series Inf retrieval knowledge disc biomed text. Arlington; 2012. p. 32–7.
11. Liu H, Hunter L, Keselj V, Verspoor K. Approximate subgraph matching-based literature mining for biomedical events and relations. PLoS One. 2013;8:e60954.
12. Peng Y, Gupta S, Wu CH, Vijay-Shanker K. An extended dependency graph for relation extraction in biomedical texts. In: Proc 2015 Workshop biomed natural language processing. Beijing; 2015. p. 21–30.
13. Bunescu RC, Mooney RJ. A shortest path dependency kernel for relation extraction. In: Proc Conf Human Language Tech Empirical Methods in Natural Language Processing: 2005. Association for Computational Linguistics: 724–31.
14. Mooney RJ, Bunescu RC. Subsequence kernels for relation extraction. In: Proc 2005 Conf (NIPS). Vancouver, MIT Press; 2005. p. 171–8.
15. Moschitti A. Making tree kernels practical for natural language learning. In: Proc 11th Conf Eur Ch Associ Comput Linguistics. Trento; 2006. p. 113–20.
16. Moschitti A. A study on convolution kernels for shallow semantic parsing. In: Proc 42nd Ann Meeting Assoc Comput Linguistics. Barcelona: Association for Computational Linguistics; 2004. p. 335–42.
17. Culotta A, Sorensen J. Dependency tree kernels for relation extraction. In: Proc 42nd Annual Meeting Association for Comput Linguistics. Barcelona: Association for Computational Linguistics; 2004. p. 423–9.
18. Quan C, Wang M, Ren F. An unsupervised text mining method for relation extraction from biomedical literature. PLoS One. 2014;9:e102039.
19. Blaschke C, Valencia A. The frame-based module of the SUISEKI information extraction system. IEEE Intell Syst. 2002:14–20.
20. Blaschke C, Andrade M, Ouzounis CA, Valencia A. Automatic extraction of biological information from scientific text: protein-protein interactions. In: Proc ISMB-99 Conf. Heidelberg: American Association for Artificial Intelligence; 1999. p. 60–7.
21. Temkin JM, Gilder MR. Extraction of protein interaction information from unstructured text using a context-free grammar. Bioinformatics. 2003;19:2046–53.
22. Kim S, Kwon D, Shin SY, Wilbur WJ. PIE the search: searching PubMed literature for protein interaction information. Bioinformatics. 2012;28:597–8.
23. Raja K, Subramani S, Natarajan J. PPInterFinder—a mining tool for extracting causal relations on human proteins from literature. Database 2013; 2013: bas052.
24. Jang H, Lim J, Lim JH, Park SJ, Park SH, Lee KC, Extracting protein-protein interactions in biomedical literature using an existing syntactic parser. In: Knowledge Disc Life Sci Literature Springer; 2006: 78–90.
25. He M, Wang Y, Li W. PPI finder: a mining tool for human protein-protein interactions. PLoS One. 2009;4:e4554.
26. Li M, Munkhdalai T, Yu X, Ryu KH. A novel approach for protein-named entity recognition and protein-protein interaction extraction. Math Probl Eng. 2015;2015:942435.
27. Peng Y, Arighi C, Wu CH, Vijay-Shanker K. Extended dependency graph for BioC-compatible protein-protein interaction (PPI) passage detection in full-text articles. In: Proc BioCreative V Challenge Workshop, vol. 30-5. Sevilla; 2015.
28. Koyabu S, Phan TT, Ohkawa T. Extraction of protein-protein interaction from scientific articles by predicting dominant keywords. Biomed Res Int 2015; 2015:928531.
29. Erkan G, Ozgur A, Radev DR. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In: Proc 2007 Joint Conf empirical methods natural language processing and computational natural language learning. Prague: Association for Computational Linguistics; 2007. p. 228–37.
30. Erkan G, Ozgur A, Radev DR. Extracting interacting protein pairs and evidence sentences by using dependency parsing and machine learning techniques. In: Proc 2nd BioCreative Challenge Evaluation Workshop: 2007, Madrid, Spain Fundación CNIO Carlos III: 287–292.
31. Miwa M, Saetre R, Miyao Y, Tsujii J. Protein–protein interaction extraction by leveraging multiple kernels and parsers. Int J Med Inform. 2009;78:e39-e46.
32. Zhou D, He Y. Extracting interactions between proteins from the literature. J Biomed Inform. 2008;41:393–407.
33. Thieu T, Joshi S, Warren S, Korkin D. Literature mining of host–pathogen interactions: comparing feature-based supervised learning and language-based approaches. Bioinformatics. 2012;28:867–75.
34. Blohm P, Frishman G, Smialowski P, Goebels F, Wachinger B, Ruepp A, Frishman D. Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. Nucl Acid Res. 2014;42:D396–400.
35. Wong A, Shatkay H. Protein function prediction using text-based features extracted from the biomedical literature: the CAFA challenge. BMC Bioinformatics. 2013;14:1.
36. Verspoor KM, Cohn JD, Ravikumar KE, Wall ME. Text mining improves prediction of protein functional sites. PLoS One. 2012;7:e32171.
37. Badal VD, Kundrotas PJ, Vakser IA. Text mining for protein docking. PLoS Comp Biol. 2015;11:e1004630.
38. Gao Y, Douguet D, Tovchigrechko A, Vakser IA. DOCKGROUND system of databases for protein recognition studies: unbound structures for docking. Proteins. 2007;69:845–51.
39. Banerjee S, Pedersen T. An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Proc 3rd Int Conf CompLinguistics Intelligent Text Processing. Mexico City: Springer-Verlag London; 2002. p. 136–45.
40. Banerjee S, Pedersen T. Extended gloss overlaps as a measure of semantic relatedness. In: Proc 18th Intl Joint Conf Artificial intelligence 2003, Acapulco, Mexico. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA: 805–810.
41. Lin D. An information-theoretic definition of similarity. In: Proc 15th Int Conf Machine Learning. Madison: Morgan Kaufmann Publishers Inc; 1998. p. 296–304.
42. Meng L, Huang R, Gu J. A review of semantic similarity measures in wordnet. Int JHybrid Inf Technol. 2013;6:1–12.
43. Pedersen T, Patwardhan S, Michelizzi J. WordNet:: Similarity: Measuring the relatedness of concepts. In: Demonstration papers at HLT-NAACL 2004: 2004, Boston, Massachusetts Association for Computational Linguistics: 38–41.

44. Miller GA. WordNet: a lexical database for English. Commun ACM. 1995;38:39–41.
45. Fellbaum C. WordNet: an electronic lexical database: MIT press, Cambridge; 1998.
46. De Marneffe MC, Manning CD, Stanford typed dependencies manual. In.: Technical report, Stanford University; 2008: 338–45.
47. De Marneffe MC, Manning CD. The Stanford typed dependencies representation. In: Proc Workshop Cross-Framework Cross-Domain Parser Evaluation. Manchester: Association for Computational Linguistics; 2008. p. 1–8.
48. Joachims T. Text categorization with support vector machines: learning with many relevant features. In: Nedellec C, Rouveirol C, editors. Machine learning: ECML-98, vol. vol. 1398. berlin: Springer; 1998. p. 137–42.
49. Joachims T. Making large-scale support vector machine learning practical. In: advances in kernel methods: MIT Press; 1999. p. 169–84.
50. Morik K, Brockhausen P, Joachims T, Combining statistical learning with a knowledge-based approach: A case study in intensive care monitoring (No. 1999, 24). In.: Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund; 1999.
51. Shatkay H, Feldman R. Mining the biomedical literature in the genomic era: an overview. J Comput Biol. 2003;10:821–55.
52. Vakser IA. Low-resolution docking: prediction of complexes for underdetermined structures. Biopolymers. 1996;39:455–64.
53. Zervanou K, McNaught J. A term-based methodology for template creation in information extraction. In: Proc 2nd Int Conf Natural Language Processing. Patras: Springer; 2000. p. 418–23.
54. Pucher M. Performance evaluation of WordNet-based semantic relatedness measures for word prediction in conversational speech. In: Proc 6th Int Workshop Comput Semantics. Tilburg; 2005.
55. Sanchez D, Sole-Ribalta A, Batet M, Serratosa F. Enabling semantic similarity estimation across multiple ontologies: an evaluation in the biomedical domain. J Biomed Inform. 2012;45:141–55.
56. Knecht LWS, Nelson SJ. Mapping in PubMed. J Med Libr Assoc. 2002;90:475–6.
57. Rebholz-Schuhmann D, Jimeno-Yepes A, Arregui M, Kirsch H. Measuring prediction capacity of individual verbs for the identification of protein interactions. J Biomed Inform. 2010;43:200–7.
58. Chowdhary R, Zhang J, Liu JS. Bayesian inference of protein–protein interactions from biological literature. Bioinformatics. 2009;25:1536–42.
59. Hakenberg J, Leaman R, Ha Vo N, Jonnalagadda S, Sullivan R, Miller C, Tari L, Baral C, Gonzalez G. Efficient extraction of protein-protein interactions from full-text articles. IEEE-ACM Trans Comp Biol Bioinf. 2010;7:481–94.
60. Plake C, Hakenberg J, Leser U. Optimizing syntax patterns for discovering protein-protein interactions. In: Proc 2005 ACM Symp applied computing. Santa Fe: ACM; 2005. p. 195–201.