Bioinformatics, 37(4), 2021, 497–505
doi: 10.1093/bioinformatics/btaa823
Advance Access Publication Date: 22 September 2020
Original Paper

OXFORD

Structural bioinformatics

Text mining for modeling of protein complexes enhanced by machine learning

Varsha D. Badal[1], Petras J. Kundrotas[1,*] and Ilya A. Vakser[1,2,*]

[1]Computational Biology Program and [2]Department of Molecular Biosciences, The University of Kansas, Lawrence, KS 66045, USA

*To whom correspondence should be addressed.

Associate Editor: Arne Elofsson
Received on August 26, 2019; revised on September 4, 2020; editorial decision on September 7, 2020; accepted on September 8, 2020

Abstract

Motivation: Procedures for structural modeling of protein–protein complexes (protein docking) produce a number of models which need to be further analyzed and scored. Scoring can be based on independently determined constraints on the structure of the complex, such as knowledge of amino acids essential for the protein interaction. Previously, we showed that text mining of residues in freely available PubMed abstracts of papers on studies of protein–protein interactions may generate such constraints. However, absence of post-processing of the spotted residues reduced usability of the constraints, as a significant number of the residues were not relevant for the binding of the specific proteins.

Results: We explored filtering of the irrelevant residues by two machine learning approaches, Deep Recursive Neural Network (DRNN) and Support Vector Machine (SVM) models with different training/testing schemes. The results showed that the DRNN model is superior to the SVM model when training is performed on the PMC-OA full-text articles and applied to classification (interface or non-interface) of the residues spotted in the PubMed abstracts. When both training and testing is performed on full-text articles or on abstracts, the performance of these models is similar. Thus, in such cases, there is no need to utilize computationally demanding DRNN approach, which is computationally expensive especially at the training stage. The reason is that SVM success is often determined by the similarity in data/text patterns in the training and the testing sets, whereas the sentence structures in the abstracts are, in general, different from those in the full text articles.

Availabilityand implementation: The code and the datasets generated in this study are available at https://gitlab.ku.edu/vakser-lab-public/text-mining/-/tree/2020-09-04.

Contact: vakser@ku.edu or pkundro@ku.edu

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

1 Introduction

Protein–protein interactions (PPI) play a key role in cellular mechanisms. Computational approaches, such as protein docking, are important for the structural characterization of PPI. Protein docking determines the structure of a protein–protein complex, given the structure of the interacting proteins (Vakser, 2014). A typical docking pipeline involves three major steps: (i) global scan generating multiple tentative protein–protein matches (docking poses), (ii) evaluation of these poses by physics-based or knowledge-based scoring functions and (iii) structural refinement of the top-scoring matches. The ability of the modeling protocol to differentiate between correct (near native) and incorrect docking poses determines the overall docking success. Knowledge of even a single residue at the protein–protein interface is a powerful constraint for the docking search, dramatically reducing the number of docking poses to be evaluated, thus significantly increasing reliability of the resulting docking models.

Such knowledge can be acquired from the PPI-related scientific publications. However, rapidly growing number of biomedical publications in public repositories, such as PubMed, renders manual extraction of relevant information nearly impossible. This necessitates utilization of automated text mining (TM) procedures for generating docking constraints (extracting interface residues from the text of publications). The textual content of the publications is easily understandable by human experts, but processing of that information by computers requires TM algorithms, specific for each particular field of study. So far, TM applications have been mainly focused on prediction of interactions between biological macromolecules (Caufield and Ping, 2019; Li *et al.*, 2016; Papanikolaou *et al.*, 2015; Raja *et al.*, 2020; Tagore *et al.*, 2019; Yu *et al.*, 2018). However, the TM algorithms applicable to protein–protein docking currently are underdeveloped.

The TM techniques extract usable bits of information from the body of text. A scientific text has varying information concentration and coverage depending on a section. Abstracts of scientific

publications typically are readily and freely available, have high information density, but have limited content coverage compared to the full-text papers (Lan and Su, 2010; Martin *et al.*, 2004; Schuemie *et al.*, 2004). The full texts use longer sentences and parenthesized material (Cohen *et al.*, 2010) and have heterogeneous distribution of information (as measured by density of keywords in various sections) (Shah *et al.*, 2003). Access to the full-text papers creates a more comprehensive source (corpus) for the TM and increases the recall compared to the abstracts (Caporaso *et al.*, 2008; Westergaard *et al.*, 2018). However, copyright restrictions generally limit the use of full text articles in the automated TM protocols (Cohen and Hersh, 2005; Rodriguez-Esteban, 2009). The number of PMC-OA (the repository of freely available full-text papers) articles is not increasing at the same rate as the number of PubMed abstracts. The full-text articles have statistical properties (such as a term frequency in the document) that are more robust, but have more noise compared to the abstracts (Lin, 2009). TM of the full-text papers has helped in extraction of various biological information (Corney *et al.*, 2004; Fink *et al.*, 2008; Friedman *et al.*, 2001; Gerner *et al.*, 2010, 2012; Mallory *et al.*, 2015; McIntosh and Curran, 2009), including one on non-structural aspects of PPI (Dogan *et al.*, 2017; Hakenberg *et al.*, 2010; Huang *et al.*, 2004; Krallinger *et al.*, 2008; Peng *et al.*, 2016).

Understanding the textual context of publications requires recognition of specific patterns in texts that can be identified by machine learning (ML) techniques, especially, deep learning (DL) approaches implemented using neural networks (NNs) with several hidden layers (Ching *et al.*, 2018; Habibi *et al.*, 2017). Each successive layer learns higher level of abstraction (Bengio *et al.*, 2013; LeCun *et al.*, 2015). NN are trained using the back-propagation algorithm (LeCun *et al.*, 2015) where the error (the difference between actual and desired output) is projected backwards layer-by-layer, with the connection weights adjusted in proportion (Rumelhart *et al.*, 1986). The NN applications include, but not limited to automatic speech recognition (Schwenk, 2007), machine translation (Mikolov, 2012), paraphrasing (Turney, 2013), image and scene annotation (Socher *et al.*, 2011a,b; Weston *et al.*, 2011), as well as prediction of protein–protein interactions (Yao *et al.*, 2019). However, DL is computationally demanding, especially at the training stage (which is necessary to repeat, e.g. when the model changes), and thus such approaches may be employed when simpler ML algorithms, e.g. Support Vector Machine (SVM), do not suffice.

For computational procedures, it is desirable to represent words using numbers. Simplistic approaches may assign a unique single number (scalar) to each word of a language (e.g. in lexicographical order). The next step is to represent a word as a series of numbers (word vector or word embedding), so that vector operations can be meaningfully applied (Mikolov *et al.*, 2013a,b). Then, the inner product of the two vectors would be a measure of similarity of the two words, the sum of the two vectors would reflect the combined meaning of the two words, and the subtraction of the two vectors (offset) would capture the relations (e.g. plural relations, like 'molecules versus molecule' and 'residues versus residue' would have similar offsets). Word vectors, e.g. implemented in the word2vec software, can be efficiently estimated on a large scale (Mikolov *et al.*, 2013a). They are widely used as a first generic step in a united architecture for solving a specific Natural Language Processing (NLP) task using deep NNs (Collobert and Weston, 2008; Irsoy and Cardie, 2014a,b; Mikolov *et al.*, 2013a,b,c; Socher *et al.*, 2011a,b), e.g. for machine translation requiring large vocabulary across multiple languages (Brants *et al.*, 2007). The word vectors are used in analysis of sentence-level sentiment, a quantitative score of a subjective information (e.g. 'tone of a speaker' or 'attitude of a customer') (Socher *et al.*, 2011a,b, 2013).

Earlier, we implemented an algorithm that searches for the patterns of letters and digits typically used by authors referring to a specific residue in a protein (referred to as basic TM in this article). We showed that such information, although mined in a simplistic manner, efficiently excludes incorrect docking models from consideration and thus significantly improves docking success rate (Badal *et al.*, 2015). However, without interpretation of the context in which the residue appears in the text, the initial pool of the extracted data inevitably contains residues that are not relevant for the binding of specific proteins. Thus, the initially extracted set of residues needs further post-processing. Recently, we investigated filtering of the non-relevant residues by several Natural Language Processing (NLP) techniques, such as keywords semantic similarities, dictionaries look-up and analysis of sentence parse trees with and without SVM model (Badal *et al.*, 2018). However, the amount of non-relevant residues still remained high. In this article, for the analysis of context in which the residue is mentioned, we use a deep recursive neural network (DRNN) model based on the concept of the word vectors. We compared the performance of the DRNN and SVM models in various training/testing schemes and showed that the DRNN model is superior, albeit slightly, to the SVM model when training is performed on the PMC-OA full-text articles and applied to classification (interface versus non-interface) of the residues spotted in the PubMed abstracts. When both training and testing is performed on the full-texts articles, the performance of the models is similar. Thus, the use of DRNN, which is computationally expensive especially at the training stage, in such cases may be unnecessary.

2 Materials and methods

2.1 Basic TM protocol

Our basic TM tool consists of information retrieval (IR; retrieval of publications relevant to a particular pair of interacting proteins), and information extraction (IE; extraction of residues in the abstracts of identified publications) (Badal *et al.*, 2015). In this study, in addition to using PubMed resources from https://www.ncbi.nlm.nih.gov/ (NCBI Resource Coordinators, 2013), we also downloaded and stored locally the PMC-OA full text articles from http://www.ncbi.nlm.nih.gov/pmc/tools/ftp/. Thus, the IR stage was modified to incorporate the local availability of the full-text articles, as opposed to E-fetch from the E-utilities for the PubMed abstracts. PMID (a unique ID for a PubMed abstract) and PMCID (a unique ID for a PMC-OA full-text articles) are different for the same article. For computational efficiency, mapping between them, allowing fetching of a full-text article given a PMID of its abstract, was implemented as a PostgreSQL table. As in our previous study (Badal *et al.*, 2015), articles, relevant to a protein pair, were retrieved by AND-queries (requiring that both proteins in the complex are mentioned in the text) and OR-queries (either of the proteins is mentioned) using NCBI E-utilities (http://www.ncbi.nlm.nih.gov/books/NBK25501). Then using PMID-PMCID mapping, the available full-text articles for that protein pair were identified (for 2 640 816 PMID only 196 912 PMCID were mapped). These full-text articles and abstracts were subjected to the IE stage of the protocol (Fig. 1), which spots different variations of the residue name and its number (Badal *et al.*, 2015). A simple residue filtering was performed by checking that the residues are on the protein surfaces. All or part of residue-containing sentences (hereafter termed as R-sentences) from the full-text articles were used to estimate the effectiveness of the basic TM on the full-texts and to train the DRNN and the SVM models. The trained models were used to classify residues found in the PubMed abstracts and full-text articles as interface or non-interface. In the same training/testing scenarios, both SVM and DRNN models used the same list of raw-mined residues. Thus, the difference in performance of the ML models generated by the two approaches originates only from the difference in the text manipulation steps (see below).

2.2 Evaluating performance of the TM protocol

Whereas residues essential for protein recognition could be outside the protein–protein interface, the goal of our TM approach is to generate constraints for docking, which implies residues at the protein–protein interface. Thus, the performance of the TM protocol for a particular PPI, for which N residue-containing articles (abstract-

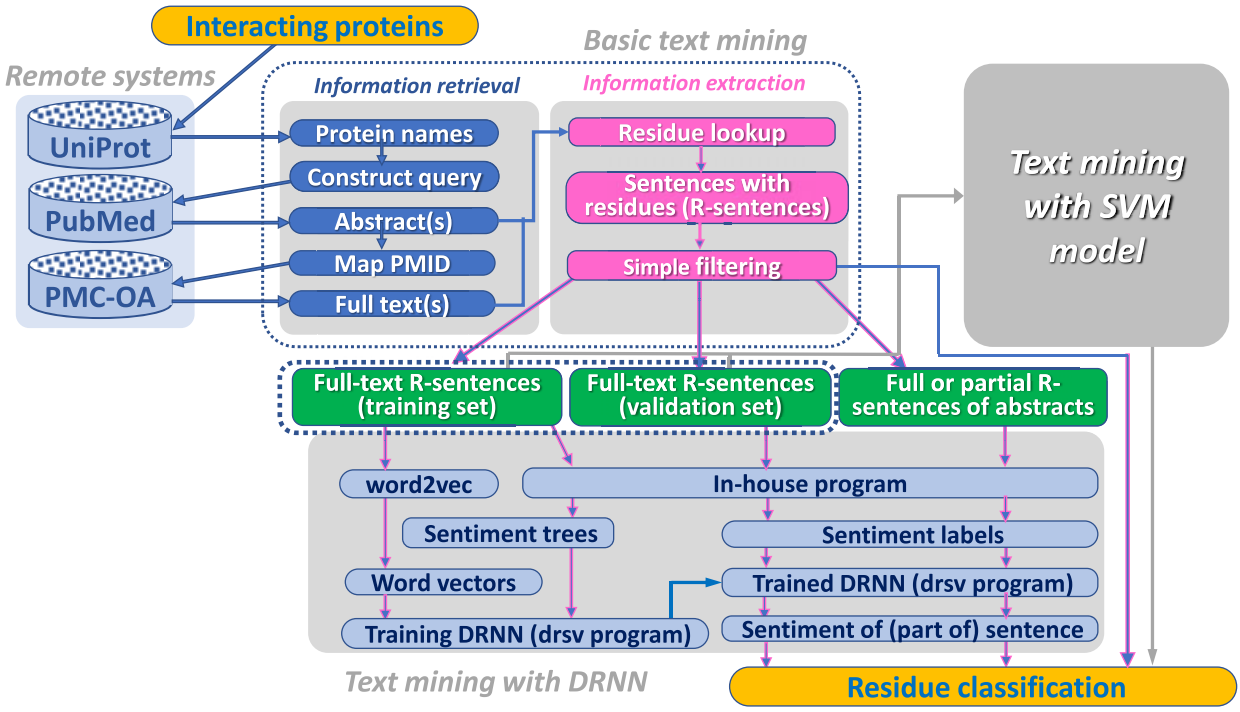

Fig. 1. Flowchart of the text-mining system. Algorithm of the TM with the SVM model is the same as in (Badal *et al.*, 2018) and thus is not shown in detail. Both full text sets are unified in one training set (shown by the dashed line) when the trained NN is tested on the PubMed abstracts

only or full-text) were retrieved, was evaluated as (Badal *et al.*, 2015).

$$P_{\mathrm{TM}} = \frac{\sum_{i=1}^{N} N_i^{\mathrm{int}}}{\sum_{i=1}^{N} (N_i^{\mathrm{int}} + N_i^{\mathrm{non}})}, \quad (1)$$

where N_i^{int} and N_i^{non} are the numbers of interface and non-interface residues, correspondingly, mentioned in article i for this PPI, which were not filtered out by an algorithm. If all residues in an article were purged, this article was excluded from the P_{TM} calculations. Distribution of P_{TM} for all complexes in the dataset provides detailed decription of efficiency of the TM algorithm for that dataset. We also compared the performance of two algorithms for residue filtering (Badal *et al.*, 2018) by

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

where $N_{tar}^{X_1}(P_{TM})$ and $N_{tar}^{X_2}(P_{TM})$ are the number of targets with P_{TM} value yielded by algorithms X_1 and X_2, respectively. Since the major contributions to P_{TM} distribution are from all-false-positive ($P_{TM} = 0$) and all-true-positive ($P_{TM}=1$) cases, algorithm efficiency can be roughly assessed just by the two extreme values of $\Delta N(P_{\mathrm{TM}})$ at $P_{TM}=0$ and $P_{TM}=1$. The values $\Delta N(0)<0$ and $\Delta N(0)>0$ indicate better performance of model X_1 with respect to model X_2 in purging of the PPI-irrelevant residues from the mined articles.

2.3 Datasets

The approaches were benchmarked on the set of 579 non-redundant (at 30% sequence identity) binary protein–protein complexes from the DOCKGROUND resource (https://dockground.compbio.ku.edu) (Kundrotas *et al.*, 2018). The dataset for training ML models (DRNN and SVM) consisted of 4982 residue-containing sentences (hereafter referred to as R-sentences), which passed the initial screening of residue-containing sentences automatically extracted by the OR-queries from the full-text PMC-OA articles (full training set). By querying the native PDB structures, these sentences were classified into 1605 positive (interface residue) and 3377 negative (non-interface residue) R-sentences. The interface residues were defined by 6 Å distance between any atoms across the chains interface. The dataset for testing the ML models comprised 5786R-sentences (or their parts around identified residues, see Section 3), extracted from the PubMed abstracts by the OR-queries (abstract testing set). Since only a small fraction of the training text came from PMC-OA abstracts of the PMC-OA articles, we did not exclude PubMed abstracts that have PMC-OA full-text articles available.

For testing the ML models on the full-text articles, the training set consisted of 803 positive and 1689 negative sentences extracted from the articles describing studies of the top 290 complexes from the full list of complexes (sorted in alphabetical order of corresponding PDB codes). The remaining sentences from the articles describing studies of the rest of the complexes comprised the test set.

2.4 Generation of keywords

To identify keywords relevant to the protein binding, we computed the differences (bias) in word frequencies (percent of the sentences with that word) calculated separately for the positive and negative R-sentences in the full training set. Words with the biases between 1 and -1, stop words and names of a protein, an amino acid or species were omitted from consideration. This resulted in 47 PPI+ive (PPI-relevant, positive bias) and 37 PPI-ive (PPI-irrelevant, negative bias) keywords (Supplementary Table S1).

2.5 Support vector machine model

The features for the SVM model consisted of the scores, calculated from the parse trees of the R- and the context (immediately preceding and following the R-sentence) sentences. This score reflects an effective count of the edges on the parse tree between a residue (or root for the context sentences) and all the keywords from the Supplementary Table S1 [detailed description is published elsewhere (Badal *et al.*, 2018)]. Sentence parse trees were built by the Perl module of the Stanford parser (De Marneffe and Manning, 2008a, 2008b) http://nlp.stanford.edu/software/index.shtml downloaded from http://search.cpan.org site. The SVM model was trained and validated using program SVMLight with linear, polynomial and

RBF kernels (Joachims, 1998, 1999; Morik *et al.*, 1999). Analysis of the results indicated (data not shown) that the best SVM performance was achieved using RBF kernel with gamma 0.25 for both training-testing schemes (training on full-texts, testing on abstracts and training and testing on full-texts only).

2.6 Docking protocol

Docking by our GRAMM procedure (Vakser, 1998) was evaluated on the unbound protein structures from the DOCKGROUND X-ray benchmark set 4 (Kundrotas *et al.*, 2018). The set originally consisted of 389 protein complexes, out of which 57 were also present in the set used in development of the TM protocols and thus were excluded from docking. The residues identified by the TM protocols were used to generate a confidence score (for details, see Badal *et al.*, 2018) to increase the weight of protein–protein matches that had these residues at the predicted interface (Badal *et al.*, 2018). The quality of a match was assessed by C^{α} ligand (smaller protein in the complex) interface root-mean-square deviation (i-RMSD) between the interface of the docked unbound ligand and corresponding atoms of the unbound ligand superimposed on the bound one in the co-crystallized complex. Docking was evaluated by the percentage of the successfully predicted complexes. The low-resolution protein–protein match was considered correct if it was inside the binding funnel (i-RMSD$\leq$8 Å; Hunjan *et al.*, 2008), making it a practical starting point for the refinement trajectories (Hunjan *et al.*, 2008). Protein complex was considered predicted successfully at low resolution if such match was among top 100. Docking was also assessed by the enrichment of the prediction pool by the correct matches across all complexes (overall increase in the number of the top-100 predictions). TM provides constraints for docking regardless of the complexity of their modeling (e.g. for protein interactions involving large conformational changes upon binding), which should be instrumental for the refinement. In practical docking, the number of models for the refinement is limited by the computational efficiency of the refinement protocol. Currently available protocols, including those with the GPU implementation, make 100 refinement trajectories practical (Dauzhenka *et al.*, 2018).

3 Results and discussion

3.1 Architecture of NN

For the DRNN training, we generated PPI-specific sentiment tree bank, which is a set of binary trees (Fig. 2) of the R-sentences from the training set with each leaf and internal node tagged by a sentiment labels a_j and b_j, respectively (j counts words in the sentence). According to the Stanford Sentiment Treebank (Socher *et al.*, 2013), we utilized five standard sentiment classes: very +ive (labeled 4), +ive (3), neutral (2), -ive (1), very –ive (0). In our case, a sentiment (or more precisely, its label), quantifies the degree to which the information in a sentence is relevant to protein–protein docking, i.e. how likely residues mentioned in the sentence are at the protein–protein interface (label 4 is most probable and label 0 is least probable). In a sentence of N words, the b_j is calculated as

$$b_j = \max(b_{j-1}, a_j) \quad \text{or} \quad b_j = \min(b_{j-1}, a_j),\ j = 2, \ldots, N, \tag{3}$$

for the positive and negative sentences, respectively. The sentiment label a_j is

$$a_j = \begin{cases} F_j & \text{, if word j is a keyword} \\ \text{round}\left(2 \pm \frac{j}{w}\right) & \text{, for the rest of the words} \end{cases}, \tag{4}$$

where F_j is the fixed sentiment label for PPI+ive and PPI-ive keywords (Supplementary Table S1).

Such labeling scheme ensures that the final sentiment label of a sentence mentioning interface residues is 3 or 4 (0 or 1 for sentences with non-interface residues) and captures the baseline trend of a sentiment, steadily increasing for the positive and decreasing for the negative sentences. The sets of a_j and b_j were the first part of the input, necessary for the DRNN training.

The second part of the training input was a set of initial word vectors (numeric weights associated with the word), $\{\vec{v}^k\}$, for each of the 74 438 unique words in the sentences of the training set (out of ~20 M total words). The vectors were generated by the word2vec program with skip-gram model (a predictive language model that works well for even rarely used words) and the default training window size of 10 (the number of considered words in the context) (Mikolov *et al.*, 2013a). The dimensionality of the word vectors was set to 300, considered sufficient for complex NLP tasks (Jurafsky and Martin, 2017; Mikolov *et al.*, 2013b; Pennington *et al.*, 2014). The word vectors corresponding to similar words were distributed close to each other (Fig. 3). The amino acids were in one region of the vector space, as were the words associated with shapes. Similarly, co-localized were words such as 'interaction' and 'complex.' Antonyms, such as hydrophobic, hydrophilic, are also in the proximity of each other, indicating that these terms are linguistically interchangeable.

Both input components were submitted to the program drsv (https://github.com/oir/deep-recursive) (Irsoy and Cardie, 2014a) to train 3-layers DRNN model. The DRNN learned over ~10 epochs (epoch is defined as a sweep through the entire training set). Beyond 10 epochs, DRNN was getting over-trained (Supplementary Fig. S1). The same program was used to evaluate the sentiment for the entire or partial sentences using trained DRNN model. In this case, the input consisted of the sentiment labels a_j (Equation 2) assigned to the words of a sentence or its parts. Such DRNN architecture with corresponding sentiment treebanks (domain knowledge specific or generic) is widely used [e.g. in the analysis of Netflix movie reviews (Irsoy and Cardie, 2014a; Socher *et al.*, 2013)].

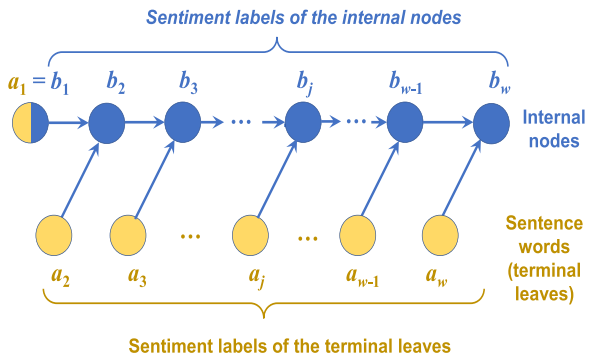

Fig. 2. Schematic representation of a sentence binary tree and associated sentiment labels

3.2 Mining of full-text articles

The full text of an article provides much more information than its abstract. But due to copyright restrictions only just over one million articles are freely available in the PMC-OA database, compared to ~26 million entries in the PubMed database of freely available abstracts. This causes significantly better TM performance on the PubMed abstracts than on the abstracts of the PMC-OA articles (Table 1).

The limited access to the full texts is counterweighted by the abundant information in them, as the overall TM performance on the PMC-OA full-text articles is comparable to that on the PubMed abstracts (Table 1). Significantly better TM performance on PMC-OA full-texts than on the PMC-OA abstracts (Table 1) points to more frequent mentioning of residues in the full texts (for 149 complexes, all mined residues were in the full texts only). However, due to lesser space constraints in the full texts, residues there are mentioned in a variety of contexts. This leads to a significantly larger number of PPI-irrelevant residues in the full texts than in the abstracts (corresponding bars at P_{TM}=0 and P_{TM}=1 in Fig. 4).

Research on a specific protein interaction could be published only in journals with limited access to their full texts. Our results

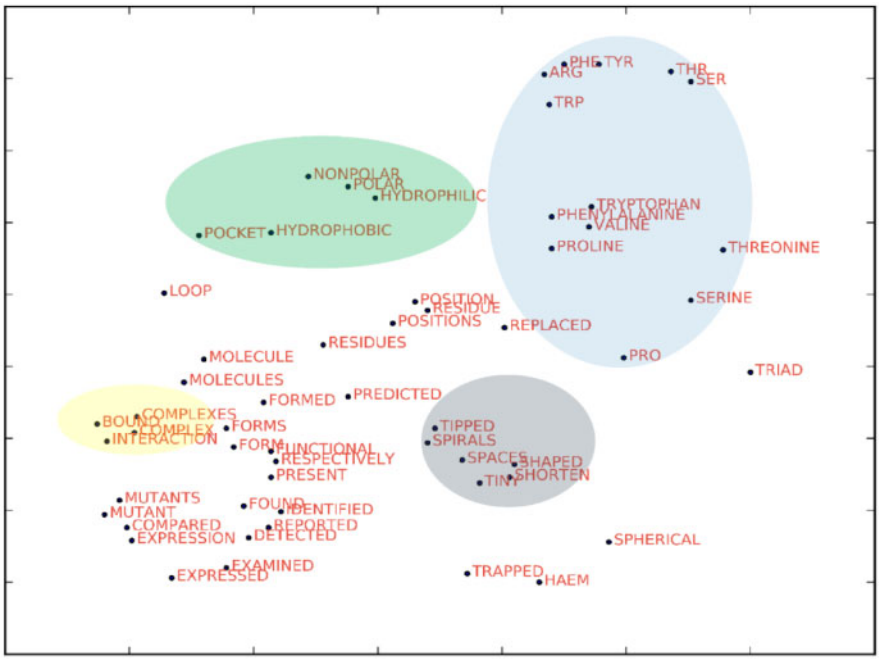

Fig. 3. Example of the initial word vectors distribution. Highlighted areas show similar words in the same region of the vector space. The distribution was generated by arbitrarily choosing a set of 55 words that are typically found in PPI publications, not meeting any scientific criteria, but representing a rich mix of domain vocabulary. The words were put in a list, along with the file containing word-vector lookup table (output of word2vec). The t-SNE software (van der Maaten and Hinton, 2008) extracted relevant 55 word-vectors and performed dimensionality reduction, until the required criteria of given perplexity (40) is met and the final dimensionality is reduced to 2. The points were plotted and labelled using pyplot in a python script. The highlighted areas were overlaid on the graph

Table 1. Overall performance of basic TM on abstracts (PubMed and PMC-OA) and full texts (PMC-OA)

Dataset	Query Type[a]	L_{tot}[b]	L_{int}[c]	Coverage (%)[d]	Success (%)[e]	Accuracy (%)[f]
PubMed abstracts	AND	128	108	22.1	18.7	84.4
PubMed abstracts	OR	328	273	56.6	47.2	83.2
PMC-OA abstracts	AND	37	21	6.3	3.6	56.7
PMC-OA abstracts	OR	164	89	28.3	15.3	54.2
PMC-OA full-text	AND	103	70	17.7	12.0	67.9
PMC-OA full-text	OR	313	238	54.0	41.1	76.0

[a]AND and OR query requires that the name of both proteins (AND) or either protein (OR) is mentioned in the returned document.
[b]Number of complexes, for which TM retrieved at least one article with residues.
[c]Number of complexes with at least one interface residue found in the retrieved articles.
[d]Ratio of L_{tot} and total number of complexes (579).
[e]Ratio of L_{int} and total number of complexes (579).
[f]Ratio of L_{int} and L_{tot}.

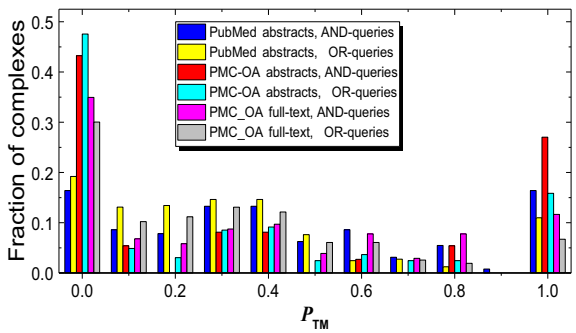

Fig. 4. Basic TM on abstracts and full-texts. The performance P_{TM} for individual complexes was calculated by Equation 1. The distribution is normalized to the total number of complexes for which residues were extracted (Table 1)

indicated that for a significant part of the complexes in our set (75 out of 579, or ~13%) this is indeed the case (one such example is shown in Supplementary Fig. S2 with the detailed description in Supplementary Text S1). Thus, we argue that, at least presently, PMC-OA full-text articles are more suitable for thorough analysis of residue-mentioning context (with consequent application to the residue purging in the PubMed abstracts) rather than for the extraction of the raw information.

Both SVM and DRNN models similarly affected the TM of the full-text articles (Table 2 and Fig. 5) and their abstracts (Supplementary Table S2 and Fig. S3). SVM model purged all initially mined residues (i.e. all mined residues were considered non-interface ones) for a smaller number of complexes (second column in Table 2). At the same time, it was slightly better in removing non-interface residues from the full-text articles (last column in Table 2). Compared to the basic TM, both SVM and DRNN models for the full-text articles significantly increased the fraction of complexes,

Table 2. Overall TM performance on test set of PMC-OA full-text articles retrieved by OR-queries with simplified residue filtering (basic TM) and with residue filtering by SVM and DRNN models trained on reduced full-text training set

Model	L_{tot}[a]	L_{int}[b]	Coverage (%)[c]	Success (%)[d]	Accuracy (%)[e]	$\Delta N(0)$[f]	$\Delta N(1)$[f]
Basic TM	157	115	60.6	44.4	73.2	–	–
SVM	87	58	33.6	22.4	66.7	–22	+15
DRNN	75	46	28.9	17.8	61.3	–24	+11

[a]Number of complexes for which TM found at least one article with residues.
[b]Number of complexes with at least one interface residue found in articles.
[c]Ratio of L_{tot} and total number of complexes (259).
[d]Ratio of L_{int} and total number of complexes (259).
[e]Ratio of L_{int} and L_{tot}.
[f]From Equation 6 with values from basic TM (first row) as X_2.

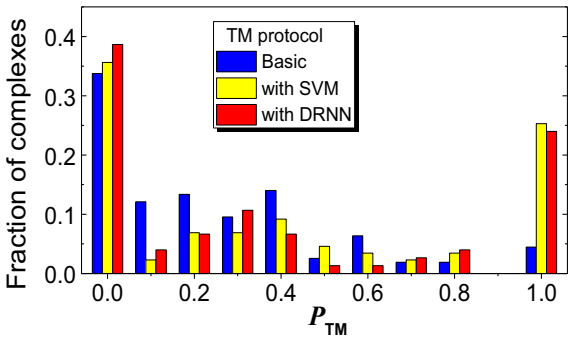

Fig. 5. Comparison of text-mining protocols on full texts. Both SVM and DRNN were trained on reduced full-text training set. The performance P_{TM} was calculated by Equation 1. The distribution is normalized to the total number of complexes for which residues were extracted (Table 2)

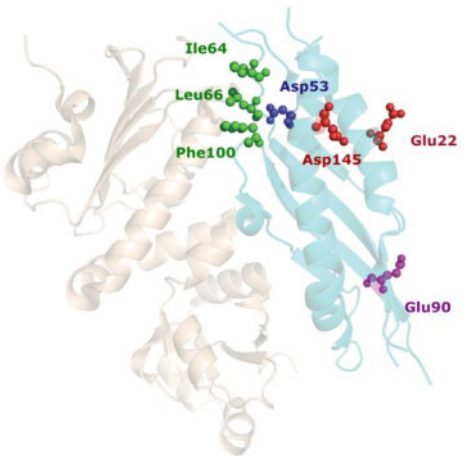

Fig. 6. Example of residues mined from full texts. The structure is 1usu, chain A (gray) and B (cyan). Out of six residues identified by the basic TM (four at the interface, in green and blue, and two not at the interface, in red) NLP SVM and DRNN models correctly classified three interface residues (green) and failed to identify Asp53 (which could be easily identified by a human reader, see Supplementary Text S2). Only one non-interface residue (magenta) was mined from the PubMed abstracts. The abstracts from PMC-OA did not predict any residues in basic TM. Details are in Supplementary Text S2

for which all mined residues are located at the complex interfaces. In the abstracts of the PMC-OA articles, DRNN and SVM removed all retrieved abstracts with the residues for ~70% of the complexes. Thus, the difference between the two models was not statistically significant. The performance of the SVM model only slightly depends on the keywords used for the SVM training and testing [Supplementary Table S3 and Fig. S4 show the results for the SVM model with the manually selected keywords from our previous study (Badal *et al.*, 2018)]. In our simplified scheme, we assigned a definite sentiment only to frequently appearing words designated as PPI keywords. Thus, we could miss infrequently occurring words or word groups that carry a strong sentiment. This is illustrated in Figure 6 where residue Asp53 at the interface of heat shock HSP82 and AHA1 proteins was incorrectly filtered by both SVM and DRNN models. However, human reader can easily identify this residue as the interface one (additional examples in Supplementary Text S3). In the future, performance of a DRNN model can be potentially improved by the use of PPI-specific hand-curated sentiment tree bank.

3.3 TM of abstracts

When both SVM and DRNN models are trained on full-text articles, and applied to classification of residues in the abstracts, the results are similar, with somewhat better performance of the DRNN model (reflected in the last column of Table 3 and the rightmost columns of Fig. 7). For this model, however, the larger fraction of complexes with only interface residues in the final list is counterweighted by the largest fraction of complexes, for which only non-interface residues were mined (left- and rightmost columns in Fig. 7). The training of the DRNN model was done on the entire set of full-text articles, but its performance only weakly depends on the size of the training set (Supplementary Table S4 and Fig. S5). The SVM model was better in removing complexes with only non-interface residues mined, but failed in increasing the number of complexes, for which all mined residues are PPI-relevant (Table 3).

We argue that performance of the SVM model suffered from the different structure of sentences in the full-texts and the abstracts. The DRNN model learns data/text patterns at a higher level of generality, and thus easily adapts to different domains, as diverse as, for example, protein docking and Netflix movie reviews. This suggests the use of DL algorithms for analysis of TM results when, for example, a particular PPI is widely studied by a variety of authors using different lexical semantic styles. On the other hand, SVM models may be better in finer analysis of articles of the same group of authors with similar writing styles.

Besides better adaptation of DRNN to the different sentence structures, it also has an advantage of an easy implementation of independent classification of multiple residues in a sentence by limiting the context to a few words around the residue (contextual window) and estimating a sentiment for that part of the sentence only. Obviously, a smaller contextual window allows independent classification of a larger number of residues in the sentence. However, due to the loss of broader contextual information embedded in the trained DRNN model, the sentiment accuracy may decrease. Our results indicate that the optimal TM performance is achieved when the sentiment is calculated for sentence fragments of seven words around the residue (Fig. 8). Overall, the DRNN model with the contextual window significantly improves filtering of the non-PPI residues, while only slightly reducing the coverage of the dataset (Table 3 and Fig. 7). Supplementary Figure S6 illustrates the

Table 3. Overall TM performance on PubMed abstracts retrieved by OR-queries with simplified residue filtering (basic TM) and with residue filtering by the SVM and DRNN models

Model	L_{tot}[a]	L_{int}[b]	Coverage (%)[c]	Success (%)[d]	Accuracy (%)[e]	$\Delta N(0)$[f]	$\Delta N(1)$[f]
Basic TM	328	273	56.6	47.2	83.2	–	–
SVM	182	135	31.4	23.3	74.1	−15	−3
DRNN (whole sentence)	179	120	30.9	20.7	67.0	−3	+6
DRNN (7-word window)	150	104	25.9	18.0	69.3	−16	+13

Note: Trained DRNN model was used for classifying residues in the entire sentence, as well as using 7-word window around mined residues. Both SVM and DRNN models were trained on the complete full-text training set.

[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 L_{tot} and total number of complexes (579).

[d]Ratio of L_{int} and total number of complexes (579).

[e]Ratio of L_{int} and L_{tot}.

[f]From Equation 6 with values from basic TM (first row) as X_2.

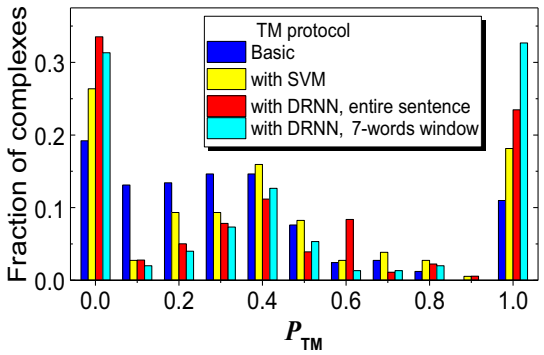

Fig. 7. Comparison of text-mining protocols on abstracts. The PubMed abstracts were retrieved by the OR-queries with simplified residue filtering (basic TM) and with the residue filtering by SVM model and DRNN. Both SVM and DRNN were trained on entire full-text training set. DRNN was applied for classifying residues in the entire sentence and with 7-words window around the mined residues. The performance P_{TM} for individual complexes was calculated by Equation 1. The distribution is normalized to the total number of complexes for which residues were extracted (Table 3)

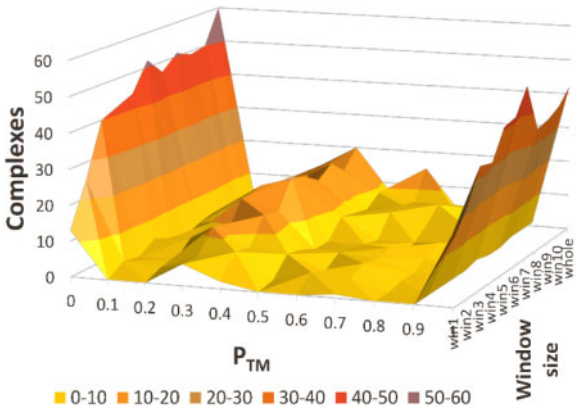

Fig. 8. TM performance with residue filtering by DRNN using different window sizes around mined residues. DRNN was trained on the entire training set of PMC-OA full-text articles. The performance P_{TM} was calculated by Equation 1

advantage of sentiment calculation using context window for cationic trypsin—tryptase inhibitor complex.

3.4 Application to modeling of protein complexes

We tested the applicability of the above protocols to protein docking. The TM protocols used were basic TM, SVM and DRNN

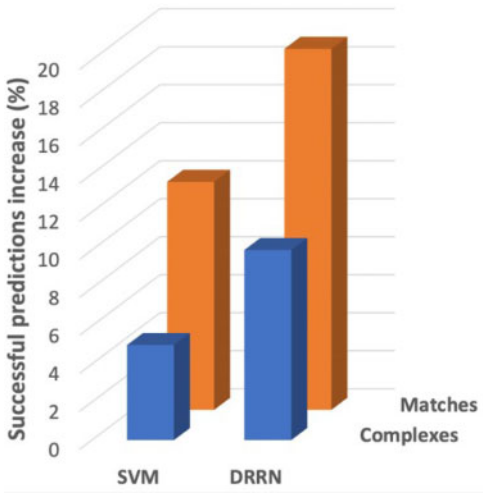

Fig. 9. The increase in successful docking predictions over the basic TM protocol. The change of the number of correctly predicted complexes is in blue and the change of the number of correct matches in top 100 is in orange

(trained on whole sentences)—all applied to full-text papers. The results showed the number of successfully predicted complexes increase over the basic TM by 5% and 10% using SVM and DRNN correspondingly. The total number of the top-100 correct matches across all complexes increased by 12% and 19%, respectively (Fig. 9). The current added value of the TM in docking is limited in part by the lack of uniform standard of residue numbering. In different studies, the numbering often depends on the sequence fragment used in the experiments. The numbering of residues in the PDB structures is even more complicated as it may depend not only on what fragment was crystallized, but also on other factors (e.g. for comparison with homologous proteins, numbers like 20A and 20B may be inserted between 20 and 21). Fortunately, in the course of time, the numbering of residues is starting to follow that of the sequences (or their isoforms) in the UNIPROT database. However, currently, improving the mapping of the mined residues, especially those from the older literature, to structures needed for docking would require complicated analysis of the textual context, which is outside the scope of this study. Also, the number of the full-text articles in the open access, from which the residues could be mined, is still relatively small. With the rapid growth of popularity of the open access publishing (Piwowar *et al.*, 2018), the increase in the success rate should grow as well. Thus, the utility of the ML approaches in application to protein docking, the basic principles of which we explored in this study, will expand accordingly, leading to more accurate modeling of protein interactions.

4 Conclusion

We continued development of the methodology for generating constraints from publicly available literature for application to structural modeling of protein complexes. Capitalizing on our earlier results on generating such constraints from the basic TM of PubMed abstracts (Badal *et al.*, 2015), improved by natural language processing techniques (Badal *et al.*, 2018), in this study, we focused on comparing performances of ML models generated by two methods (Deep Recursive Neural Network and Support Vector Machine) in filtering non-interface residues from the same list of initially mined residues either in PubMed abstracts or PMC-OA subset of freely available full text articles.

The PMC-OA full text articles, despite representing a small subset of scientific publications, provide a useful source for training of DRNN model. The networks can be applied to classification of residues found in the abstracts, where the sentence structures are, in general, different from those in the full-text articles. In such case, the DRNN model is superior to SVM model, because the success of the latter is often determined by the similarity in data/text patterns in the training and the testing sets. Our study provides an insight into the optimal context size for TM applications, based on the significant improvement of the DRNN model's performance when the sentiment was calculated for a part of the sentence around the mined residue rather than for the entire sentence. The results indicate that the bank of sentiment trees, specific for protein–protein interactions and curated by the experts in the field, is essential for further performance improvement of the ML-enhanced text-mining. Overall, following our previous results on NLP application to abstracts (Badal *et al.*, 2018), we showed that DRNN model similarly outperform the basic TM on the abstracts, and both SVM and DRNN models outperform the basic TM when applied to the full-text papers. Greater availability of the full-text papers should increase usefulness of this source of information for structural modeling of protein complexes. A simpler SVM approach is often sufficient for filtering residues mined from the texts with patterns similar to those used for training (abstracts—abstracts or full texts—full texts). Thus, the use of DL approaches (which are computationally demanding, especially, at the training stage) in such cases may be unnecessary.

By its nature, the approach based on full-text articles depends on the pool of published open access articles about the target protein complex. Thus, naturally its application to the recent challenging targets is still limited. However, with the growth of popularity of the open access publishing, the utility of the approach will grow. Our study focused on applicability of the TM to modeling of protein complexes, in anticipation of the inevitable future growth of the open access publishing.

Funding

This study was supported by NIH [R01GM074255] and NSF [DBI1565107 and DBI1917263].

Conflict of Interest: none declared.

References

Badal,V.D. *et al.* (2015) Text mining for protein docking. *PLoS Comput. Biol.*, **11**, e1004630.

Badal,V.D. *et al.* (2018) Natural language processing in text mining for structural modeling of protein complexes. *BMC Bioinformatics*, **19**, 84.

Bengio,Y. *et al.* (2013) Representation learning: a review and new perspectives. *IEEE Trans. Patt. Anal. Mach. Intell.*, **35**, 1798–1828.

Brants,T. *et al.* (2007) Large language models in machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Prague, Czech Republic. pp. 858–867.

Caporaso,J.G. *et al.* (2008) Intrinsic evaluation of text mining tools may not predict performance on realistic tasks. In *Pacific Symposium Biocomputing*. NIH Public Access. Big Island, HI. World Scientific. pp. 640–651.

Caufield,J.H. and Ping,P. (2019) New advances in extracting and learning from protein–protein interactions within unstructured biomedical text data. *Emerg. Top. Life Sci.*, **3**, 357–369.

Ching,T. *et al.* (2018) Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface*, **15**, 20170387.

Cohen,A.M. and Hersh,W.R. (2005) A survey of current work in biomedical text mining. *Brief. Bioinf.*, **6**, 57–71.

Cohen,K.B. *et al.* (2010) The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, **11**, 492.

Collobert,R. and Weston,J. (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*. ACM. pp. 160–167.

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

Dauzhenka,T. *et al.* (2018) Computational feasibility of an exhaustive search of side-chain conformations in protein–protein docking. *J. Comput. Chem.*, **39**, 2012–2021.

De Marneffe,M.-C. and Manning,C.D. (2008a) Stanford typed dependencies manual. *Technical report*, Stanford University. pp. 338–345.

De Marneffe,M.C. and Manning,C.D. (2008b) The Stanford typed dependencies representation. In *Proceeding of the Workshop Cross-Framework Cross-Domain Parser Evaluation. Association for Computational Linguistics, Manchester, UK*. pp. 1–8.

Dogan,R.I. *et al.* (2017) The BioC-BioGRID corpus: full text articles annotated for curation of protein–protein and genetic interactions. *Database*, **2017, baw147**

Fink,J.L. *et al.* (2008) BioLit: integrating biological literature with databases. *Nucleic Acids Res.*, **36**, W385–W389.

Friedman,C. *et al.* (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, **17**, S74–S82.

Gerner,M. *et al.* (2010) An exploration of mining gene expression mentions and their anatomical locations from biomedical text. In *Proceedings of the 2010 Workshop Biomedical Natural Language Processing*. Association for Computational Linguistics. pp. 72–80.

Gerner,M. *et al.* (2012) BioContext: an integrated text mining system for large-scale extraction and contextualization of biomolecular events. *Bioinformatics*, **28**, 2154–2161.

Habibi,M. *et al.* (2017) Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, **33**, I37–I48.

Hakenberg,J. *et al.* (2010) Efficient extraction of protein–protein interactions from full-text articles. *IEEE-ACM Trans. Comput. Biol. Bioinf.*, **7**, 481–494.

Huang,M. *et al.* (2004) Discovering patterns to extract protein–protein interactions from full texts. *Bioinformatics*, **20**, 3604–3612.

Hunjan,J. *et al.* (2008) The size of the intermolecular energy funnel in protein–protein interactions. *Proteins*, **72**, 344–352.

Irsoy,O. and Cardie,C. (2014a) Deep recursive neural networks for compositionality in language. In: Advances Neural Information Processing Systems, pp. 2096–2104.

Irsoy,O. and Cardie,C. (2014b) Modeling compositionality with multiplicative recurrent neural networks. arXiv: 1412.6577.

Joachims,T. (1998) Text categorization with support vector machines: learning with many relevant features. In: Nedellec,C. and Rouveirol,C. (eds.) *Machine Learning: ECML-98*. Springer, Berlin, Heidelberg. pp. 137–142.

Joachims,T. (1999) Making large-scale support vector machine learning practical. In: *Advances in Kernel Methods*. MIT Press. pp. 169–184.

Jurafsky,D. and Martin,J.H. (2017) Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson, Upper Saddle River, NJ.

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

Kundrotas,P.J. *et al.* (2018) Dockground: a comprehensive data resource for modeling of protein complexes. *Protein Sci.*, **27**, 172–181.

Lan,M. and Su,J. (2010) Empirical investigations into full-text protein interaction Article Categorization Task (ACT) in the BioCreative II. 5 Challenge. *IEEE/ACM Trans. Comput. Biol. Bioinf. (TCBB)*, **7**, 421–427.

LeCun,Y. *et al.* (2015) Deep learning. *Nature*, **521**, 436–444.

Li,A. *et al.* (2016) A text feature-based approach for literature mining of lncRNA–protein interactions. *Neurocomputing*, **206**, 73–80.

Lin,J. (2009) Is searching full text more effective than searching abstracts? *BMC Bioinformatics*, **10**, 46.

Mallory,E.K. *et al.* (2015) Large-scale extraction of gene interactions from full-text literature using DeepDive. *Bioinformatics*, **32**, 106–113.

Martin,E.P. *et al.* (2004) Analysis of protein/protein interactions through biomedical literature: text mining of abstracts vs. text mining of full text articles. In: *Knowledge Exploration in Life Science Informatics*. Springer. pp. 96–108.

McIntosh,T. and Curran,J.R. (2009) Challenges for automatically extracting molecular interactions from full-text articles. *BMC Bioinformatics*, **10**, 311.

Mikolov,T. (2012) Statistical language models based on neural networks. PhD Thesis, Brno University of Technology.

Mikolov,T. *et al.* (2013a) Efficient estimation of word representations in vector space. *arXiv: 1301.3781*.

Mikolov,T. *et al.* (2013b) Distributed representations of words and phrases and their compositionality. In: *Advances Neural Information Processing Systems*, pp. 3111–3119.

Mikolov,T. *et al.* (2013c) Linguistic regularities in continuous space word representations. In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. pp. 746– 751.

Morik,K. *et al.* (1999) Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring (No. 1999, 24). *Technical report, SFB 475*, Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

NCBI Resource Coordinators. (2013) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **41**, D8.

Papanikolaou,N.L. *et al.* (2015) Protein–protein interaction predictions using text mining methods. *Methods*, **74**, 47–53.

Peng,Y. *et al.* (2016) BioC-compatible full-text passage detection for protein–-protein interactions using extended dependency graph. *Database*, **2016**, baw072.

Pennington,J. *et al.* (2014) Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference Empirical Methods in Natural Language Processing (EMNLP)*. pp. 1532–1543.

Piwowar,H. *et al.* (2018) The state of OA: a large-scale analysis of the prevalence and impact of Open Access articles. *PeerJ*, **6**, e4375.

Raja,K. *et al.* (2020) Automated extraction and visualization of protein–protein interaction networks and beyond: a text-mining protocol. *Methods Mol. Biol. (Clifton, N.J.)*, **2074**, 13–34.

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

Rumelhart,D. *et al.* (1986) Learning representations by back-propagating errors. *Nature*, **323**, 533–538.

Schuemie,M.J. *et al.* (2004) Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, **20**, 2597–2604.

Schwenk,H. (2007) Continuous space language models. *Comput. Speech Lang.*, **21**, 492–518.

Shah,P.K. *et al.* (2003) Information extraction from full text scientific articles: where are the keywords? *BMC Bioinformatics*, **4**, 20.

Socher,R. *et al.* (2011a) Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. pp. 129–136.

Socher,R. *et al.* (2011b) Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the. Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. pp. 151–161.

Socher,R. *et al.* (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Citeseer. p. 1642.

Tagore,S. *et al.* (2019) ProtFus: a comprehensive method characterizing protein–protein interactions of fusion proteins. *PLoS Comput. Biol.*, **15**, e1007239.

Turney,P.D. (2013) Distributional semantics beyond words: supervised learning of analogy and paraphrase. *Trans. Assoc. Comput. Linguist. (TACL)*, **1**, 353–366.

Vakser,I.A. (1998) Low-resolution docking: prediction of complexes for underdetermined structures. *Biopolymers*, **39**, 455–464.

Vakser,I.A. (2014) Protein–protein docking: from interaction to interactome. *Biophys. J.*, **107**, 1785–1793.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

Westergaard,D. *et al.* (2018) A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts. *PLoS Comput. Biol.*, **14**, e1005962.

Weston,J. *et al.* (2011) Wsabie: scaling up to large vocabulary image annotation. In: *IJCAI*. pp. 2764–2770.

Yao,Y. *et al.* (2019) An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, **7**, e7126.

Yu,K. *et al.* (2018) Automatic extraction of protein–protein interactions using grammatical relationship graph. *BMC Med. Inf. Decis. Mak.*, **18**.