Text Mining for Protein-Protein Docking

By

Varsha Dave Badal

Submitted to the graduate degree program in the Center for Computational Biology and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Date Defended: 3 July 2018

The dissertation committee for Varsha Dave Badal certifies that this is the approved version of the following dissertation:

Text Mining for Protein-Protein Docking

Chair: Ilya A. Vakser, Ph.D.

Co-Chair: Petras J. Kundrotas, Ph.D.

Date Approved: 3 July 2018

Abstract

Scientific publications are a rich but underutilized source of structural and functional information on proteins and protein interactions. Although scientific literature is intended for human audience, text mining makes it amenable to algorithmic processing. It can focus on extracting information relevant to protein binding modes, providing specific residues that are likely be at the binding site for a given pair of proteins. The knowledge of such residues is a powerful guide for the structural modeling of protein-protein complexes. This work combines and extends two well-established areas of research: the non-structural identification of protein-protein interactors, and structure-based detection of functional (small-ligand) sites on proteins. Text-mining based constraints for protein-protein docking is a unique research direction, which has not been explored prior to this study. Although text mining by itself is unlikely to produce docked models, it is useful in scoring of the docking predictions. Our results show that despite presence of false positives, text mining significantly improves the docking quality. To purge false positives in the mined residues, along with the basic text-mining, this work explores enhanced text mining techniques, using various language processing tools, from simple dictionaries, to WordNet (a generic word ontology), parse trees, word vectors and deep recursive neural networks. The results significantly increase confidence in the generated docking constraints and provide guidelines for the future development of this modeling approach. With the rapid growth of the body of publicly available biomedical literature, and new evolving text-mining methodologies, the approach will become more powerful and adequate to the needs of biomedical community.

Acknowledgments

I wish to express my gratitude towards several people who motivated, guided, encouraged and critiqued me during my years as a graduate student at the Center of Computational Biology, University of Kansas.

First and foremost, I would like to thank my advisor, Professor Ilya Vakser, whom I had first met as an undergraduate many years back and whose work first introduced me to the exciting field of proteins and their interactions. He always possessed a keen sense of where the field is headed. A text mining tool to generate constraints for protein docking is his brainchild. He encouraged and guided me throughout my years at graduate school not only on subject at hand but also fostered the spirit to be on a constant lookout for what is to come, by introducing me to the ISMB, MPI conferences and the associated community of scientists.

I am extremely grateful to Dr. Petras Kundrotas, my co-adviser, for instilling in me the attention to detail and the rigors of science. It is impossible to describe the many ways he has groomed me into a scientist. I thank my advisors for investing in me the countless hours preparing me and providing me feedback and encouragement all along.

I thank all the individuals serving on my thesis defense committee: Dr. Eric Deeds, Dr. Christian Ray, Dr. Joanna Slusky, Dr. Yinglong Miao and Dr. Krzysztof Kuczera for their time and efforts. I also thank Dr. Wonpil Im, Dr. John Karanicolas for their time and comments for my pre-doctoral examination. The faculty has periodically helped me gain new insights and perspectives during the course work and weekly seminars.

I thank all the former and current members of Vakser lab including graduate students Dr. Ivan Anishchenko, Saveliy Belkin, Nathan Jenkins, Ian Kotthoff and all the postdoctoral research fellows for having interesting discussions, maintaining friendly creative and respectful

environment. I thank all the members of Center of Bioinformatics including Debbie Douglass-Metsker, Matthew Copeland for providing the administrative and technological support during my graduate studies.

I thank my parents, Sheela & V.K. Dave, my in-laws, Sushma & A.K. Badal and my sister Ritu Sharma who have always encouraged and supported my personal endeavors.

I am grateful to my husband and children, Sumit Badal, Jai D. Badal and Rohan D. Badal for bringing me joy, providing me unconditional support and accommodating their time and schedule for my research and studies. My husband and my kids have been my source of stability and comfort.

Table of Contents

List of Figures

List of Tables

List of Text

Chapter 1

Introduction

Protein-protein interactions (PPI) are an integral part of life processes. The function of a protein is largely determined by its structure. Thus, structural characterization of protein-protein complexes is important for understanding and manipulating molecular mechanisms in living systems. Because of the inherent limitations of the experimental approaches, only a fraction of protein-protein complexes have their structure determined experimentally. This makes computational modeling indispensable for large-scale structure characterization of PPI [1].

Structural modeling of protein-protein complexes (protein docking) predicts the structure of the protein complex from the structures of the individual proteins. Docking typically yields multiple candidate models requiring further assessment/scoring [2]. Such scoring can significantly benefit from constraints (cues) for elimination of non-native models that are not distinguishable based on energy, shape, and other common docking characteristics. An exponentially growing number of research publications on protein structure and function is a rich material for PPI cue generation (**Figure 1-1**). Text mining (TM) is a well-established field for automated extraction of information from publications [3, 4] It has been widely used for non-structural prediction of protein interactors [5-7], and structural determination of functional (small ligand) sites on proteins [8, 9]. Prior to our study, it had not been used in protein docking. Critical Assessment of Prediction of Interactions (CAPRI) is a community wide challenge to assess performance of the protein docking efforts [10]. The submitted blind predictions are compared to unpublished experimentally determined structures, providing an objective assessment. The cues generated using our text-mining tools have been successfully utilized in the effort since 2014. Systematic evaluation of the

docking approaches with our text mining-generated constraints on comprehensive benchmark sets of protein-protein complexes showed a significant improvement of the docking success rates.

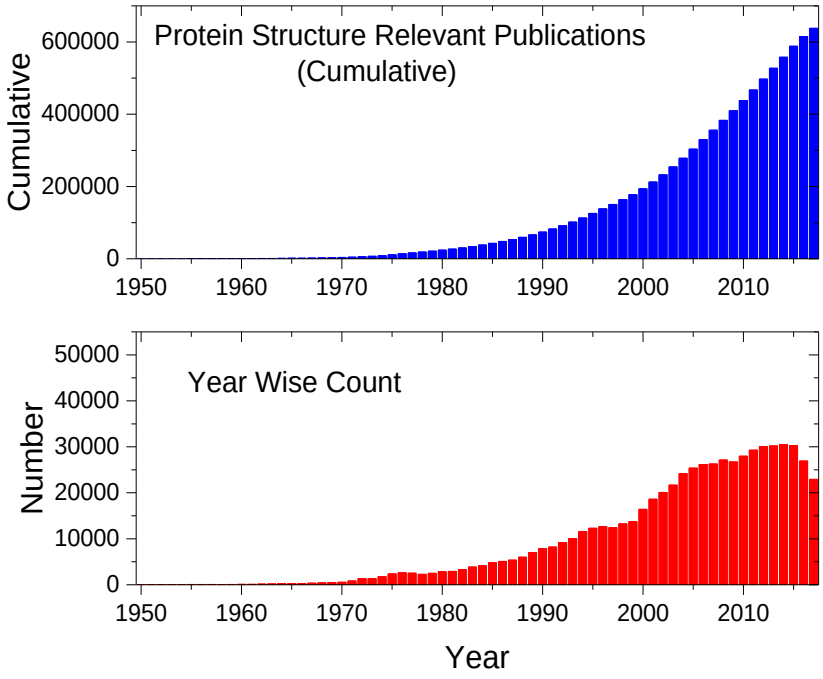


Figure 1-1: *The growth of the number of articles in PubMed (http://dan.corlan.net/medline-trend.html).*

1.1 An overview of protein docking

Early protein docking approaches focused on the global search of the rigid-body complementarity of the two proteins, typically assuming no additional available information other than the structure of the participating proteins [11]. These *ab initio* or free docking approaches, still popular in the field, take advantage of the proteins structural complementarity, as well as complementarity of the physicochemical characteristics - electrostatics, hydrogen bonding, hydrophobicity, and such.

Following the rapid growth of experimentally derived data on protein interactions, the docking techniques have been increasingly incorporating data-driven approaches, such as knowledge-based potentials [12], and binding site characteristics [13]. The ultimate data-driven docking techniques,

based on similarity to the experimentally determined complexes of similar proteins, has become known as template-based or comparative docking [14].

1.2 Stages in protein docking

Typical protein docking routine consists of three (sometimes partially overlapping) stages: (*i*) global rigid-body scan, (*ii*) scoring and (*iii*) local structural refinement.

The scan stage in free docking approaches systematically samples possible mutual orientations obtained by relative translation and rotation and ranks them according to the quality of the fit. In template-based docking, the global scan search is for matching the sequence and/or the structure to the pool of template complexes, and assembly of the target proteins according to the matching templates. Scoring re-ranks the tentative complexes predicted by either free or template-based scan, based on a variety of information, including energetics, statistics, and such. Refinement stage enhances structural details in the top ranked models. Despite the difference in the nature of structural inaccuracies of the free and comparative docking, similar refinement techniques are applicable to both [15]. Refinement typically involves high resolution rigid-body adjustment, side chain repacking, and backbone conformational changes.

1.3 Free docking scan

The basic shape matching is often augmented by physics or knowledge-based potentials to better discriminate the candidate models. Physics-based potentials (e.g. electrostatic, hydrophobic, and such) [16-18] focus on the underlying physiochemical principles. Statistical potentials rely on deviation from a random distribution of residue or atom pairings [19, 20].

The main approaches to the free global scan are: (*i*) correlation by Fast Fourier Transform (FFT docking) performing efficient exhaustive sampling of the discrete search space – an idea borrowed from pattern recognition [21], (*ii*) geometric hashing, matching a list of surface features or motifs described as a hash [22], and (iii) Monte Carlo protocols relying upon randomized sampling [23].

Some notable FFT docking programs are GRAMM [24], ZDOCK [25], PIPER [19], and HEX[26, 27]. GRAMM focuses on the fast, low-resolution geometric matching, which tolerates structural inaccuracies, and serves as the starting point for detailed modeling. ZDOCK and PIPER integrate pairwise statistical potential with the FFT based geometric fitting. HEX uses spherical harmonics to speed-up the FFT-based search.

RosettaDock [23] implements a low-resolution Monte Carlo search with simultaneous optimization of the backbone displacement (otherwise usually performed at the final stage). Some programs emphasize integration of experimental information in the docking process. HADDOCK [28, 29] integrates biochemical and biophysical interaction data (chemical shift perturbation) with ambiguous distance information from NMR studies.

1.4 Template-based docking

Sometimes called docking by analogy, the template-based (or comparative) docking paradigm follows template-based methodologies of the individual proteins modeling. The structure prediction of the individual proteins, historically, started as *ab initio*. However, along with accumulation of the experimentally determined structures, its focus gradually shifted to template-based approaches. A similar transition is underway in the protein docking field. The template-based docking scan hinges on the observation that similar protein pairs form similar complexes.

Sequences of the target proteins are aligned with sequences of the putative templates, with a good match implying structural correspondence. Since this match is obtained by sequence alignment, the process is called sequence homology-based structure prediction [30].

In the absence of strong sequence similarity, the homology paradigm is still applicable because the structure is more conserved than the sequence. Evidence suggests that structure-based templates are available to model nearly all protein-protein complexes [14]. The full structure alignment technique aligns target proteins with the entire structure of the template complexes. The partial (interface) structure alignment approach aligns the target protein only with the co-crystalized interfaces of the template complexes [31-33].

1.5 Refinement

Proteins undergo conformational change upon binding. Thus, to accurately model the atomic details of the protein-protein interface, the rigid-body global scan requires structural refinement that includes conformational search of the predicted interface area [11]. Side-chain repacking is aided by exploring alternative rotational isomers configurations (rotamer libraries). Advanced docking approaches also include the backbone flexibility. The backbone refinement, for the interface and the surrounding area, or for the entire complex [23] can be achieved by structure perturbation or by maximizing Ramachandran probability. For the backbone refinement, HADDOCK performs molecular dynamics-based simulated annealing. One popular approach to the structural refinement is normal mode analysis where the docked complex is modeled as elastic network, with perturbations generating oscillations at fixed frequencies (modes). HexDock performs pose dependent analysis with only a few modes [34], and SwarmDock allows linear

combination of the modes [35]. RosettaDock refinement is based on a tree-like representation of molecular structure allowing simultaneous rigid-body, side-chain, and backbone refinement [36].

1.6 Text mining for PPI

Because of the complexity of the docking search space, constraints on the possible docking solutions are of great value for the predictive protocols. Such constraints implemented at the scoring stage re-rank the docking output, improving discrimination of the false-positive solutions. The focus of this study is on exploring generation of such constraints from the text mining of biomedical literature (**Figure 1-2**).

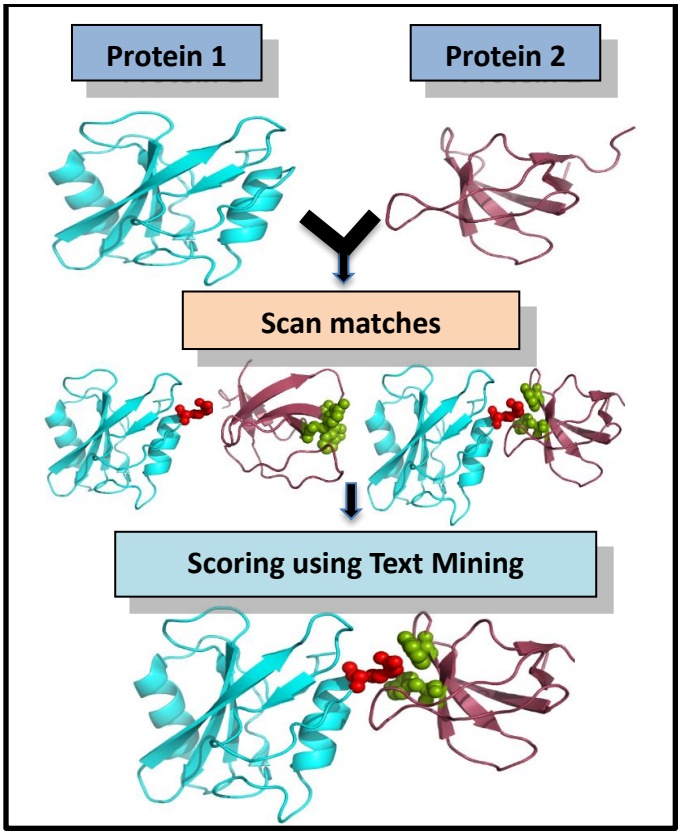


Figure 1-2: *Cues generated by text mining are used at the scoring stage to re-rank docking output.*

Text mining has been used in non-structural approaches to recreating the networks of protein-protein interactions, and in structure-based detection of functional (small molecule) binding sites on proteins. We synergistically use these two areas of research for the structural modeling of protein-protein complexes. Our procedure retrieves published articles on the protein-protein interaction and extracts the relevant residues.

The text mining systems retrieve relevant text from the repository (Information Retrieval, IR), and attempts to extract meaningful information (Information Extraction, IE). Part of Speech Tagging (POS) and Named Entity Recognition (NER) help identify concepts and objects of interest in the Natural language Processing (NLP) systems. POS tagger attempts to assign a part of speech tag to every word encountered during the textual analysis. NER focuses on nouns (or Named Entities) and the kinds of things they refer to. Maintaining a simple dictionary may suffice in many cases. Since our entities of interest are residues, each a combination of an amino acid name and a number, they are easily identified using regular expressions [37].

1.7 Summary

Incorporating mined cues from publications at the scoring stage allows one to discriminate false-positive poses from the docking scan. The primary aim of this study has been to explore the effectiveness of the text mining for a mainstream docking approach.

Chapter 2 investigates generation of docking constraints by basic text mining techniques that rely on co-occurrence as the source of confidence in the prediction. The Chapter is a reprint of Badal VD, Kundrotas PJ and Vakser IA, Text Mining for Protein Docking. *PLoS Comput Biol*, 2015, 11: e1004630. Supporting information for this Chapter is in Appendix A.

Chapter 3 investigates the use of sophisticated NLP techniques, starting with the distance measures based on generic word ontologies (WordNet) and dictionaries of various sizes. Syntactic (grammatical) information from the parse trees of embedding and surrounding sentences as context and Support Vector Machines (SVM) are utilized to purge irrelevant residues. The Chapter is a reprint of Badal VD, Kundrotas PJ, Vakser IA: Natural language processing in text mining for structural modeling of protein complexes**.** *BMC Bioinformatics* 2018, 19: 84. Supporting information for this Chapter is in Appendix B.

Chapter 4 explores recent developments in deep learning (DL) for language composition, including deep recursive neural networks (DRNN). Distributed representation of words and the binary parse tree are the input to a DRNN that learns word composition and performs sentiment analysis. We explore effectiveness of the basic and enhanced (DL and NLP) text mining on publication abstracts and the full text papers. The Chapter is a reprint of Badal VD, Kundrotas PJ, Vakser IA, Enhanced text mining of biomedical literature for modeling of protein complexes. *Submitted*. Supporting information for this Chapter is in Appendix C.

Chapter 2

Text Mining for Protein Docking

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

[1]Center for Computational Biology and [2]Department of Molecular Biosciences, The University of Kansas, Lawrence, Kansas 66047, USA

PLoS Comp Biol. 2015; 11:e1004630.

2.1 Introduction

The rapidly growing amount of publicly available information from biomedical research is a modern day phenomena that is likely to continue and accelerate in the future. Most of this information is readily accessible on the Internet, providing a powerful resource for predictive biomolecular modeling. The accumulated data revolutionized structure prediction of proteins in the 80's [38] and, recently, of protein complexes [39-41] due to the growth of Protein Data Bank (PDB) [42], providing enough structural "templates" for the prediction targets. Instead of painstaking and generally unreliable exploration of the enormous search space, based on the physical "first principles," nowadays tools can simply proceed to the solution based on similarity to the existing, previously determined structures.

In our opinion, the next stage of this revolution is brewing due to the rapidly expanding amount of information, other than experimentally determined structures, which still can be used as constraints in biomolecular structure prediction [43]. In this paper we present the first, to our knowledge, approach to structural modeling of protein-protein (PP) complexes (protein docking), based on the input from automated text mining (TM) of publications on the Internet.

Protein-protein interactions (PPI) are central for many cellular processes. Structural characterization of PPI is essential for fundamental understanding of life processes and applications in biology and medicine. Because of the inherent limitations of experimental techniques and rapid development of computational power and methodology, protein docking is a tool of choice in many studies. One of the main problems in protein docking [11] is identification of a near-native match among the large, often overwhelming, number of putative matches produced by a global docking scan. To detect the near-native matches at the docking post-processing stage, a scoring procedure is performed by re-ranking of the scan output matches,

typically using energy/scoring functions, which are too computationally expensive or impossible/impractical to include in the global search. Such scoring schemes may be based on structural, physicochemical, or evolutionary considerations [44]. For some PPI, information on the docking mode (e.g. one or more residues at the PP interfaces) is available prior to the docking. If this information is certain, there is no need for the docking global scan, and the search can be performed in the sub-space that satisfies the constraints. However, if the probability of such information is <100%, it may rather be included in the post-processing of the global scan, as part of the scoring.

Given the inherent uncertainties of the global-search docking predictions, such independent information on the binding modes is extremely valuable [45]. Such information may be available on the case-by-case basis. However, for docking server predictions that can be used by the broad biological community an automated search for such information can be of great value.

The PPI research is an extremely active field, yielding a vast amount of publications on interacting proteins [46]. These publications quickly become available online (e.g. through PubMed, http://www.ncbi.nlm.nih.gov/pubmed), and are a growing resource for automated mining of the PP binding mode. Many applications (PubMed ENTREZ, NLProt, MedMiner, etc.) utilizing TM techniques have been developed to improve access to the published knowledge [47]. TM converts textual information into database content and complex networks, facilitating development of novel working hypothesis [4]. In biology, TM tools have been used to mine generic or specific information on genes, proteins and their functional relationships. Natural Language Processing (NLP) and Support Vector Machines (SVM) have been used to extract information on connection between proteins in PPI networks [3, 5, 48-52]. Along with the networks of interacting proteins, TM tools have been used to generate a dataset of non-interacting proteins [53]. Full-text

articles on metabolic reactions and mutation impacts have been mined by rule-based parser, pattern matching, and entity taggers (protein and gene names along with specific keywords) [54, 55]. Automated TM procedure was developed to extract subcellular localization and function of proteins [56]. TM in combination with the dynamic perturbation analysis has been employed to increase confidence in predicted protein functional site [57], suggesting that if a residue is mentioned in an abstract on the protein structure, it is likely to be in the functional site.

TM approaches are also implemented in many Web-based applications. There are different TM tools for identification of interacting proteins from biological literature and databases [58]. GENIA corpus (a collection of semantically annotated documents) has been specifically designed for testing NLP approaches [59]. PESCADOR extracts a network of interactions from a user-provided set of PubMed abstracts [60]. LAITOR can further filter this mined interactome according to the specific user needs [61]. CRAB extracts data from MEDLINE abstracts, which are relevant to tumor-related chemicals posing risk to human health [62]. PIE utilizes word and syntactic features to effectively capture PPI patterns from biomedical literature [7, 63]. eFIP mines information on phosphorylation and related interactions of a given protein using rule-based NLP [64]. PPInterFinder extracts Medline abstracts on human proteins using co-occurrences of protein names, specific keyword dictionary, and pattern matching [65]. BioQRator can annotate PPI-relevant entity relationships from the biomedical publications [66].

In this paper, we propose the first, to our knowledge, approach to TM constraints for PP docking. Our methodology, by design, is a combination and expansion of two well-developed TM fields: (1) identification of interactors in PPI networks, and (2) detection of protein functional (small ligand) sites. We use the first one as the source of expertise on TM of PPI (existing approaches are concerned with the fact of interaction, not the mode of interaction), and the second

one as the source of expertise on TM for structural prediction of the binding sites on proteins (existing approaches are for small non-protein ligands). The method was tested on PubMed abstracts of publications on protein complexes from DOCKGROUND (http://dockground.compbio.ku.edu) and showed a significant improvement of the docking success rates.

2.2 Methods

2.2.1 Text-mining protocol

The principal stages of the TM protocol are shown in **Figure 2-1**. We divide our procedure into two parts, information retrieval (selecting abstracts containing names of both or either proteins in a complex) and information extraction (detecting occurrence of residues in the retrieved abstracts). The abstracts were further filtered by SVM model with optimal sets of features. The TM tool was benchmarked on 579 PP complexes with known bound X-ray structures from Dockground and applied for re-scoring of the initial docking models for 99 protein pairs from the Dockground unbound benchmark set 3.

2.2.2 Information retrieval

Protein name and UniProtKB ID, corresponding to the particular PDB code and protein chain, were obtained from PDB. To simulate the "real case scenario" when the structure of the target protein is unknown, PubMed ID (PMID) of the direct citation (publication describing the X-ray structure of the complex) was extracted from the PDB and the publication excluded from the further consideration. To further test our methodology, we also restricted the analysis to abstracts published prior to the direct citation paper. Using the UniProtKB ID, protein information in XML

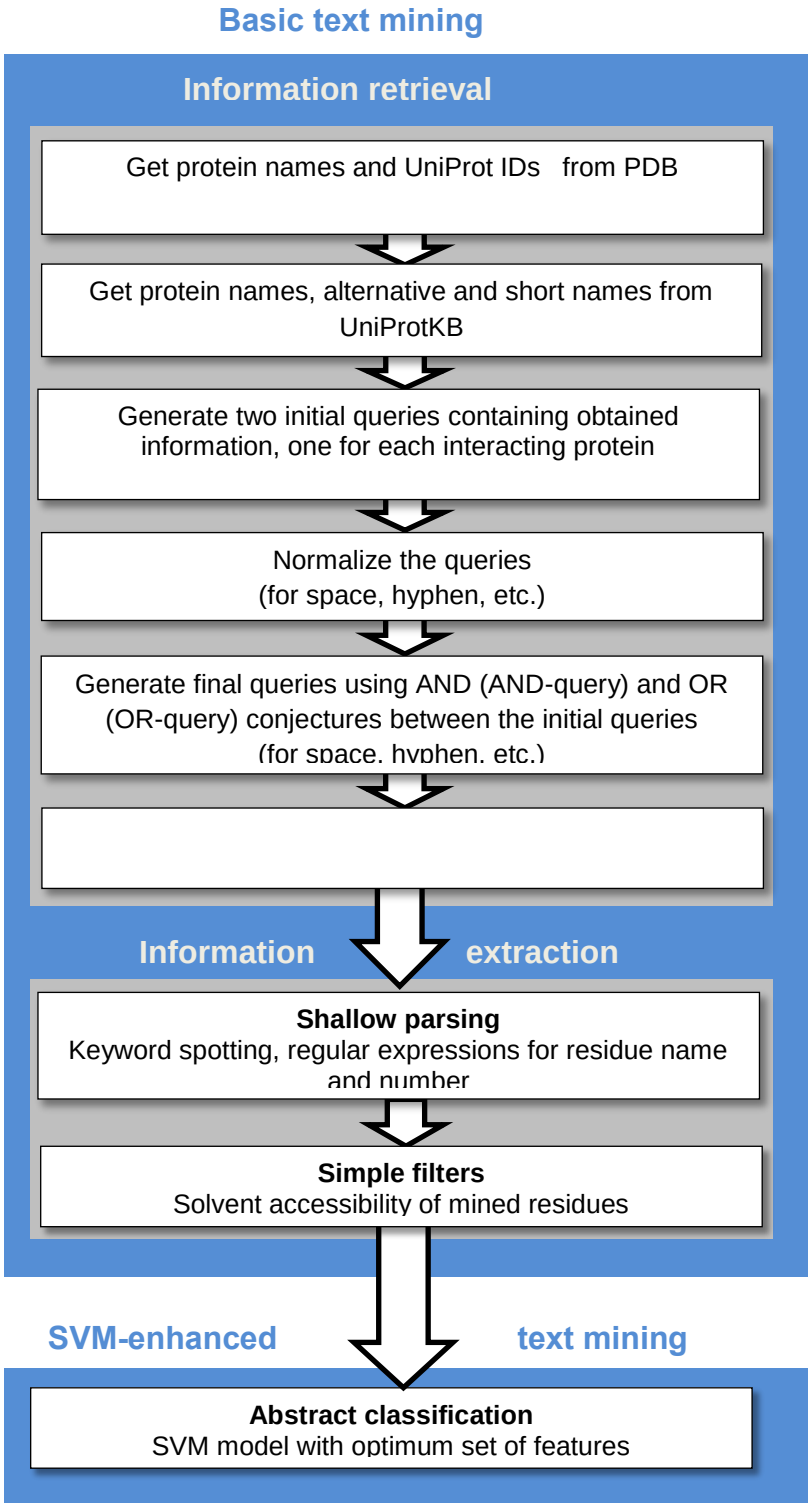


Figure 2-1: *Flowchart of the text mining protocol.*

format was acquired from the UniProtKB [67]. Information from both PDB and UniProtKB was accessed through REST (REpresentational State Transfer) Web Services (http://www.rcsb.org/pdb/software/rest.do), (http://www.uniprot.org/help/programmatic_access). For the query construction, at the current stage, we used only recommended, short and alternative protein names, ignoring organism name, classification of the monoclonal antibodies ("CD_antigen" tag), all parts of gene information (name, synonyms, ordered locus names, and open reading frame), E.C (Enzyme Commission) numbers, as well as UniProtKB terms "Uncharacterized protein." Inclusion of all this additional information into the search queries requires implementation of deep parsers, which is in our plans for the future research.

Protein names were normalized by replacing reserved characters with their URL encodings (spaces replaced by %20, etc.), by removing extra (trailing) spaces, and by hyphen replacement. The query for a protein with a hyphen in its name contains OR-connected versions of the name with hyphen, hyphen removed and replaced by space. For example, query for "IL-15R-alpha" (PDB: 2z3q, chain B) also includes following variations: "IL-15Ralpha", "IL15R-alpha", "IL15Ralpha", "IL%2015Ralpha", "IL15R%20alpha", and "IL%2015R%20alpha". Short names with < 3 symbols were ignored and additional AND-connected keyword "protein" was added to the 3-symbols names.

For 13 PP complexes in the set, protein names coincided with generic, frequently used words ("act" for 1yrt, chain A, UniProtKB: P0DKX7 or "hot" for 2ido, chain B, UniProtKB: Q71T70). To reduce noise, search queries for such complexes contained also MESH terms [68] (combinations of the MESH terms under heading "Biochemical Phenomena").

For 87 proteins, UniProtKB had section "Cleaved into the following X chains" referring to different domains. In such cases, we considered several scenarios. In the case of exact match

between PDB and UniProtKB recommended protein names (29 proteins), we assumed that the PDB structure comprised all the domains mentioned in UniProtKB and included into the query OR-connected recommended name for the entire protein and the names of all the domains. For example, X-ray structure for cationic trypsin (PDB 2xtt, chain B, UniProtKB: P00760) contains both domains (Alpha-trypsin chain 1 and Alpha-trypsin chain 2) mentioned in UniProtKB and the PDB name matches exactly the UniProtKB recommended name. If PDB name matched exactly only one of the domain names (17 proteins), than the query included the name of only this domain along with the recommended protein name. For example, "Protease inhibitor SGPI-1" (PDB 2xtt, chain A, UniProtKB: O46162) is the PDB protein name, which matches exactly one of the cleaved components and does not match recommended UniProtKB name "Serine protease inhibitor I/II". If the PDB protein name did not match exactly neither UniProtKB recommended name, nor any domain names (41 proteins), we considered only the recommended and the PDB names. For example, "epithelial-cadherin" is the PDB protein name (PDB 2omz, chain B, UniProtKB: P12830), which is not the same as the recommended UniProtKB name "Cadherin-1" and none of the cleaved components ("E-Cad/CTF1", "E-Cad/CTF2", "E-Cad/CTF3"). String comparison was performed using Perl module Text::Levenshtein, which implements Levenshtein similarity string matching algorithm (http://search.cpan.org/dist/Text-Levenshtein/lib/Text/Levenshtein.pm).

After constructing two queries, "query1" and "query2", one for each protein in a particular complex, two final queries were assembled: "query1 AND query2*"* (termed here as AND-query) and "query1 OR query2*"* (OR-query). The AND- and OR-queries were submitted to ESearch and EFetch modules of NCBI E-utilities tool [69], (http://www.ncbi.nlm.nih.gov/books/ NBK25501). To keep track on which protein is studied in the retrieved abstracts, two parts of the OR-query

were submitted separately. Maximum of 100,000 PubMed abstracts with publication dates between January 1, 1971 and November 30, 2014 were retrieved for each submitted query.

2.2.3 Information extraction

Abstracts of publications corresponding to 579 complexes, retrieved by the E-utilities from the PubMed (the number of abstracts varies for different types of queries, see Results and Discussion), were searched for the residues using regular expressions (**Table 2-1**) obtained by the manual inspection of 100 abstracts that mention residues. We considered patterns with only three-letter or full residue names, since mining of one-letter residue abbreviations requires deep parsing of the surrounding text, which is beyond the scope of our current study. However, if keywords related to mutagenesis studies ("mutation", "mutagenesis", "mutagen", "mutant", "substitution") were spotted, one-letter abbreviations for mutation (e.g., "S4A") were included in the search patterns. For the mutations, both original and substitution residues were taken (e.g., for the pattern "S4A", both Serine 4 and Alanine 4 were considered as the mined residues).

Since residues participating in docking are on the protein surface, the names and numbers of the extracted residues were checked against the names and numbers of the surface residues from the original PDB file. For the AND-query, the check was performed against both chains of the original complex, whereas for the OR-query, the examination was done only for the protein mentioned in the retrieved abstract (to reduce noise due to the accidental match of the residue name and numbers). Surface residues were defined as those with $\geq 25\%$ of their surface exposed to solvent [70]. The solvent accessible area was calculated by the program surfv [71]. Only the residues with both name and number matching the residues from the original PDB file were considered further (we termed them "identified residues"). In the case of mismatch between PDB and UniProt sequence numbering, we mapped the UniProt sequence on the PDB one as in Ref.

[72]. An identified residue was considered correct if any of its heavy atoms was ≤ 6 Å from any heavy atom of the interacting protein in the co-crystallized complex, which means that the residue is at the PP interface. Performance of the TM protocol for a particular PPI, for which a query extracted *N* abstracts containing residues, was quantified as a fraction of correct (interface) residues among all identified residues

$$P_{\mathrm{TM}} = \frac{\sum_{i=1}^{N} N_i^{\mathrm{int}}}{\sum_{i=1}^{N} \left(N_i^{\mathrm{int}} + N_i^{\mathrm{non}} \right)}, \qquad (2\text{-}1)$$

where N_i^{int} and N_i^{non} are numbers of interface (correct) and non-interface (incorrect) residues in abstract *i*.

Table 2-1: *Regular expressions for amino acids in the information extraction part of the text mining protocol*

Parameter	Value
Number	[1-9][0-9] [a]
Amino acid (AA)	[Ala,…, Val] OR [ala,…, val] OR [ALA,…, VAL] [b]
Three letter residue	AA(no space)Number OR AA(space)Number OR AA – Number **|** AA(Number)
Full AA	[Alanine,...,Valine] OR [alanine,...,valine] [c]
Full word residue	Full_AA(no space)Number OR **|** Full_AA(space)Number OR Full_AA – Number OR Full_AA(Number)
Single AA	[A,....,V] [d]
Single letter mutation	Single_AA(no space)Number(no space)Single_AA
Three letter mutation	AA(no space)Number(no space)AA OR AA-Number(no space)AA

a Non-zero digit followed by any number of digits
b Three-letter abbreviation for standard amino acids
c Full name of amino acid
d One-letter abbreviation for amino acid

2.2.4 Generation of feature sets for SVM models

We generated a set of features by handpicking 60 words from carefully read randomly selected 21 PPI abstracts and 43 non-PPI abstracts (**Table 2-2**). Subsets of 50, 40, 30, 20 and 10 features were also selected based on our understanding of importance of a feature for PPI description. We refer to these sets of features as *manually selected*, abbreviated as MF*xx*, where *xx* is the number of features in the set.

We also generated a set of features by automated counting of words in the abstracts. We refer to this set and all of its subsets as *automatically selected*, abbreviate as AF*xx*, where *xx* is the number of features in the set. For this set, L_{pos} = 450 positive and L_{neg} = 855 negative abstracts, respectively satisfying conditions

$$\begin{aligned} &\left[N_i^{\text{int}} - N_i^{\text{non}} > 4\right] \texttt{OR} \left[N_i^{\text{non}} = 0 \texttt{ AND } N_i^{\text{int}} > 0\right] \quad \text{and} \\ &\left[N_i^{\text{non}} - N_i^{\text{int}} > 4\right] \texttt{OR} \left[N_i^{\text{non}} > 0 \texttt{ AND } N_i^{\text{int}} = 0\right], \end{aligned} \tag{2-2}$$

were selected from 1,523 abstracts retrieved by the AND-queries. Positive and negative abstracts were further randomly split into training (80% of abstracts, or $L_{\text{pos}}^{\text{train}}$ = 360 positive and $L_{\text{neg}}^{\text{train}}$ = 684 negative abstracts) and validation (remaining 20%) sets, and features were selected from the training set.

Specific protein and amino acid names were excluded from the counting, as they were part of the queries in the TM protocol. Stop words (“and”, “as”, “because”, “the”) were also purged from the abstracts. The abstracts were subjected to the tokenizer [73] for the suffix stripping by the Porters stemming algorithm [74] in order to get the stem (root) forms of the remaining abstract words. We slightly modified the original algorithm so that a root of a word would accommodate

wider variability in the spelling of words with the same meaning. For example, words “include” and “inclusion” are counted by the root “inclu-”, words “mutant”, “mutagenesis”, “mutation”, “mutagen”, “mutated”, “mutations”, “mutation” are accounted for by the root “muta-”, etc.

$$f_k(m) = \frac{1}{L_k^{\text{train}}} \sum_{i=1}^{L_k^{\text{train}}} J_i(m), \tag{2-3}$$

The normalized counts for each stem (feature) were calculated separately for the positive (k = pos) and the negative (k = neg) abstracts in the training set

where $J_i(m)$ is the number of times feature m appears in abstract i. All features satisfying conditions

$$\left| f_{\text{pos}}(m) - f_{\text{neg}}(m) \right| > 0.02 \quad \text{and} \quad f_{\text{pos}}(m) + f_{\text{neg}}(m) > 0.2, \tag{2-4}$$

were selected for the full set AF143 (automatically selected 143 features). The criteria are meant to balance a maximal number of features and a strong signal. The full set was sorted based on the ratio

$$\delta(m) = \frac{f_{\text{pos}}(m) - f_{\text{neg}}(m)}{f_{\text{pos}}(m) + f_{\text{neg}}(m)}, \tag{2-5}$$

and consists of 76 PPI-relevant ($\delta(m) > 0$) and 67 PPI-non-relevant ($\delta(m) < 0$) features (**Table 2-2**).

Table 2-2: *Sets of features (stems) for SVM models.*
Manually selected features are sorted alphabetically and automatically selected features are sorted based on the ratio δ (Eq. (2-5) large to small. PPI-relevant features are in bold.

Number of words in a bag	Bag of words
Manual selection	
60	activ, **affin,** alloster, **associ, attach, between, bind, bond, bound**, **catalyt**, **chang**, cleavag, cofactor, **complex, conform**, cooper, conjug, **conserv**, **contact,** cycliz, delet, diminish, **direct, domain**, downstream, enhanc**, enzym**, facilit, growth, increas, **induc**, induct, inhibit**, interact, interfac**, involv, **linkag,** mechan, metabol", modifi, modul, phosphoryl, **positio, preferenti, proxim,** reassoci**, receptor,** recognit, redox, regulatori, **signal, specif, stabil**, stimul**, substrat**, suppress, **surfac**, **target, transform, trigger**
50	affin, alloster, associ, attach, bind, bond, bound, catalyt, chang, cleavag, complex, conform, conserv, cooper, contact, cycliz, delet, diminish, direct, domain, downstream, enhanc, enzym, facilit, growth, increas, induc, inhibit, interact, interfac, involv, linkag, mechan, metabol, modifi, modul, preferenti, reassoci, recognit, regulatori, signal, specif, stabil, stimul, substrat, suppress, surfac, target, transform, trigger.
40	affin, alloster, associ, attach, bind, bond, bound, catalyt, cleavag, complex, conform, conserv, cooper, contact, cycliz, delet, diminish, domain, enhanc, enzym, facilit, increas, induc, inhibit, interact, interfac, linkag, mechan, modifi, modul, preferenti, recognit, regulatori, specif, stabil, substrat, surfac, target, transform, trigger.
30	affin, alloster, associ, attach, bind, bond, bound, cleavag, complex, conform, conserv, contact, cooper, domain, induc, interfac, interact, ,linkag, , mechan, modifi, modul, preferenti, recognit, regulatori specif, stabil, surface, substrat, target, transform
20	alloster, bind, bond, bound, cleavag, complex, conform, contact, conserv, domain, induc, interfac, interact, mechan, modul, preferenti, recognit, specif, stabi, surface.
10	alloster, bind, complex, conform, conserv, contact, induc, interface, interact, recognit.
Automated selection	
143	polymorph, **interfac**, **energi, bond**, **free**, antibodi, **beta**, phenotyp, patient, promot, light, **degre**, **conjug**, gene, **affin**, **label,** diseas, filament, affect, **signal, ca2+, crystal,** properti, **complex, interact,** resist, level, **valu,** detect, membran, contain, **contribut**, inclu, **terminu**, **produc,** genera, regul, shown, examin, transcript, normal, lower, **time**, **base**, **stabil**, express, **critic**, phosphoryl, **subunit**, function, assai, **peptid, catalyt, surfac,** enhanc, investig, **format, positio**, **determin, residu**, **bind, loop,** rate, **effici**, report, factor**, molecul, inhibit**, **prolifer, deriv,** sequenc, alter, singl, mediat, **structur, purifi,** induc, depend, dna, **reveal,** sensit, **receptor**, compar, **model**, **cleavag, via, product**, **target, variant, specif**, loss, **growth**, potenti, **requir,** essenti, caus, **decreas**, low, **substrat,** associ, mechan, **conform**, **fold**, **contrast**, similar, type, **involv**, found, **novel**, region, **exhibit,** wild, vitro, **observ, develop**, **fragment,** famil**, conserv,** cell, **identifi,** stud, reduc, **provid,** demonstr**, acid, data, link,** effect, presenc, activ, **result**, role, domain, **chain**, **enzym, form, alpha, index**, site, increas, suggest, **mutant**, protein

2.2.5 SVM models

We define an SVM model as an SVM classifier with a kernel function, trained and validated with a particular set of features. For the training and validation of the SVM models we used readily available SVMLight [75, 76] with polynomial, $K(X_i, X_j) = (\alpha X_i X_j + C)^d$, and radial-base (RBF), $K(X_i, X_j) = \exp(-\gamma |X_i - X_j|^2)$ kernel functions (X_i and X_j are support and test feature vectors). We tested different values of parameters d and γ while parameters α and C had default values (α = 1 and C = 0), and distinguished a particular case of the polynomial kernel with d = 1, as the linear kernel. We have also investigated how results are affected by varying degree d of polynomial and parameter γ of RBF kernels. Validation of the SVM models was carried out in the classification mode where an abstract was identified as positive or negative depending on the sign of the SVM-score. In some cases, abstracts with SVM scores close to zero (within a margin) were considered as "unclassified" and excluded from the performance evaluation.

All SVM models were trained on 1044 abstracts and validated on different 261 abstracts (see above). Performance of an SVM model was evaluated in usual terms of precision P, recall R, accuracy A [77], and Matthews' correlation coefficient MCC [78]

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad A = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{TN} + \mathrm{FP}},$$

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}, \qquad (2\text{-}6)$$

where TP, FP, TN, and FN are, correspondingly, the numbers of correctly identified positive, incorrectly identified positive, correctly identified negative and incorrectly identified negative abstracts in the validation set.

2.2.6 Docking with text-mining constraints

Basic TM protocol with the OR-queries was used to mine residues for 99 complexes from the DOCKGROUND benchmark set 3 [79], containing the unbound X-ray structures for the co-crystallized complexes (bound structures). Queries for individual proteins and 63 binary complexes were generated as described above. For 36 multimeric complexes, queries were generated using OR-combinations of queries for all monomers in a multimeric chain (e.g., for complex AB: CDE the OR-query was "(queryA OR queryB) OR (queryC OR queryD OR queryE)"). Abstract of publications on the X-ray structure of the co-crystallized complex were excluded from consideration using corresponding PMID from the PDB entry. For validation, the extracted residues were matched to the residues in the bound structures of the dataset (numbering and chain IDs in the bound and the unbound structures is often different). Extracted residues were ranked, in descending order, separately for each interactor (single or multimeric) by the confidence function

$$f(R) = \min\left(10, \sum_{i=1}^{N_R} a_i\right), \qquad (2\text{-}7)$$

where N_R is the total number of distinct abstracts, in which residue R is mentioned, and $a_i = 2$, if abstract i was retrieved by the AND-query and $a_i = 1$, if the abstract was retrieved by the OR-query only. Top five residues for each interactor were used as constraints in our GRAMM docking program [80] giving an extra weight (proportional to $f(R)$) to the scoring function if the identified residue was at the interface of a docking model. The upper limit of 10 in Equation (2-7) was chosen to balance the diversity of low confidence (f = 1) vs. high confidence (f = 10) constraints and potential overrepresentation of a residue in publications (very high f values). If > 5 residues had the highest f values, then preference was given to the residues with scores containing more

contributions from the abstracts retrieved by the AND-queries. Otherwise, the excess residues were removed from the list randomly.

For validation, the residues at the crystallographically determined interface (reference residues) were extracted from the co-crystallized complexes using 6 Å distance cutoff between the heavy atoms of the proteins in the complex. All pairs of these interface residues were ranked in ascending order by the distance between their C^{α} atoms. The top three pairs were submitted to GRAMM with the highest possible confidence score 10 (reference constraints).

The unbound structures were docked by GRAMM once using the TM constraints and then, for comparison, 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). These matches were subjected to scoring by the sum of the *f* values (Equation (2-7)) if constraints were generated for the complex. If no constraints were generated, the score was zero. All matches were then re-sorted according to these scores. The quality of a match was assessed by C^{α} 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 corresponding atoms of the unbound ligand superimposed on the co-crystallized bound structure.

2.3 Results and Discussions

2.3.1 Basic text mining

Overall performance of two query types. The ultimate success of our TM approach relies heavily on the text pool obtained during information retrieval stage (**Figure 2-1**). Queries for mining texts on interactions of two proteins often are generated based on the co-occurrence principle [61], requiring that information on *both* proteins be presented in the abstract of a publication (AND-

query, see Methods). For generating docking constraints, however, it could be desirable to extract a more diverse text set by requiring presence of information on *either* protein (OR-query). This could be especially helpful for proteins that bind several partners at the same interface. However, the "brute-force" use of the OR-queries may also result in many irrelevant abstracts (allosteric sites, substrate preference, signaling and conformational changes, etc.).

To clarify this issue, we have analyzed abstracts for 579 protein complexes from DOCKGROUND retrieved by the AND- and OR-queries. The original publications, describing the PDB structures of the complex, were excluded from consideration. The AND-queries retrieved 220,603 abstracts for 277 complexes; and 18,670 residues (with the names that match features in **Table 2-2**) were extracted from 11,732 abstracts for 193 complexes. In 11,732 abstracts for 193 complexes, residues were detected 18,670 times. The application of the simple filters (see Methods) reduced these numbers to 1,375 residues (identified residues) in 1,660 abstracts for 128 complexes. Of those, 571 residues for 108 complexes were found to be correct (at the PP interfaces). For 21 complexes, all identified residues were correct ($P_{TM} = 1$), and for 20 complexes, all identified residues were outside the interface ($P_{TM} = 0$). The OR-queries retrieved 2,640,816 abstracts for 492 complexes; and 207,931 residues were extracted from 150,060 abstracts for 431 complexes. Residue filtering resulted in 5,781 identified residues in 18,528 abstracts for 328 complexes, out of which 1,919 residues in 273 complexes were correct. All identified residues were correct in 36 complexes, and no interface residues were identified for 55 complexes. All abstracts retrieved by the AND-queries were retrieved by the OR-queries as well.

Comparison of the overall basic TM performance for AND- and OR-queries (first two data rows in **Table 2-3**) suggests significantly higher coverage, with comparable accuracy for the OR–queries. However, as data in **Figure 2-2** indicates, the OR-queries also extracted many irrelevant

abstracts with non-interface residues (bars for the OR-queries with weaker TM performance [smaller P_{TM} values] are larger than the corresponding bars for the AND-queries). For example, for SH2D1A-p59Fyn complex (1m27) AND-query did not retrieve any abstracts, whereas OR-query retrieved 6 abstracts, from which 3 interface and 1 non-interface residues were extracted (**Figure 2-3**; for the detailed description, see **Text A-1**). **Figure A-1** and **Text A-1** provide more examples of the basic TM performance with different P_{TM} and a detailed explanation of what residues were extracted from the abstracts, retrieved by the AND- and OR-queries.

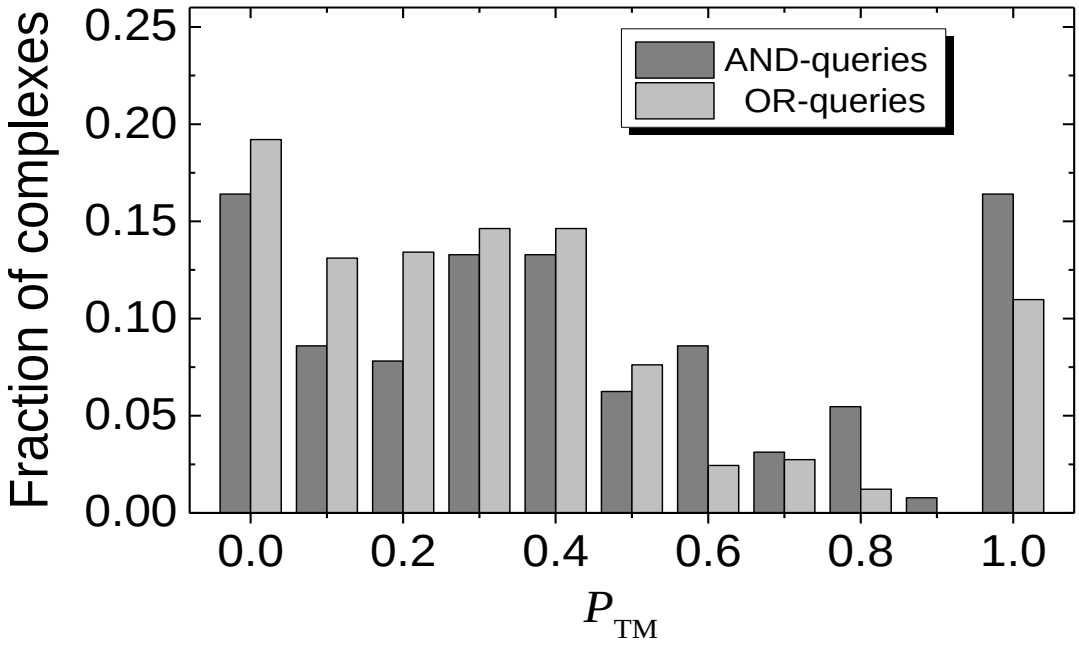


Figure 2-2: *Distribution of complexes according to the quality of the basic TM.*
*The TM performance is according to P_{TM} (Eq.(2-1)). The distribution is normalized to the total number of complexes for which residues were identified (column 3 in **Table 2-3**)).*

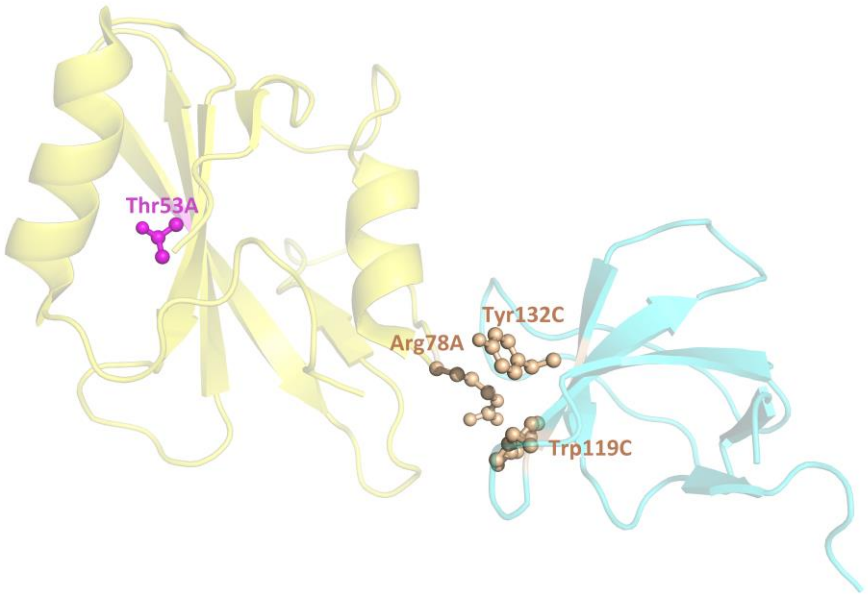


Figure 2-3: *Examples of residues extracted from an abstracts retrieved by OR-query. The structure, chain ID, and residue numbers are from 1m27. Interface and non-interface residues are in brown and magenta, correspondingly.*

Correction for different residue numbering. For 430 out of 1158 monomers, the numbering of residues in PDB files did not match that in the UniProt. For these monomers, we modified the filtering of the initial pool of extracted residues (described above), which resulted in 1,619 identified residues in 2,028 abstracts for 142 complexes, for AND- queries; and 6,735 identified residues in 20,040 abstracts for 342 complexes, for OR- queries. All identified residues were correct for 25 and 31 complexes, and no interface residues were identified for 24 and 59 complexes for AND- and OR-queries, respectively. Analysis of these results (third and fourth data rows in **Table 2-3** and **Figure A-2**) suggests that the numbering mapping only slightly improved TM performance.

Mining of abstracts published before the PDB structure paper. To further test the predictive power of our approach, for each complex, we considered only abstracts with publication date earlier than that of the paper on the PDB structure. This reduced the pool to 84,366 abstracts for 263 complexes, and 1,586,097 abstracts for 487 complexes retrieved by the AND- and OR-queries, respectively. For AND- queries, 7,956 residues were extracted from 3,944 abstracts, and standard residue filtering (see Methods) resulted in 776 identified residues in 814 abstracts for 96 complexes. For OR- queries, 114,472 residues were extracted from 81,418 abstracts, and standard residue filtering resulted in 3,731 identified residues in 9,321 abstracts for 268 complexes. All identified residues were correct for 21 and 29 complexes, and no interface residues were identified for 21 and 66 complexes, for AND- and OR-queries, respectively. The analysis of these TM results (5^{th} and 6^{th} data rows in **Table 2-3** and **Figure A-3**) showed no significant change in the TM performance.

2.3.2 SVM-enhanced text mining

Optimization of SVM models. For the manual mode of feature selection, we considered full MF60 set and five of its subsets, MF50, MF40, MF30, MF20 and MF10 (upper part of **Table 2-2**). For the automated mode of feature selection, we started with the full AF143 set (lower part of **Table 2-2**) and gradually remove features with smallest $|\delta(m)|$ (Eq. (2-5)). For each subset, we trained and tested SVM procedure with several different kernels, with and without a margin for the abstract classification (see Methods).

Introduction of the margin only slightly changes SVM performance (**Figure A-4** and **Figure A-6** for the MF*xx* sets with linear and RBF $\gamma = 1$ kernels, respectively) and filters out considerable number of the abstracts (e.g., for the MF50 set, 13 and 223 abstracts were classified within the 0.05 margin by the linear and the RBF $\gamma = 1$ kernels, respectively). Results for the AF*xx*

sets and other kernels do not show significant change in the performance of SVM with the margin as well (examples of data for the linear kernel are compared in **Figure A-8** and **Figure A-9**). Varying degree *d* of the polynomial (**Figure A-5** and **Figure A-10**) and parameter γ of the RBF (**Figure A-7** and ***Figure* A-11**) kernels for both MF*xx* and AF*xx* sets also did not change the SVM performance significantly. Thus, for simplicity, we present and analyze results only for the linear (polynomial $d = 1$) and RBF $\gamma = 1$ kernels without the margin. Hereafter, we will abbreviate SVM models as AB, where A stands for the feature set (see Methods) and B = L, R for linear or RBF kernels, correspondingly.

In this study, we utilized a binary SVM classifier (abstracts are categorized as either positive or negative). Then, the performance of our SVM models can be quantified by a single measure, Matthew correlation coefficient, MCC (Eq. (2-6)) and the optimum SVM model would have the maximum MCC value [78]. Results, presented in

Figure 2-4, show that three SVM models (MF50L, AF138L, and AF24L) have approximately the same maximum MCC value ~0.25. The AF138L model has the best recall (57.8%), but worst accuracy and precision (64% and 48.1%, respectively), whereas the AF24L model achieved the best accuracy and precision (66.7% and 51.7%, respectively), but the worst recall (51.1%). The MF50L model has all parameters between the AF138L and AF24L models (**Table 2-4**). The variations in the model parameters do not exceed 10% (**Table 2-4**). Thus we kept all three models for further consideration.

No models with the RBF kernel had similar performance, except the AF*xx* sets with $xx < 15$ (**Figure 2-4**). Such small number of features in the SVM model is clearly not enough for statistically reliable results and such models were discarded. As number of features in the model increases, the performance of the RBF kernel deteriorates, in particularly, due to the large amount of abstracts

with the SVM score close to zero (**Figure A-6, Figure A-7, Figure A-11**). This correlates with the conclusion of other studies [81-83] that for the most text categorization problems, the best performance is achieved by the linear separation of feature vectors.

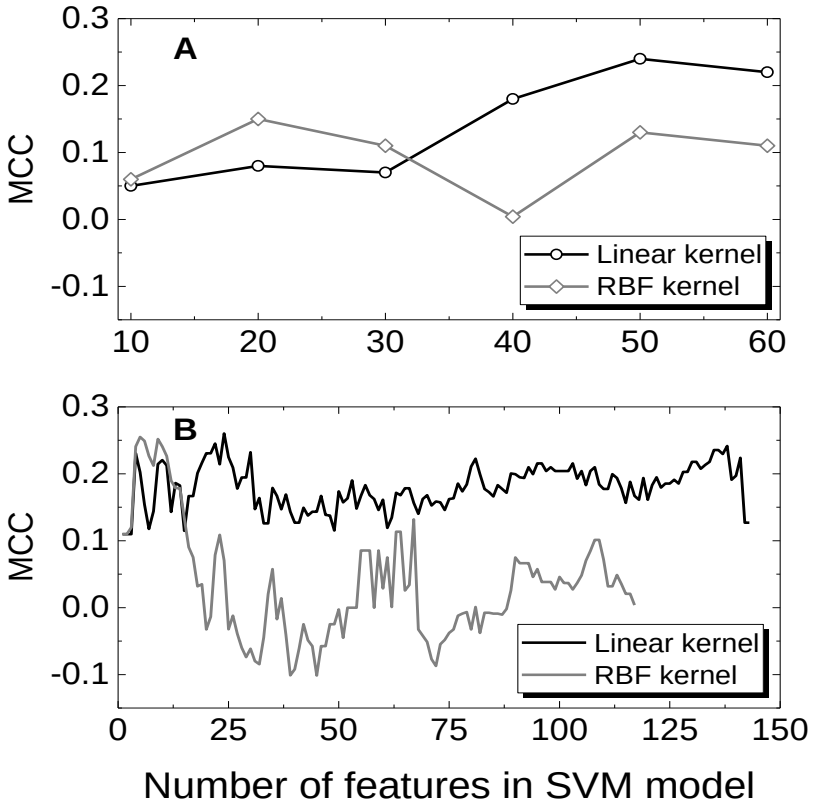


Figure 2-4*:* *Matthews correlation coefficient vs. number of features in SVM model.*
The Matthews correlation coefficient (MCC) is calculated according to Eq. (2-6). The features were selected manually (A) and in automated mode (B), for linear and RBF SVM kernels. The data was obtained on the validation set of 261 abstracts. The SVM models were trained on 1,044 abstracts (see Methods).

Abstract-wise feature selection, irrespective of the number times a feature appears in an abstract, used earlier to extract features for prediction of protein function and localization [56, 84], changes the rank of the initial features. However, the MCC values, calculated for the SVM models

with two different methods of feature selections, are in the same range (**Figure A-12**). Thus significant changes in the SVM performance should not be expected.

Table 2-3*: Performance of basic and SVM-enhanced TM protocols.*
The SVM models were trained and tested on abstracts retrieved by the AND-queries. Best models were applied to abstracts retrieved by the OR-queries (see Methods). Total number of complexes in the dataset is 579, if not specified otherwise.

Query type	SVM model	L_{tot} [a]	L_{int} [b]	Coverage (%) [c]	Success (%) [d]	Accuracy (%) [e]
Basic text mining (numbering of residues from PDB only)						
AND		128	108	22.1	18.7	84.4
OR		328	273	56.6	47.2	83.2
Basic text mining (numbering of residues from PDB and UniProt)						
AND		142	118	24.5	20.4	83.1
OR		342	283	59.1	48.9	82.7
Basic text mining (numbering of residues from PDB, only abstracts prior original publication)						
AND		96	75	16.6	13.0	78.1
OR		268	202	46.3	34.9	75.4
SVM-enhanced text mining (numbering of residues from PDB only)						
OR	MF50L	266	211	45.9	36.4	79.3
OR	AF138L	269	213	46.5	36.9	79.2
OR	AF24L	253	193	43.7	33.3	76.3
Basic text mining on benchmark 3 (99 complexes, numbering of residues from PDB only)						
OR		93	82	93.9	82.8	88.2

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
d Ratio of L_{int} and total number of complexes
e Ratio of L_{int} and L_{tot}

Table 2-4: *Classification of abstracts in the test set by the three optimal SVM models. Total number of abstracts 261 (90 PPI-relevant and 171 non-PPI).*

SVM model	TP	FN	TN	FP
MF50L	48	42	123	48
AF138L	52	38	115	56
AF24L	46	44	128	43

Performance of SVM-enhanced text-mining protocol. While retrieving abstracts with the residues for significantly larger amount of PPI, the OR-queries bring also up many irrelevant residues. As the first step in mitigating this problem, we filtered 7,991 abstracts for 328 complexes (hereafter, called the original set of complexes), for which residues were found in the abstracts retrieved by the OR-queries, using three optimal SVM models (MF50L, AF138L and AF24L), trained and validated on the 1,305 abstracts retrieved by the AND-queries (see above). In this approach, an abstract is classified either as positive (publication in the PPI context) or negative (not PPI-relevant context) and then only positive abstracts are retained for the P_{TM} calculation (Eq. (2-1).

MF50L, AF138L and AF24L models removed at least one abstract for 296, 294 and 302 complexes, respectively, which is ~ 90 % of the initial set. Overall performance of the SVM-enhanced TM did not change significantly (middle part of **Table 2-3** and **Figure 2-5 A**) compared to the basic TM (upper part of **Table 2-3** and **Figure 2-2**). However, complexes, for which SVM models erroneously remove interface residues (P_{TM} decreases) constitute only ~ 1/3 of the initial set (**Figure 2-5 B**). The SVM models filtered out all PPI-relevant and all PPI-irrelevant (with non-interface residues) abstracts for ~13 % and ~ 7 % of the initial dataset, respectively (hatched parts of the $\Delta P_{TM} < 0$ and $\Delta P_{TM} = 0$ bars in **Figure 2-5 B**).

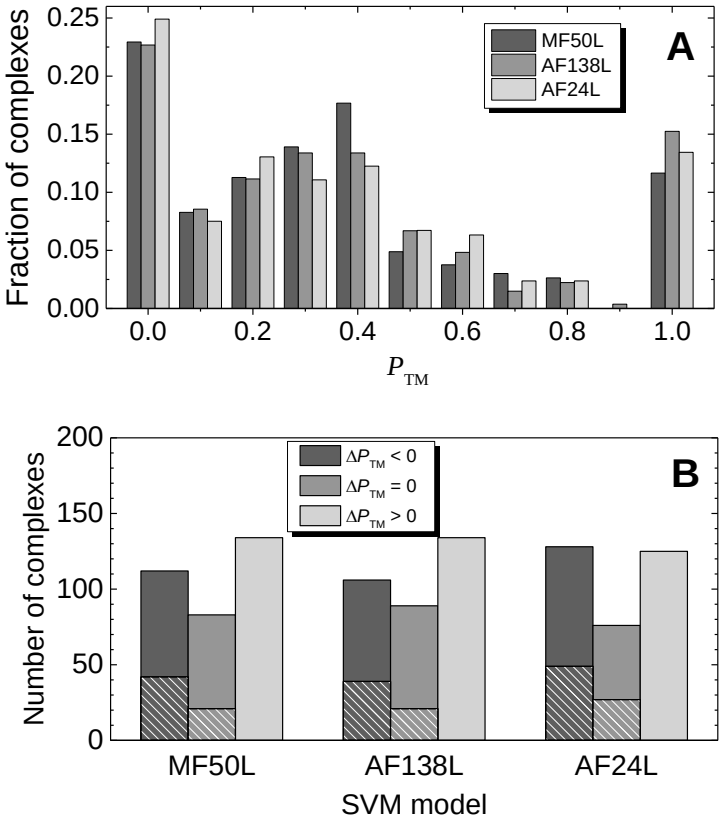


Figure 2-5: *Performance of the best SVM models.*
*The abstracts were retrieved by the OR-queries. Distribution of complexes (A) is shown according to the TM performance, P_{TM} (Eq.(2-1). The distribution is normalized by the total number of complexes for which residues were retrieved (column two in **Table 2-3**)). After filtering of abstracts by the optimal models, for a number of complexes (B) P_{TM} improves ($\Delta P_{TM} > 0$), does not change ($\Delta P_{TM} = 0$) and gets worse ($\Delta P_{TM} < 0$). Hatched areas show the number of complexes, for which the optimal models removed all abstracts.*

Analysis of performance of the SVM models on several complexes (**Table A-1** and **Text A-2**) revealed somewhat erratic performance of different models, caused by the inconsistency in residue context where interface residues are present in the abstracts with prevailing non-PPI features and vice versa (e.g., Ala11 of TIMP3 was found in the abstract of study about vasoconstricting peptide administration, not directly relevant to this protein binding [85]).

2.3.3 Docking with the text mining constraints

We ran the free docking by GRAMM to model complexes of unbound proteins from the DOCKGROUND X-ray benchmark 3 [79] using constraints generated by the basic TM protocol with OR-queries (see Methods).

In the unbound set of 99 complexes, by design the component proteins have both the co-crystallized and separately resolved X-structures, and as such were presumably on average more extensively studied than the complexes from the main bound set of 579 complexes used in this study for TM evaluation. This resulted in a significantly larger pool of publications extracted by the OR-queries (68 abstracts per complex for the unbound set, compared to 32 abstracts per complex for the bound set). Thus, a significantly larger number of residues per complex were identified (**Figure A-13**) and the TM performed better on the unbound set than on the larger bound one (last row in **Table 2-3**). However, the number of irrelevant (non-interface) residues was also significantly larger, reducing TM effectiveness (**Figure A-14**). The AND-queries retrieved abstracts with residues for 37 complexes only and TM protocol with the AND-queries was not used here separately. However, for residue ranking (Eq. (2-7)), we kept track of which residues were retrieved by the AND-queries. The TM results based on OR-queries for the top 10 residues per complex (5 for each protein, ranked by the frequency of the residue occurrence, Eq. (2-7)) were significantly better (**Figure A-14**). Thus, these residues were submitted to GRAMM docking program for scoring of the docking scan output.

To single out the role of the TM constraints, they were applied to re-rank unrefined and otherwise unscored docking models output directly from the GRAMM scan (the baseline for evaluating the impact of the TM constraints). For comparison, the re-ranking was also done separately with the correct interface residues as constraints (see Methods). We used strict (at least,

one model with *i*-RMSD ≤ 5 Å in top 10 predictions) and relaxed (at least, one model with *i*-RMSD ≤ 8 Å in top 100 predictions) success criteria.

The TM scoring significantly increased docking success rates, by 71% (compared to the baseline shown as blue columns in **Figure 2-6**) according to the stricter criterion, and by 32% according to the relaxed one (**Figure 2-6**). The results on the reference set of constraints, corresponding to the correct interface residues, showed that 27 complexes have near-native matches in the top 20,000 scan predictions according to the strict criterion, and 62 according to the relaxed one. The RMSD was calculated between the unbound ligand predicted match and the unbound ligand structurally aligned with the bound in the complex. Such alignment has significant mismatches with the receptor in a number of complexes, due to the conformational change upon binding. So the near-native matches for such complexes cannot be predicted by the surface complementarity-based rigid body free docking (this correlates with the docking decoys results [86] where a near-native match was found only for 61 complexes in 500,000 top scan matches).

The observed increase of the docking success rate is the result of constraints from the basic TM only. One can assume that the deep parsing/NLP will lead to further improvement of the docking quality, closer to the level of the reference constraints (**Figure 2-6**).

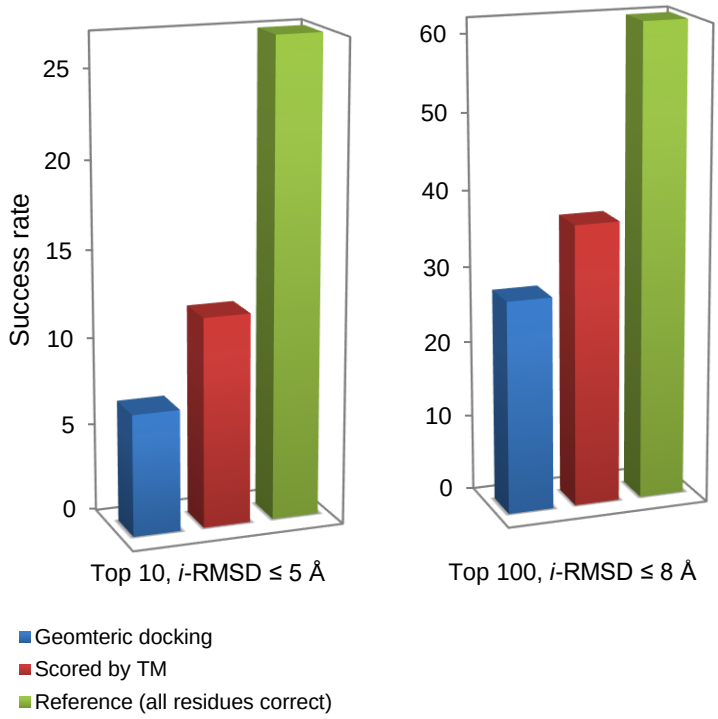


Figure 2-6: *Docking with TM constraints.*
The results of benchmarking on the unbound X-ray set from DOCKGROUND*. A complex was predicted successfully if at least one in top ten matches had ligand C^{α} interface RMSD ≤ 5 Å (A), and one in top hundred had RMSD ≤ 8 Å (B). The success rate is the percentage of successfully predicted complexes in the set. The low-resolution geometric scan output (20,000 matches) from GRAMM docking, with no post-processing, except removal of redundant matches, was scored by TM results. The reference bars show scoring by the actual interface residues (see text).*

2.4 Conclusions

TM has been widely used in recreating PPI networks, as well as in detecting functional sites (small ligand binding sites) on protein structures. Combining and expanding these two well-developed research areas, we applied TM to structural modeling of protein-protein complexes (protein docking). Abstracts of publications on 579 protein complexes from DOCKGROUND were retrieved from PubMed, using AND- and OR-queries (both proteins and at least one protein mentioned in

the text, correspondingly). The AND-queries identified more correct residues than the OR-queries, but retrieved abstracts with residues for significantly less complexes. SVM was used to improve the performance of OR-queries. The SVM models generated using simple bag-of-words representation of the text, removed irrelevant information extracted by the OR-queries, albeit not enough for an accurate discrimination of non-interface from the interface residues, as shown by the inconsistent performance of different SVM models. Whereas human expertise can consistently distinguish relevant from non-relevant to the interface information (as shown by our evaluation of a small subset of abstracts), a reliable and accurate automated procedure requires greater sophistication than the basic one used in our study.

The basic TM was used to generate constraints for docking, and tested on the protein-protein unbound docking benchmark set. TM significantly increased the docking success rates. Contextual analysis by deep parsing on sentence/residue level (an on-going study in our group) should improve the detection of the interface residues, and further increase the docking success rates. The preliminary results in this proof-of-concept study showed that TM is a promising approach to protein docking, with its utility increasing along with the rapidly growing amount of publicly available information on protein complexes.

Chapter 3

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

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

[1]Center for Computational Biology and [2]Department of Molecular Biosciences, The University of Kansas, Lawrence, Kansas 66047, USA

BMC Bioinformatics 2018; 19:84.

3.1 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 [11]. Both docking paradigms produce a large pool of putative models, and selecting the correct one is a non-trivial task, performed by scoring procedures [87]. Often knowledge of a few binding site residues is enough for successful docking [88].

In recent years, the number of biomedical publications, including PPI-relevant fields, has been growing fast [46]. 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 pool of extracted knowledge. TM+NLP techniques are widely used in biological text mining [89-102], particularly for the extraction and analysis of information on PPI networks [5-7, 49, 51, 53, 103-112] and for the prediction of small molecules binding sites [8, 9].

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.

3.2 Methods

3.2.1 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) [79]. The basic stage of the procedure consists of two major steps: information retrieval and information extraction [37] (**Figure 3-1**). 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

$$P_{\mathrm{TM}} = \frac{\sum_{i=1}^{N} N_i^{\mathrm{int}}}{\sum_{i=1}^{N} \left(N_i^{\mathrm{int}} + N_i^{\mathrm{non}} \right)}, \qquad (3\text{-}1)$$

where N_i^{int} and N_i^{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_{TM} calculations). It is convenient to compare the performance of two algorithms for residue filtering in terms of

$$\Delta N\left(P_{\mathrm{TM}}\right)=N_{tar}^{X_1}\left(P_{\mathrm{TM}}\right)-N_{tar}^{X_2}\left(P_{\mathrm{TM}}\right), \tag{3-2}$$

where $N_{tar}^{X_1}\left(P_{\mathrm{TM}}\right)$ and $N_{tar}^{X_2}\left(P_{\mathrm{TM}}\right)$ are the number of targets with P_{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_{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.

3.2.2 S*election 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 [113, 114], Lin [115] and Path [116, 117], correspondingly, between the token (words) in a residue-containing sentence and the generic keywords. These Perl modules, provided by the WordNet [118, 119] (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.

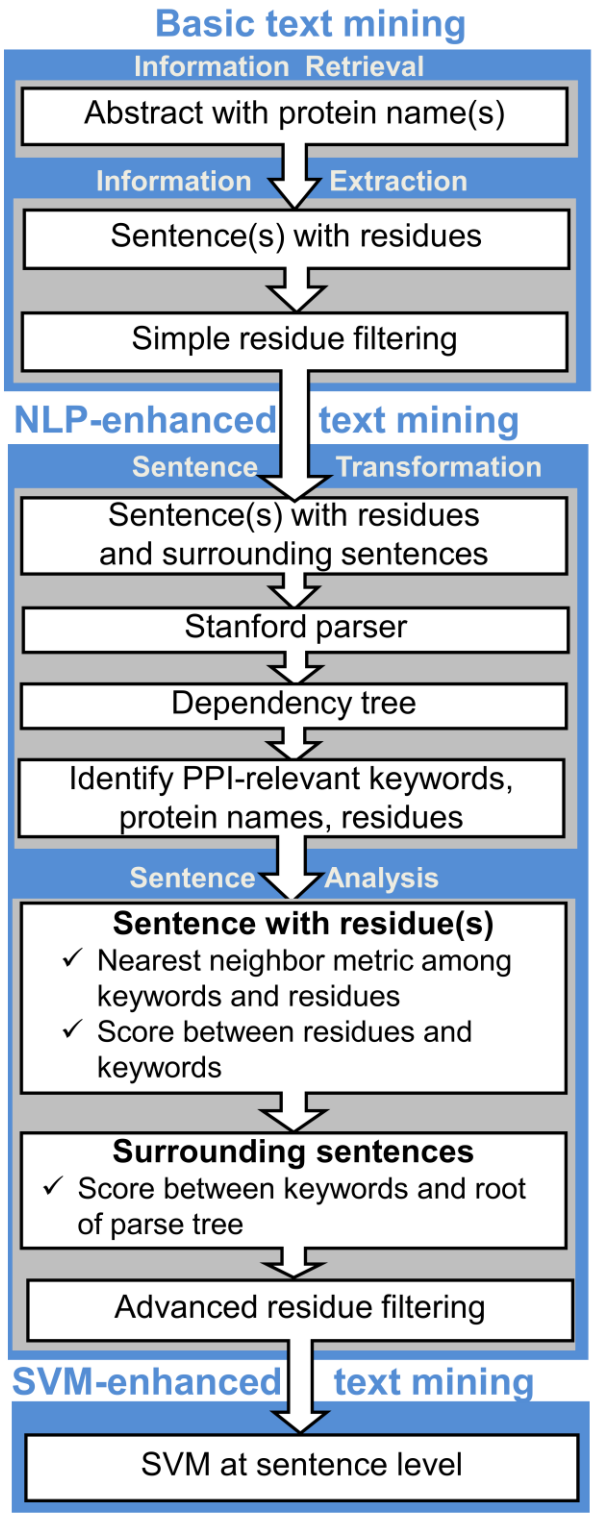


Figure 3-1: *Flowchart of NLP-enhanced text mining system. Scoring of surrounding sentences is shown for Method 3 (see text).*

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 3-3**) were selected from manual analysis of the parse trees for 500 sentences from 208 abstracts on studies of 32 protein complexes.

3.2.3 Scoring of residue-containing and context sentences

The parse tree of a sentence was built by the Perl module of the Stanford parser [120, 121] (http://nlp.stanford.edu/software/index.shtml) downloaded from http://search.cpan.org. The score of a residue in the sentence was calculated as

$$S_{\mathrm{X}} = \sum_i \frac{1}{d^{+}_{\mathrm{X}i}} - \sum_j \frac{1}{d^{-}_{\mathrm{X}j}}, \qquad (3\text{-}3)$$

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 (**Figure B-1**).

An add-on value to the main S_{X} score (Eq. (3-3)) 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_{X} score, but between the keywords and the root of the sentence on the parse tree.

3.2.4 SVM model

The features vector for the SVM model was constructed from the S_{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

$$S_{\text{prot}} = \begin{cases} 0, & \text{if no protein names in the sentence} \\ 1, & \text{if only name of one protein in the sentence} \\ 2, & \text{if name of both proteins in the sentence} \end{cases}, \quad (3\text{-}4)$$

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 [75, 122, 123]. 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 [124]

$$P = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad R = \frac{\text{TP}}{\text{TP}+\text{FN}}, \qquad A = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FN}+\text{TN}+\text{FP}}, \quad F = 2\frac{P \times R}{P+R}, \quad (3\text{-}5)$$

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 (**Figure B-2-Figure B-7**) showed that the best performance was achieved using RBF kernel with gamma 16. Thus, this model was incorporated in the TM protocol (**Figure 3-1**).

3.2.5 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

$$f(R) = \min\left(10, \sum_{i=1}^{N_R} a_i\right), \tag{3-6}$$

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 [125]. 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

$$f(R)=\begin{Bmatrix} 10, & if\ \ for\ some\ i, a_i\ is\ retrieved\ in\ AND\ query\ and\ \ passes\ NLP \\ 8, & if\ \ a_i\ is\ retrieved\ in\ AND\ query \\ 6, & if\ \ any\ a_i\ retrieved\ in\ OR\ query\ \ passes\ NLP \\ max(5, count\ of\ \ abstracts\ containing\ R) \end{Bmatrix}, \qquad (3\text{-}7)$$

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^{α} - C^{α} 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.

3.3 Results and Discussion

3.3.1 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 [118, 119] (generic English lexical database with words grouped into sets of cognitive synonyms), which does not contain any knowledge-domain specific vocabularies [126]. 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 3-1** and **Figure 3-2**). 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 [127]. 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. [128] that hierarchical structure of generic and domain-specific vocabularies are different and thus, for example, MESH specific vocabulary [129] provides more accurate knowledge representation of medical concepts compared to the generic WordNet lexicon.

Next, we tested applicability of the 7 specialized dictionaries (**Table 3-2**) 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 3-2** and **Figure 3-3**) 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 3-2**) 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 3-2**). Interestingly, the best performing dictionary by Schuhmann *et al.* [130] contains the smallest number of words.

Table 3-1: *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 [113, 114]	319	267	55.1	46.1	83.7	-3	-1
OR	Lin [115]	251	184	43.4	31.8	73.3	+8	-8
OR	Path [116, 117]	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 L_{tot} and total number of complexes
d Ratio of L_{int} and total number of complexes
e Ratio of L_{int} and L_{tot}
f Calculated by Eq.(3-2)

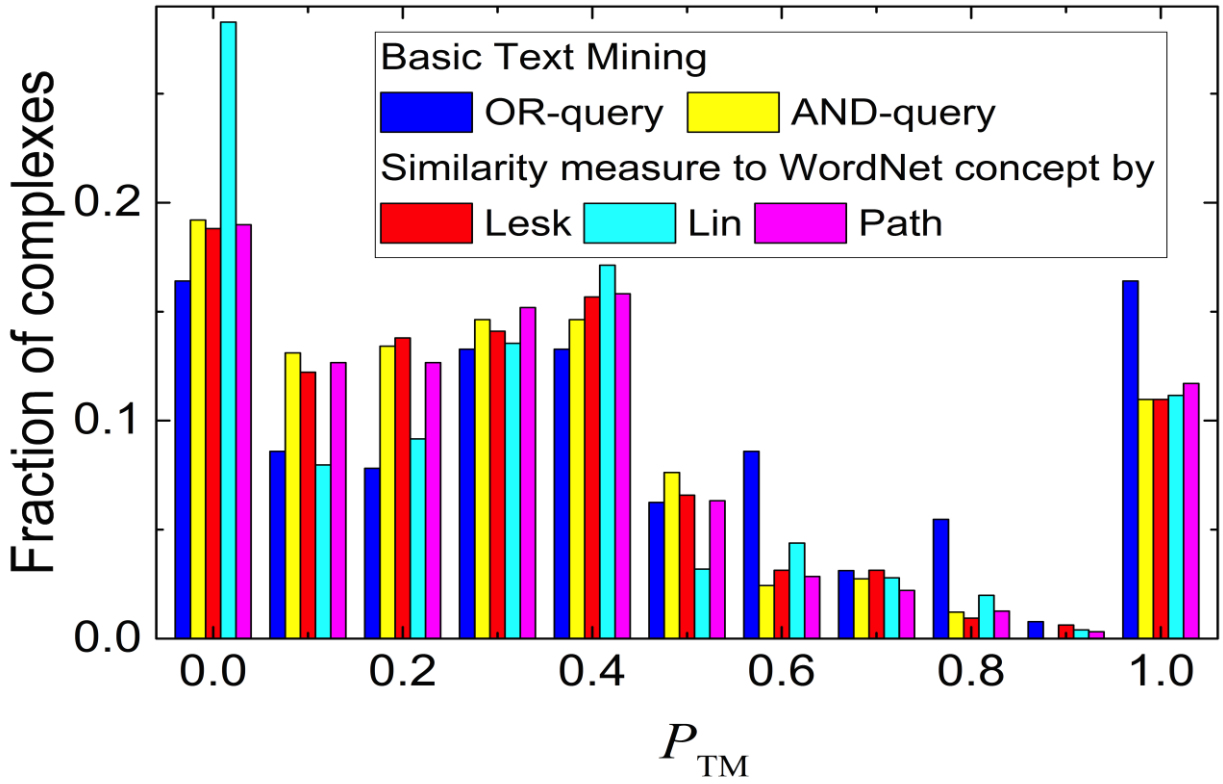


Figure 3-2: *Performance of basic and advanced text mining protocols.*
*Advanced filtering of the residues in the abstracts retrieved by the OR-queries was performed by calculating various similarity scores (see legend) between the words of residue-containing sentences and generic concept words from WordNet. The TM performance is calculated using Equation (3-1). The distribution is normalized to the total number of complexes for which residues were extracted (third column in **Table 3-1**).*

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 **Figure 3-3** and the bottom row in **Table 3-2**). This suggests that even a limited amount of text provided by abstracts can be used to extract reliable PPI-relevant keywords.

Table 3-2: *Overall text-mining performance with the residue filtering based on spotting in the residue-containing sentences keyword(s) from specialized dictionaries.*
For definitions of columns 3 - 9, see footnotes to ***Table 3-1****. Full content of in-house dictionary is in* **Table 3-3***, but only PPI+ive part was used to calculate the data in this Table.*

Dictionary and reference	Number of PPI keywords	L_{tot} [a]	L_{int} [b]	Coverage (%) [c]	Success (%) [d]	Accuracy (%) [e]	$\Delta N(0)$ [f]	$\Delta N(1)$ [f]
Blaschke et al, [6]	43	265	205	45.8	35.4	77.4	0	-8
Chowdhary *et. al*, [131]	191	284	233	49.1	40.2	82.0	-7	-4
Hakenberg et al [132]	234	297	232	51.3	40.1	78.1	6	-7
Plake et al [133]	73	291	230	50.3	39.7	79.0	1	-1
Raja *et al* [104]	412	302	247	52.2	42.7	81.8	0	-5
Schuhmann et al [130]	64	212	152	36.6	26.3	71.7	-1	5
Temkin et al [5]	174	283	223	48.9	38.5	78.8	0	-9
Own dictionary	16	224	169	38.7	29.2	75.4	-6	8

3.3.2 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 ***3-3***) on the parse tree, quantified in the score S_X calculated by Eq. (3-3) 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 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.

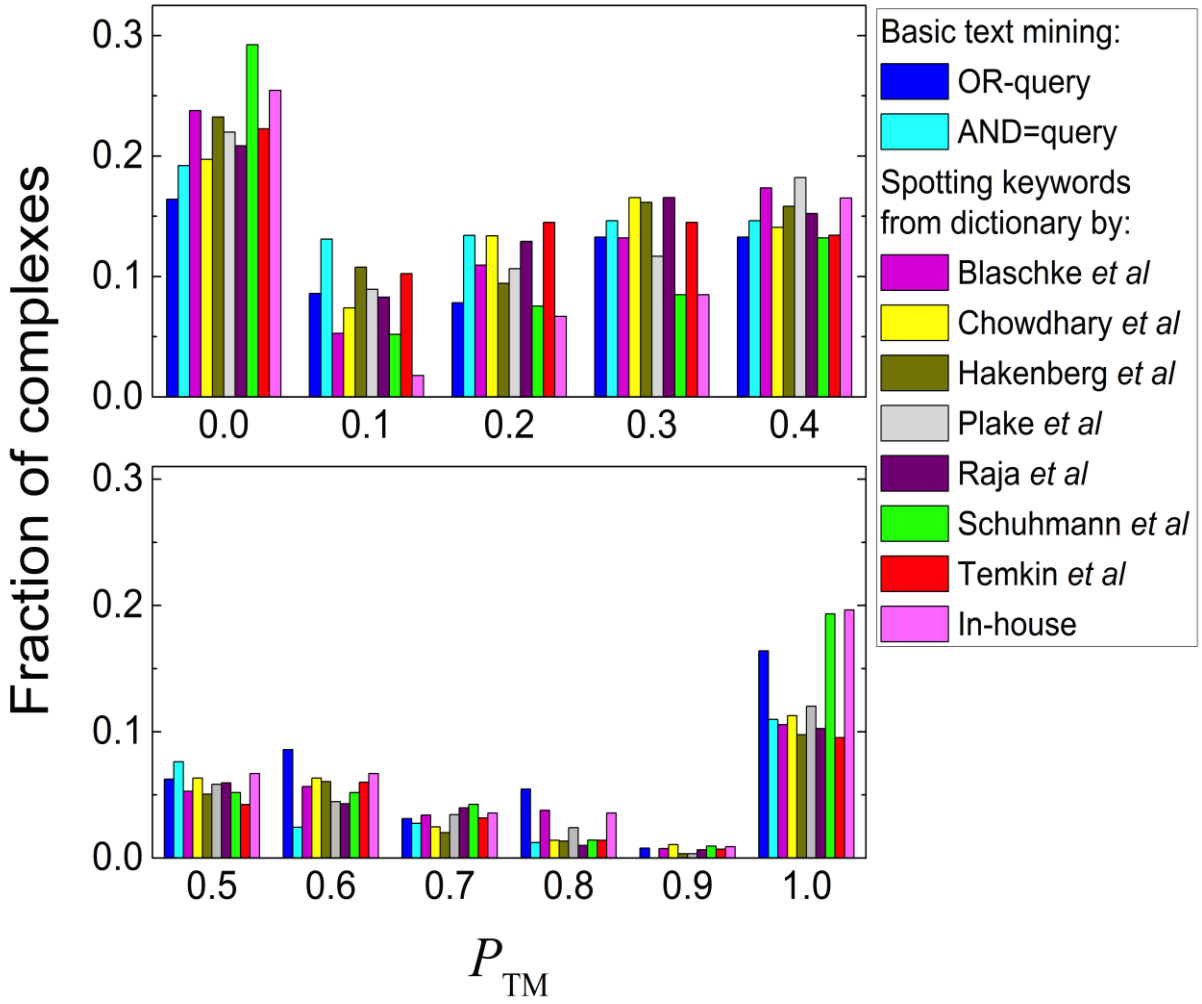


Figure 3-3: *Performance of basic and advanced text mining protocols.*
*Advanced filtering of the residues in the abstracts retrieved by the OR- queries was performed by spotting PPI-relevant keywords from various specialized dictionaries (see legend). The TM performance is calculated using Equation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

Table 3-3: *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 interface residues tend to have $S_X > 0.25$ (**Figure B-8**). 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 3-4** and red bars in **Figure 3-4**).

Table 3-4: *Overall text-mining performance with the residue filtering based on analysis of sentence parse tree.*
*Keywords used in the analysis were taken from our dictionary (***Table 3-3***). For definitions of columns 2 - 8, see footnotes to* ***Table 3-1****.*

Method of parse tree analysis	L_{tot}	L_{int}	Coverage (%)	Success (%)	Accuracy (%)	$\Delta N(0)$	$\Delta N(1)$
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

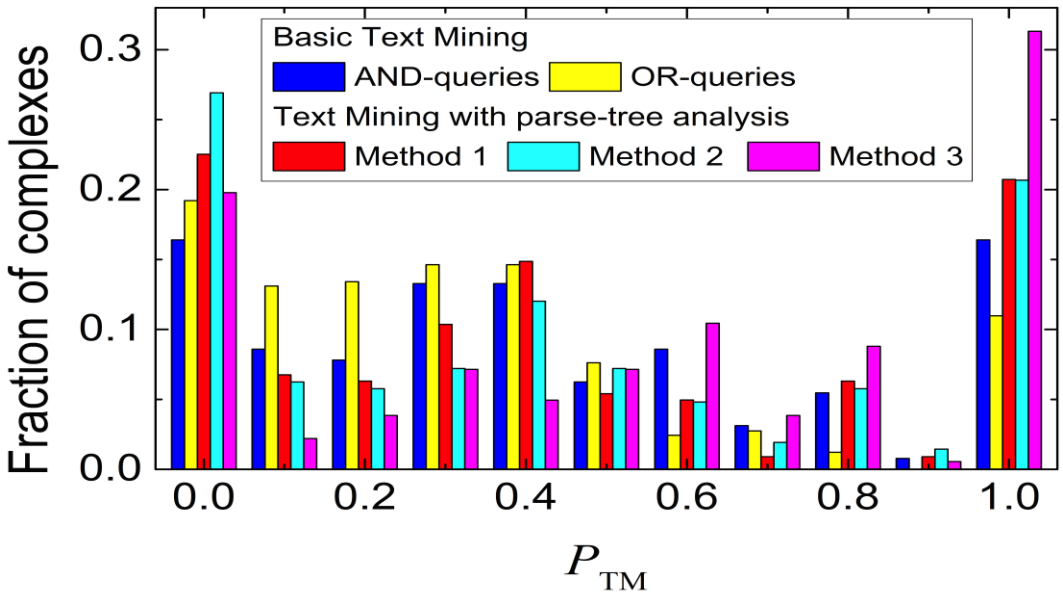


Figure 3-4: *Performance of basic and advanced text mining protocols.*
Advanced filtering of the residues in the abstracts retrieved by the OR-queries was performed by different methods of analysis of the sentence parse trees (for method description see first column in ***Table 3-4****) The TM performance was calculated using Equation (3-1). The distribution is normalized to the total number of complexes for which residues were extracted (second column in* ***Table 3-4****).*

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 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 3-4** and cyan bars in **Figure 3-4**) 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 3-4**). An example of successful filtering of non-interface residues is shown in **Figure 3-5** for the chains A and B of 2uyz. Out of five residues mined by the basic TM protocol, only one residue (**Figure 3-5**, 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 **Table B-1** 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 (**Figure B-9**) did not differ much from the results obtained on the full set of abstracts.

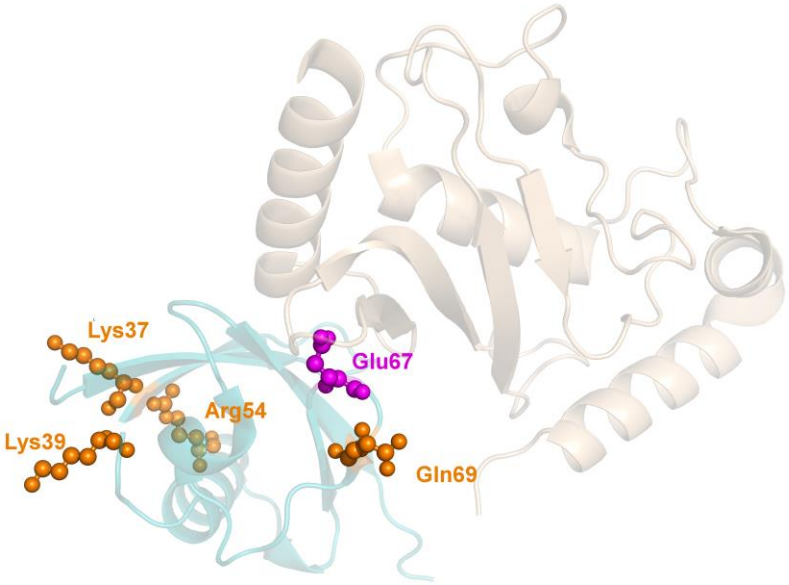


Figure 3-5: *Successful filtering of mined residues by the SVM-based approach of the parse-tree analysis*
*(Method 3 in **Table 3-4**). The structure is 2uyz chains A (wheat) and B (cyan). Residues mined by the basic TM protocol are highlighted. The ones filtered out by the advanced TM protocol are in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

3.3.3 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 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 (3-7)), 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^{α} 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 $\leq$ 5 Å in top 10 predictions. The results (**Figure 3-6**) 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.

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.

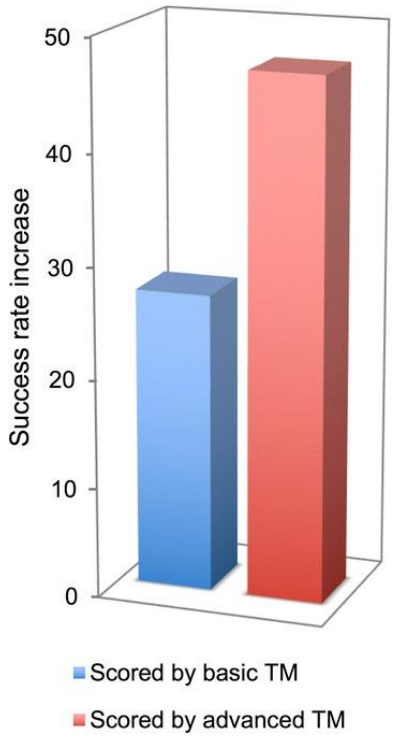


Figure 3-6: *TM contribution to docking.*
The success rate increase of the rigid-body global docking scan by GRAMM using constraints generated by basic TM and the advanced TM with NLP.

3.4 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.

.

Chapter 4

Enhanced text mining of biomedical literature for modeling of protein complexes

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

[1]Center for Computational Biology and [2]Department of Molecular Biosciences, The University of Kansas, Lawrence, Kansas 66047, USA

Submitted

4.1 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 structures of the interacting proteins [11]. A typical docking pipeline involves three major steps: (*i*) global scan generating multiple tentative docking poses, (*ii*) evaluation of these poses by scoring functions, and (***iii***) structural refinement of the top-scoring predictions. The ability to differentiate near-native/correct predictions from false-positives determines the overall docking performance. Because of the complexity of the docking problem, information on the docking target that can constrain the docking search is of great value.

Scientific publications are a rich source of information on protein complexes and their structures. With the growing number of PPI-related biomedical publications in public repositories, such as PubMed, the amount of this information is rapidly increasing. However, while the textual content of the publications is easily understandable by experts in the field, for the benefit of the modeling tools development and application to PPI, such content has to be processed as an automated input to the docking procedures.

The text mining (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 [134-136]. The full texts use longer sentences and parenthesized material [137] and have heterogeneous distribution of information (as measured by density of keywords in various sections) [138]. Access to the full-text papers creates a more comprehensive source (corpus) for the text mining and increases the

recall compared to the abstracts [139]. However, copyright restrictions generally limit the use of full text articles in the text mining [140, 141]. The number of PMC-OA articles (the freely available subset of full-text papers) is not increasing at the same rate as the number of PubMed abstracts. The full-text articles have statistical properties (such as term or document frequency) that are more robust, but have more noise compared to the abstracts [142]. Text mining of the full-text papers has helped in extraction of various biological information [89, 143-148], including one on non-structural aspects of PPI [132, 149-152].

Specific patterns in datasets can be identified by machine learning (ML) techniques, especially, deep learning (DL) approaches implemented using neural networks (NN) with hidden layers several levels deep. Each successive layer learns higher level of abstraction [153, 154]. NN are trained using the back-propagation algorithm [153] where the error (the difference between actual and desired output) is projected backwards layer-by-layer, with the connection weights adjusted in proportion [155]. The NN applications include, but not limited to automatic speech recognition [156], machine translation [157], paraphrasing [158] , and image and scene annotation [159, 160].

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 [161, 162]. 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 vs. molecule" and "residues vs. residue" would have similar offsets). Word vectors can be efficiently estimated

on a large scale [163]. They are widely used as a first generic step in a united architecture for solving a specific Natural Language Processing (NLP) task using deep neural networks [160-166], e.g. for machine translation requiring large vocabulary across multiple languages [167]. The word vectors are used in a sentence-level sentiment analysis [168, 169] (scoring or quantification of subjective information such as “tone of a speaker” or “attitude of a customer”).

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. We showed that even such basic TM technique enhances docking success rate [37]. However, the information, mined in such simplistic manner, needs substantial post-processing in order to remove inappropriate data which inevitably occur in the prediction pool when no effort is made to understand a larger context of the residue mentioning. Later, we improved filtering out residues non-relevant to the protein binding by utilizing simple NLP techniques, although the amount of such residues still remained high [170]. In this paper, we present a deep recursive neural network (DRNN) model based on the word vectors for analyzing sentiments of residue-containing sentences and context of residue mentioning. The model was trained on a large body of full-text articles from PMC-OA and applied to purging of irrelevant residues, mined from the PubMed abstracts. We also show that while full-text papers are a richer source of information, their current availability limits their usefulness for extracting PPI-relevant information by automated TM procedures. The results show that, similarly to NLP, the DL approach is significantly better than the basic TM in extracting information relevant to modeling from the abstracts. Both DL and NLP are superior to the basic TM is processing of the full-text papers. With the increase of the full-text papers availability, the usefulness of this source of information for structural modeling of protein complexes will further grow.

4.2 Methods

4.2.1 Basic text mining protocol

Our basic TM tool consists of information retrieval (IR) and information extraction (IE) [37]. In this study, in addition to using PubMed resources from https://www.ncbi.nlm.nih.gov/ [171], 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. Thus, 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 [37], in order to retrieve relevant articles for a protein pair, we generate AND-queries (both proteins in the complex are mentioned in the text) and OR-queries (either of the proteins is mentioned) for the PubMed abstracts 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.

The basic IE (**Figure 4-1**) was performed on the retrieved articles/abstracts by spotting different variations of the residue name and number, and performing simple filtering [37]. In this study, the sentences containing the initially mined residues from the full-text articles were used to estimate the effectiveness of the basic TM on the full-texts and to train the DRNN neural network. The trained DRNN was further used to classify residues (interface or non-interface) mined from the PubMed abstracts.

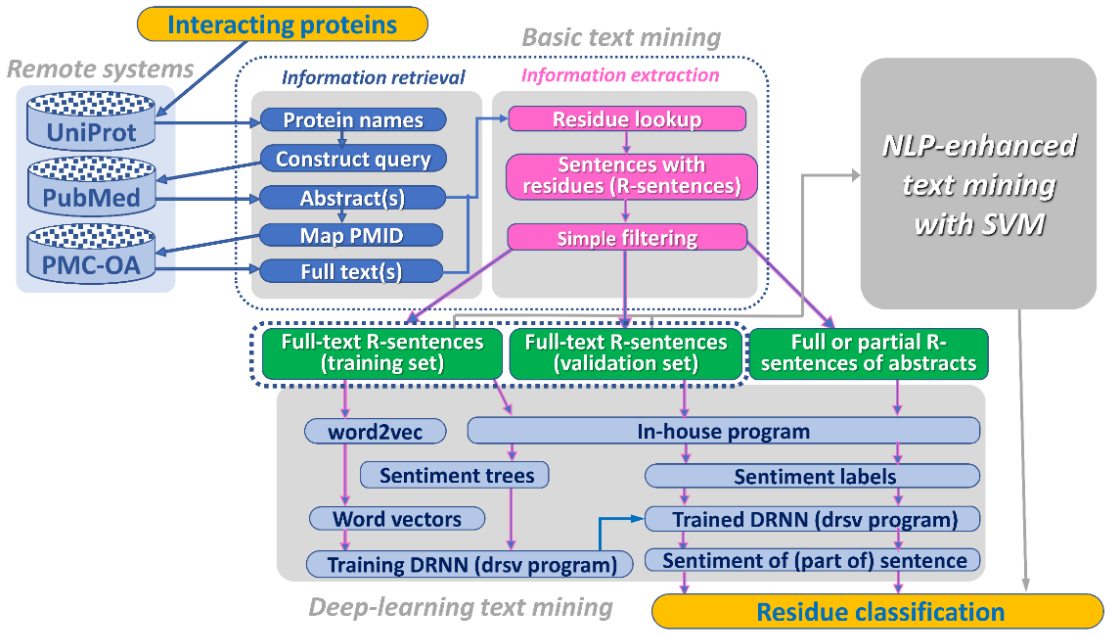


Figure 4-1: *Flowchart of the text-mining system.*
Algorithm of the NLP-enhanced text mining with SVM is the same as in Ref. [170] and thus is not shown in detail. Both full text sets are unified in one training set (shown by the dashed line) when the trained neural network is tested on the PubMed abstracts.

4.2.2 Datasets

The approaches were benchmarked on the set of 579 non-redundant (at 30% sequence identity level) binary protein-protein complexes from the DOCKGROUND resource (https://dockground.compbio.ku.edu) [172]. The dataset for training DRNN consisted of 4,982 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. Those sentences were classified into 1,605 positive (interface residue) and 3,377 negative (non-interface residue) R-sentences (full training set). The interface residues were defined by 6 Å distance between atoms in different chains. The dataset for testing the trained

DRNN model comprised 5,786 R-sentences (or their parts around identified residues, see Results), 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. We also performed DRNN training and testing using PMC-OA full texts only. In this case, the R-sentences of the full training set were divided into training and validation (testing) sub-sets by splitting the alphabetically sorted list of 579 PDB codes of our dataset of the protein-protein complexes so that each part contains approximately half of the R-sentences (not every PDB code had R-sentences associated with it). The 803 positive and 1,689 negative R-sentences in the top 50% of the PDB list (259 complexes) were used for the DRNN testing (full-text testing set). The 802 positive and 1,688 negative R-sentences from the bottom 50% of the list (320 complexes) were used the DRNN training (reduced full-text training dataset). These datasets were also used to train Support Vector Machine (SVM) model.

4.2.3 Deep learning architecture

For the DRNN training, we generated PPI-specific sentiment tree bank, which is a set of binary trees (**Figure 4-2**) 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 [168], we utilized five standard sentiment classes: very +ive (labeled 4), +ive (3), neutral (2), -ive (1), very –ive (0). In a sentence of N words, the b_j is calculated as

$$b_j = \max\left(b_{j-1}, a_j\right) \quad \text{or} \quad b_j = \min\left(b_{j-1}, a_j\right),\ j = 2, \ldots, N \tag{4-1}$$

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

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

,

where F_j is the fixed sentiment label for PPI+ive and PPI-ive keywords, determined by its score.

Scores for keywords **(Table C-1)** were generated by dividing sentences from full-text papers with mined residues into positive (residue at the interface) and negative (residue not at the interface). For each word, we computed its frequency (percent of sentences with the keyword) for the positive sentences and for the negative sentences. The difference of the two values represented the score (bias). This set is identical to the one used for SVM training and testing (see below). Words with the score between 1 and -1 were ignored (bias < 1%). A protein, amino acid, or species names were treated in the same manner. Stop words were removed.

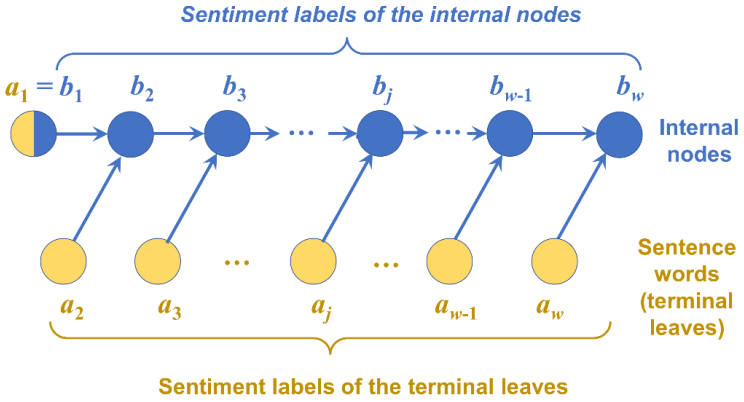


Figure 4-2: *Schematic representation of a sentence binary tree and associated sentiment labels.*

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), $\left\{\overrightarrow{v^k}\right\}$, for each of the 74,438 unique words in the sentences of the training set (out of ~20M 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) [161-163]. The dimensionality of the word vectors was set to 300, considered sufficient for complex NLP tasks [161, 173, 174]. The word vectors corresponding to similar words were distributed close to each other (**Figure *4*-*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) [165] 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 (**Figure C-3**). 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 (Eq., (4-2) 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 [165, 168]).

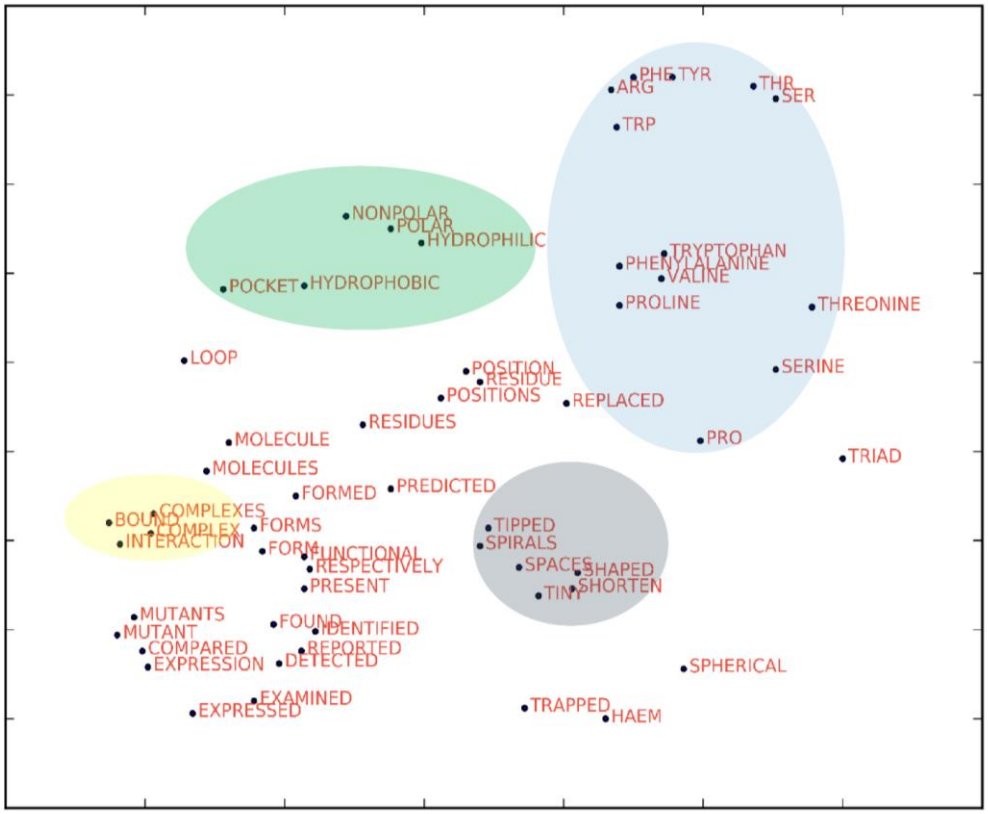


Figure 4-3: *Example of the initial word vectors distribution.*
Highlighted areas show similar words in the same region of the vector space. The distribution was generated by arbitrarily choosing a set of 55 words that are typically found in PPI publications, not meeting any scientific criteria, but 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 [175] 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.

4.2.4 NLP-hybrid approach

Previously, we explored various Natural Language Processing (NLP) approaches at semantic and syntactic levels for filtering residues mined from the PubMed abstracts. We showed that NLP hybrid approach, which consists of Support Vector Machine (SVM) Model with features extracted

from the parse trees of the residue-containing (R-sentences) and context sentences (immediately preceding and following the R-sentence) yields the best performance [170]. Thus, in this paper, we explored performance of the NLP hybrid approach on the PMC-OA full-text articles.

Parse trees of R- and context sentences were built by the Perl module of the Stanford parser [121, 176] (http://nlp.stanford.edu/software/index.shtml) downloaded from http://search.cpan.org. The score of a residue in a R-sentence was calculated as

$$S_{\mathrm{X}} = \sum_i \frac{w_i}{d_{\mathrm{X}i}}, \tag{4-3}$$

where summation is over all keywords from **Table C-1** in the sentence, w_i is the score of keyword *i* (**Table C-1**), and $d_{\mathrm{X}i}$ is the distance between the residue and keyword *i*, calculated by the edge counting in the parse tree. The S_{x} scores tend to be more positive for the interface and negative for the non-interface residues (**Figure C-1**). For the context sentences, scores, similar to the score S_{x}, but between keywords (**Table C-1**) and the root of the sentence on the parse tree, were calculated. We also utilized an additional score accounting for the presence of one ($S_{\mathrm{prot}} = 1$) or both ($S_{\mathrm{prot}} = 2$) protein names in the R- and context sentences ($S_{\mathrm{prot}} = 0$ if no protein names were mentioned). All these scores were used as features in the SVM model.

The SVM model was trained and validated (50-50 random split) on a subset of 1,605 positive sentences (with interface residues) and 3,377 negative sentences (with non-interface residues) from PMC-OA full-text (the training set also included the context sentences) using the program SVMLight with linear, polynomial and RBF kernels [75, 122, 123]. SVM performance was evaluated in terms of precision *P*, recall *R*, accuracy *A*, and *F* score [124]

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad A = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FN}+\mathrm{TN}+\mathrm{FP}}, \quad F = 2\frac{P \times R}{P+R}, \tag{4-4}$$

where TP, FP, TN, and FN are, the numbers of correctly identified interface, incorrectly identified interface, correctly identified non-interface and incorrectly identified non-interface residues in the validation set, respectively. Analysis of the results (**Figure C-2**) showed that the best SVM performance was achieved using RBF kernel with gamma 0.25, owing to the highest recall (0.41 vs. 0.40 and 0.39) for the top three highest accuracy gammas.

4.2.5 Performance evaluation

The performance of the TM protocol for a particular PPI, for which *N* residue-containing articles (abstract-only or full-text) were retrieved, was evaluated as

$$P_{\mathrm{TM}} = \frac{\sum_{i=1}^{N} N_i^{\mathrm{int}}}{\sum_{i=1}^{N} \left(N_i^{\mathrm{int}} + N_i^{\mathrm{non}} \right)} \tag{4-5}$$

,

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 one of the algorithms. If all residues in an article are purged, this article was excluded from the P_{TM} calculations. We also compared the performance of two algorithms for residue filtering [170]

$$\Delta N\left(P_{\mathrm{TM}}\right) = N_{tar}^{X_1}\left(P_{\mathrm{TM}}\right) - N_{tar}^{X_2}\left(P_{\mathrm{TM}}\right), \tag{4-6}$$

where, $N_{tar}^{X_1}(P_{\mathrm{TM}})$ and $N_{tar}^{X_2}(P_{\mathrm{TM}})$ are the number of targets with P_{TM} value yielded by algorithms X_1 and X_2, respectively. The negative values of $\Delta N(0)$ and the positive values of $\Delta N(1)$ indicate successful purging of the PPI-irrelevant residues from the mined articles by a tested algorithm, since the major contributions to P_{TM} distribution are from all false positive ($P_{\mathrm{TM}} = 0$) and all true positive ($P_{\mathrm{TM}} = 1$) cases.

4.3 Results and Discussion

4.3.1 Basic text mining of full-text articles

The full-text of a paper provides much more information than its abstract. But due to copyright restrictions only just over 1 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 *4-1***).

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 *4-1***). Significantly better TM performance on PMC-OA full-texts than on the PMC-OA abstracts (**Table *4-1***) 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 $P_{\mathrm{TM}} = 0$ and $P_{\mathrm{TM}} = 1$ in **Figure *4-4***).

Table 4-1: *Overall performance of basic TM on abstracts (PubMed and PMC-OA) and full texts (PMC-OA).*

Dataset	Query Type	L_{tot}[a]	L_{int}[b]	Coverage (%)[c]	Success (%)[d]	Accuracy (%)[e]
PubMed	AND	128	108	22.1	18.7	84.4
PubMed	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 Number of complexes, for which TM retrieved at least one article with residues
b Number of complexes with at least one interface residue found in the retrieved articles
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}

Research on a specific protein interaction could be published only in the journals with limited access to their full texts. Our results 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 **Figure 4-5** with the detailed description in **Text C-1**). 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.

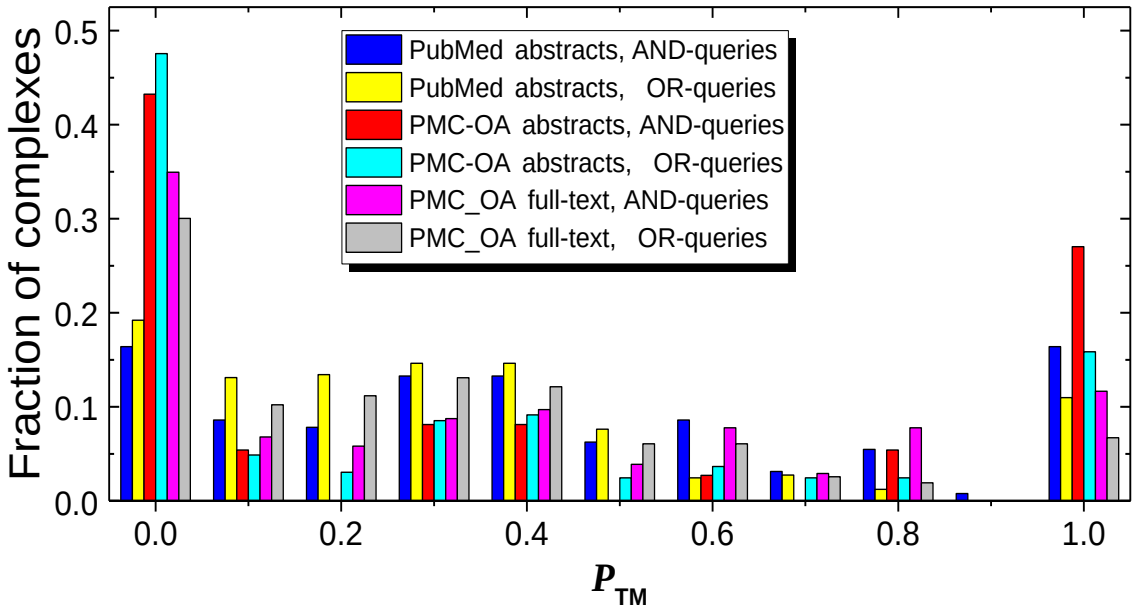


Figure 4-4: *Comparison of basic text mining on abstracts and full-texts.*
*The performance is calculated on 579 complexes by Eq.(4-5). The distribution is normalized to the total number of complexes for which residues were extracted (**Table 4-1**).*

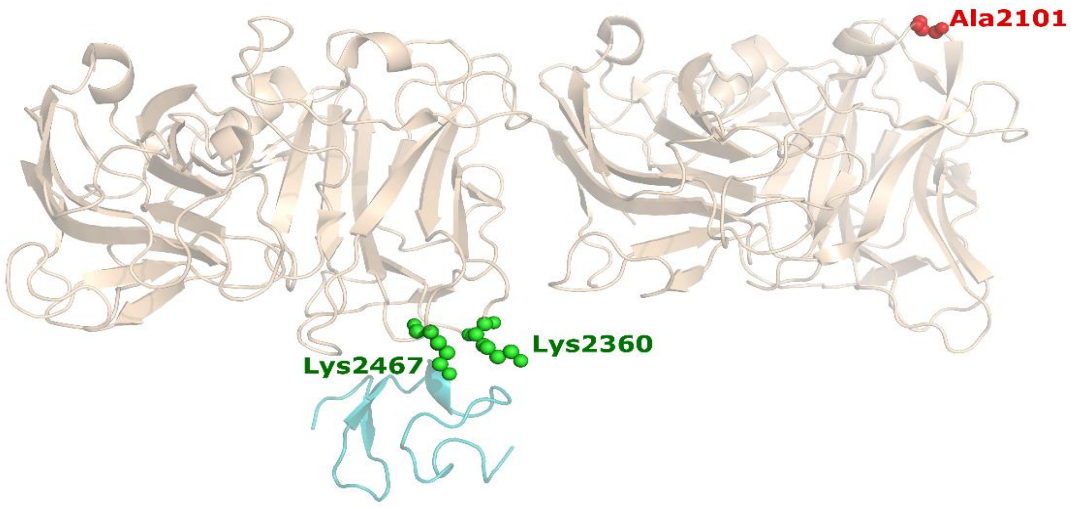


Figure 4-5: *Example of residues mined from abstracts but not from full texts.*
*The abstracts are from PubMed and the full texts are from PMC-OA. The structure is 3a7q chain A (gray) and B (cyan). Interface (green) and non-interface (red) residues are mined by the basic TM protocol. Details are in **Text C-1**.*

4.3.2 Enhanced text mining of full-text articles

Earlier, we showed that NLP techniques classify PPI-relevant residues mined from the PubMed abstracts better than the basic TM [170]. In this study, we applied the same methodology, with automatically generated keywords (**Table C-1**) for filtering initially mined residues in the PMC-OA full-text test set. The same set of the initially mined residues was also purged by the DRNN model, trained with the same set of keywords. Both SVM and DRNN models were trained on the same reduced full-text training set. Both methodologies similarly improved the TM of the full-text articles (**Table 4-2** and **Figure 4-6**) and their abstracts (**Table C-3** and **Figure C-6**). SVM model purged all initially mined residues for a smaller number of complexes. At the same time, it was better in removing non-interface residues from the full-text articles (**Table 4-2**). In the abstracts of the PMC-OA articles, NLP and DL removed all retrieved abstracts with the residues for ~ 70% of the complexes. Thus, these results were not statistically significant. The SVM performance only slightly depends on the keywords used for the SVM training and testing (**Table C-2** and **Figure C-5** show the results for the SVM model with the manually selected keywords from our previous study [170]). 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. In the future, DRNN performance can be potentially improved by the use of PPI- specific hand-curated sentiment tree bank. **Figure 4-7** illustrates NLP and DL-enhanced TM performance on PMC-OA full-text articles on heat shock HSP82 and AHA1 proteins.

Table 4-2: *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 NLP and DL. NLP included SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automated keywords. DL consisted of Deep Recursive Neural Network model for classifying residues in the entire sentence. NLP SVM and DL models were trained on reduced full-text training set.*

Method	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	–	–
NLP	87	58	33.6	22.4	66.7	–22	+15
DL	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 Eq. (4-6) with values from basic TM (first row) as X_2*

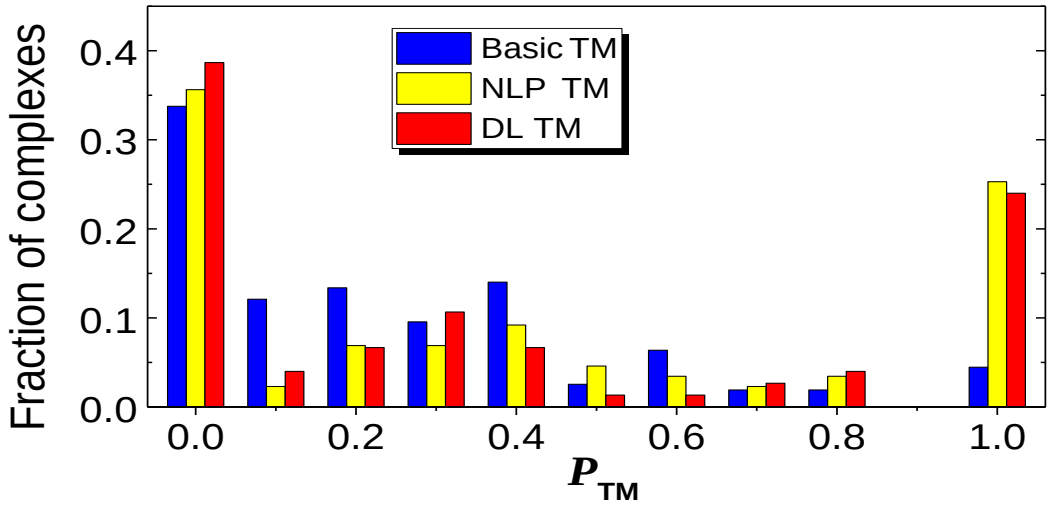


Figure 4-6: *Comparison of text-mining protocols on full texts.*
*The full texts are the PMC-OA set. The performance was calculated by Eq. (4-5). The distribution is normalized to the total number of complexes for which residues were extracted (**Table 4-2**).*

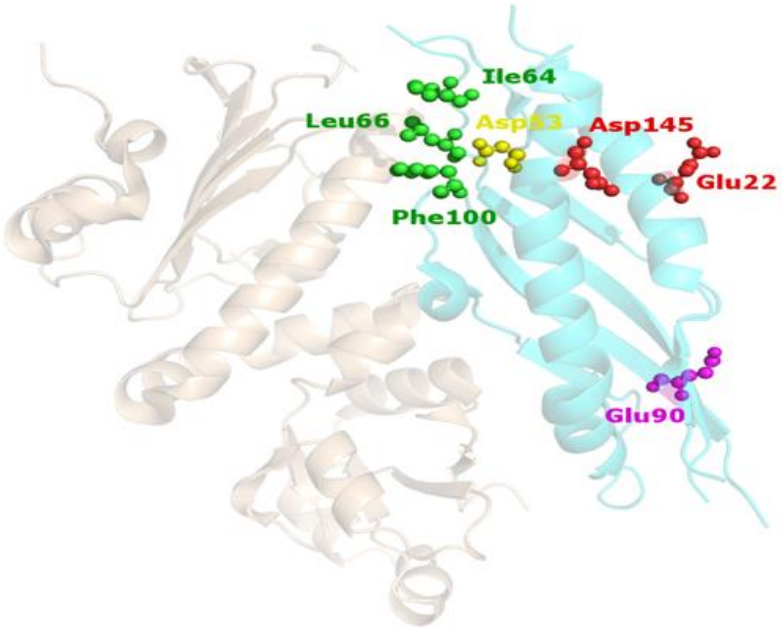


Figure 4-7: *Example of residues mined from full texts.*
The structure is 1usu, chain A (gray) and B (cyan). Basic TM identified 6 residues (4 at the interface, in green and yellow, and 2 not at the interface, in red). NLP SVM and DRNN models correctly classified 3 interface residues (green). Only one 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 ***Text C-2****.*

4.3.3 Enhanced text mining of abstracts

When both SVM and DRNN are trained on full-text articles, and applied to classification of residues in the abstracts, the results are similar, with somewhat better performance of DRNN (**Table 4-3** and **Figure 4-8**). For DRNN, 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. 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 4-3**).

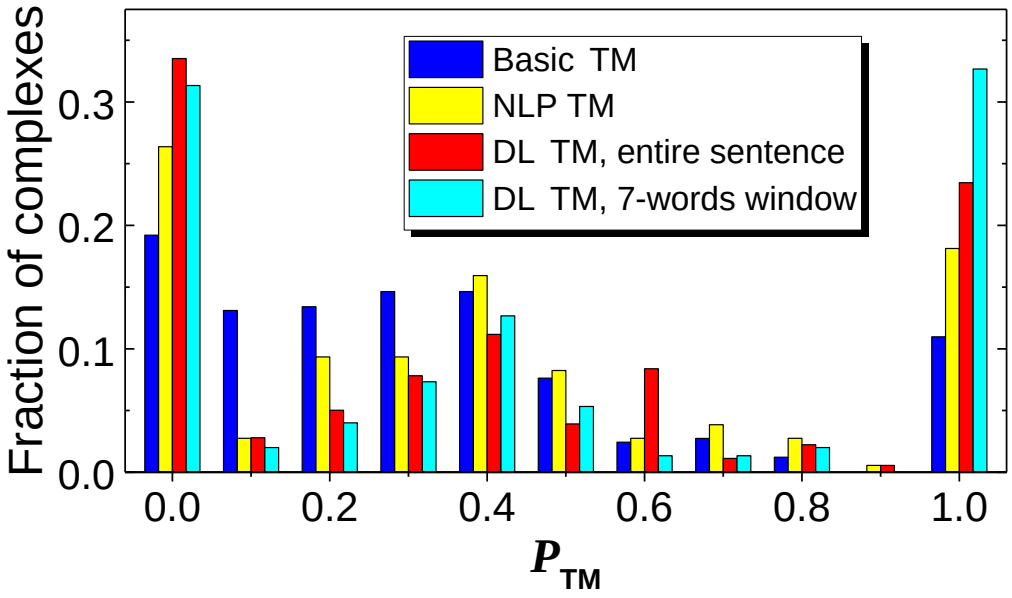


Figure 4-8: *Comparison of text-mining protocols on abstracts.*
*The PubMed abstracts were retrieved by the OR-queries with simplified residue filtering (basic TM) and with the residue filtering by NLP and DL. The NLP comprised SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automated keywords (**Table C-1**). DL consisted of DRNN model for classifying residues in the entire sentence and with 7-words window around the mined residues. The performance is calculated by Eq.(4-5). The distribution is normalized to the total number of complexes for which residues were extracted (**Table 4-3**).*

The DRNN training was done on the entire set of full-text articles. However, its performance only weakly depends on the size of the training set (**Table C-4** and **Figure C-7**). One can argue that the SVM performance suffered from the different structure of sentences in the full-texts and the abstracts. DRNN 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, NLP may be better in finer analysis of articles of the same group of authors with similar writing styles.

Table 4-3: *Overall TM performance on PubMed abstracts retrieved by OR-queries with simplified residue filtering (basic TM) and with residue filtering by NLP and DLs.*
NLP included SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automated keywords. DL consisted of Deep Recursive Neural Network model for classifying residues in the entire sentence, as well as using 7-words window around mined residues. NLP and DRNN were trained on complete full-text training set.

Method	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	–	–
NLP	182	135	31.4	23.3	74.1	-15	-3
DL (Whole sentence)	179	120	30.9	20.7	67.0	–3	+6
DL (7-words window)	150	104	25.9	18.0	69.3	–16	+13

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 Eq. (4-6) with values from basic TM (first row) as X2

Despite similar performance of NLP and DL methodologies, the latter 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, the sentiment accuracy may decrease. Our results indicate that the optimal TM performance is achieved when the sentiment is calculated for sentence fragments of 7 words around the residue (**Figure 4-9**). Overall, DRNN with the contextual window significantly improves filtering of the non-PPI residues, while only slightly

reducing the coverage of the dataset (**Table *4-3*** and **Figure *4-8***). **Figure *4-10*** illustrates the advantage of sentiment calculation using context window for cationic trypsin - tryptase inhibitor complex.

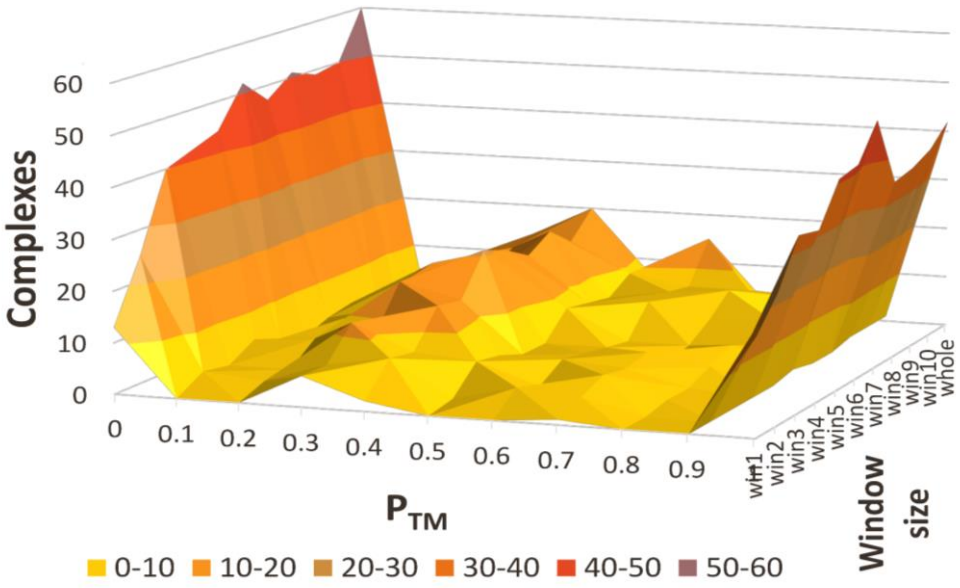


Figure 4-9: *TM performance with residue filtering by DL using different window sizes around mined residues.*
DL consisted of Deep Recursive Neural Network model for classifying residues in the PubMed abstracts, trained on the entire training set of PMC-OA full-text articles. The TM performance was calculated by Eq.(4-5).

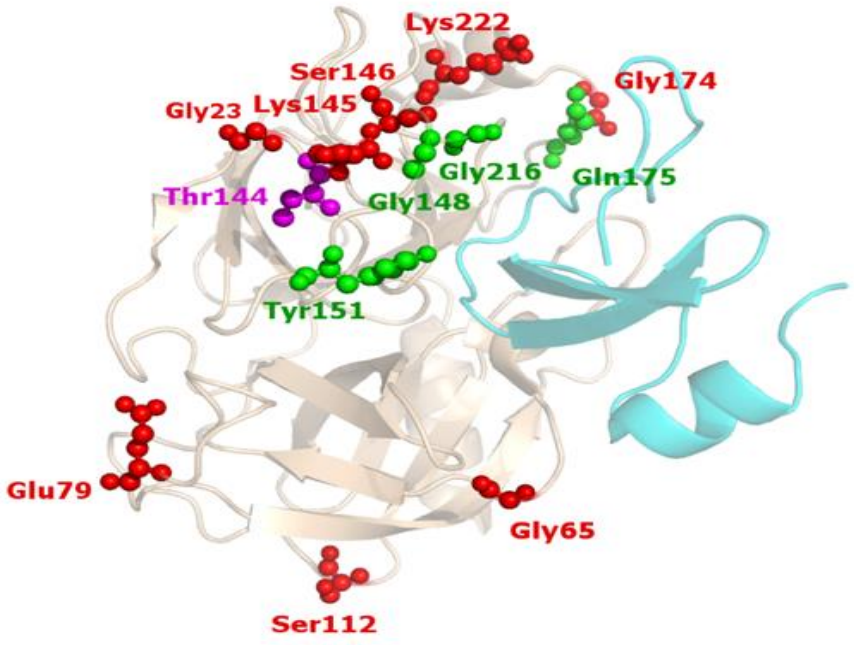


Figure 4-10: *Example of residues mined from abstracts by DL in full sentence and with 7-words window.*
The structure is 2uuy chain A (gray) and B (cyan). Residues correctly identified at the interface are in green, and incorrectly identified in red. DL using full sentence correctly identified 2 out of 3 residues (the incorrectly identified residue is in magenta). DL using window size of 7 correctly identified 2 residues (Gly216, Tyr151). Details are in ***Text C-3****.*

4.4 Concluding Remarks

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 text mining of PubMed abstracts [37], improved by natural language processing techniques [170], in this study, we focused on filtering non-interface residues from the list of initially mined residues in PubMed abstracts and PMC-OA subset of freely available full text articles by natural language processing and deep learning methods.

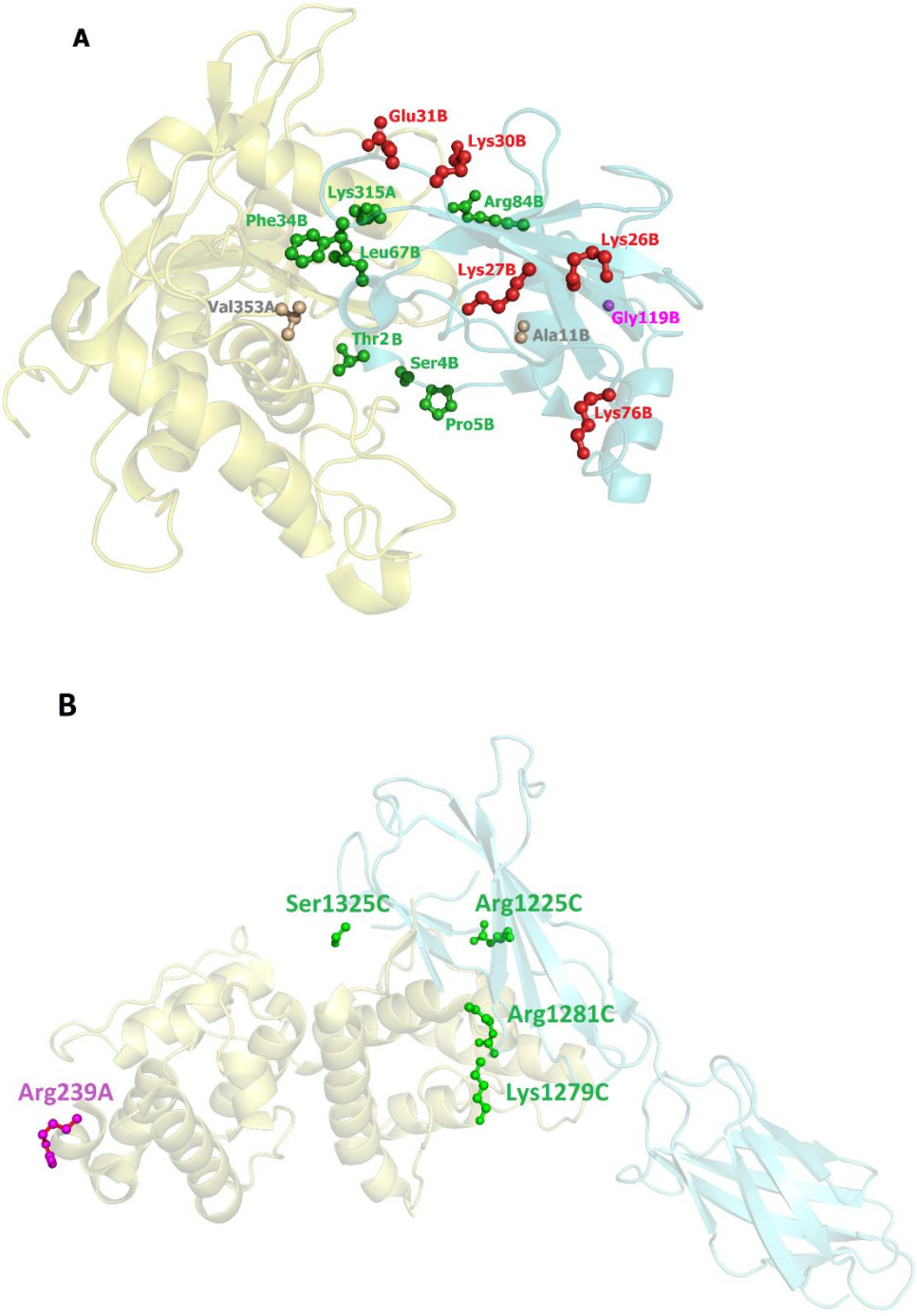
The PMC-OA full text articles, despite representing a small subset of all scientific publications, provide a useful source for training of Deep Recursive Neural Networks. The networks can be applied to classification of residues found in the abstracts, where the sentence structures are, in general, different from those in the full-text articles. In such case, DRNN 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 DRNN 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 DL-enhanced text-mining. Overall, following our previous results on NLP application to abstracts [170], we showed that DL similarly significantly outperforms the basic TM on the abstracts, and both NLP and DL significantly outperform the basic TM on 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.

Conclusion

The results of this work show that the text mining is a valuable tool for protein-protein docking. Incorporating NLP methodologies enhances the quality of the constraints. The results indicate that small specialized dictionaries are as effective as the generic word ontologies, such as WordNet, in distinguishing residues relevant to docking. Parse trees (Stanford parser) and SVM with domain relevant feature words provide the best performance. However, they run a risk of choosing features too specific to the training set. Word vectors together with deep learning for the sentiment analysis offer an elegant solution to this problem. The results show that freely available PMC-OA articles are a viable alternative to PubMed abstracts in generation of the docking constraints. Incorporating the context using the surrounding sentences further improves the performance. Limiting the context around the residues is desirable for distinguishing between multiple instances within a sentence. NLP and DL similarly significantly outperform the basic TM on the paper abstracts. Both NLP and DL significantly outperform the basic TM on the full-text papers. Greater availability of the full-text papers should increase the usefulness of this source of information for structural modeling of the protein complexes.

Overall, the study shows the value and the potential of the data-driven approaches for the future development of predictive methodologies for modeling of protein interactions. High-throughput modeling of large biomolecular systems requires powerful automated techniques. Computerized mining of the rapidly increasing number of biomedical publications is one such approach, the importance of which will grow with further development of the text mining methodologies, the greater output of scientific research, and wider public availability of the research outcomes.

Appendix A


A
Glu31B
Lys30B
Lys315A
Arg84B
Phe34B
Leu67B
Lys26B
Lys27B
Val353A
Gly119B
Ala11B
Thr2B
Ser4B
Pro5B
Lys76B
B
Ser1325C
Arg1225C
Arg1281C
Arg239A
Lys1279C

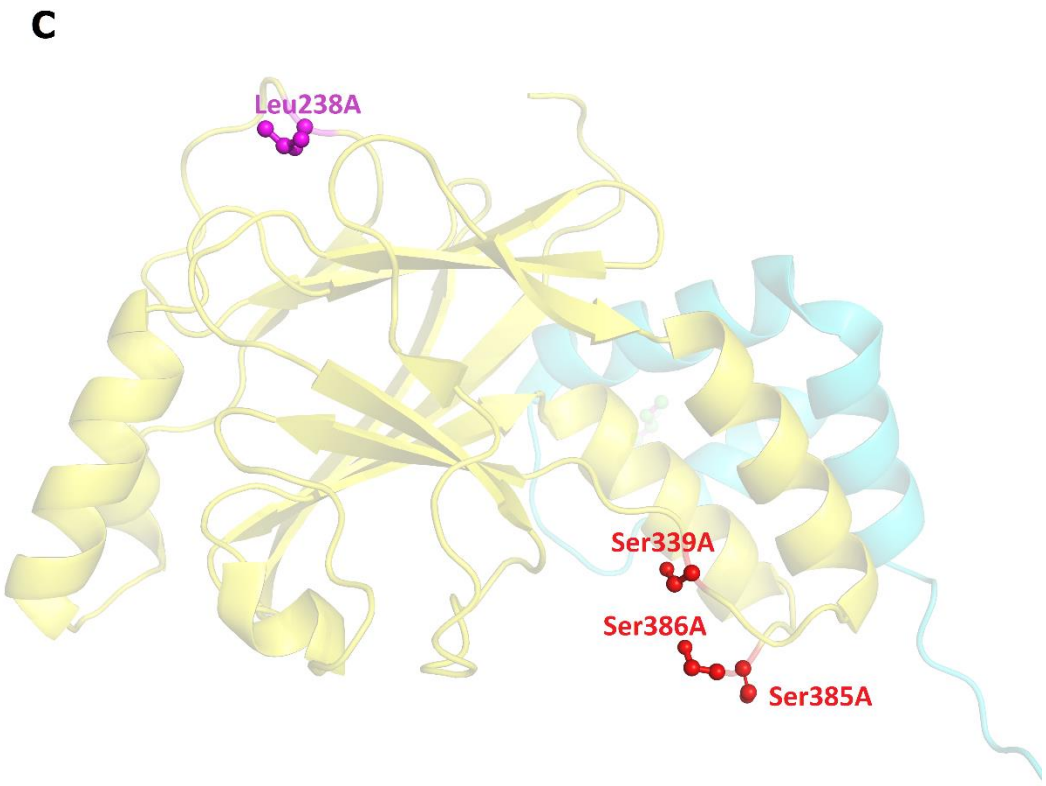


Figure A-1: *Examples of residues extracted by the basic TM.*
The structure, chain ID, and residue numbers are from 3cki (A), 3f7p (B), and 1zoq (C). Interface and non-interface residues detected by the AND-query are in green and red, respectively. Additional interface and non-interface residues detected by the OR-query are in brown and magenta, respectively.

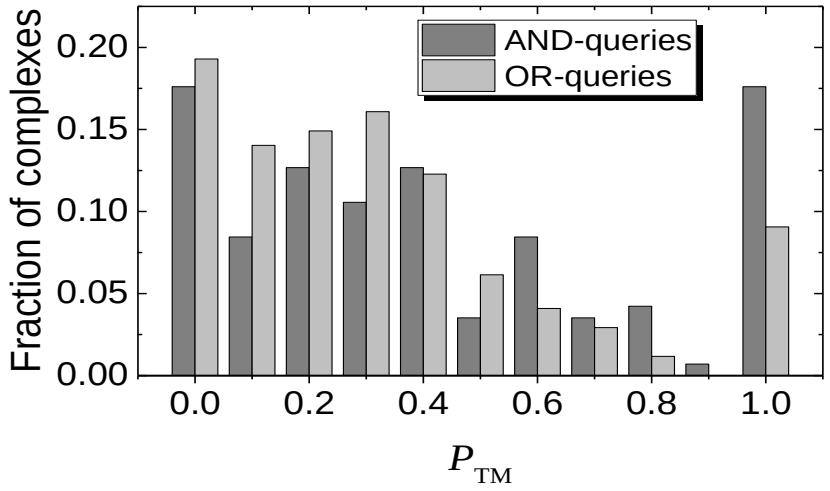


Figure A-2: *Distribution of complexes according to the quality of the basic TM, accounting for mismatch between residue numbering in PDB and UniProt sequences.*
*The TM performance is according to P_{TM} (Eq.(2-1)). The distribution is normalized to the total number of complexes for which residues were identified (column 3 in **Table 2-3**).*

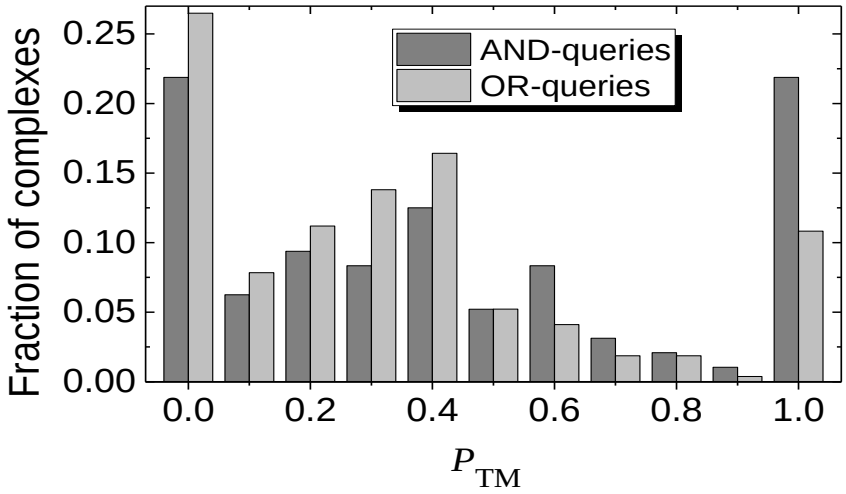


Figure A-3: *Distribution of complexes according to the quality of the basic TM, excluding abstracts published after the paper on the original PDB structure.*
*The TM performance is according to P_{TM} (Eq.(2-1)). The distribution is normalized to the total number of complexes for which residues were identified (column 3 in **Table 2-3**).*

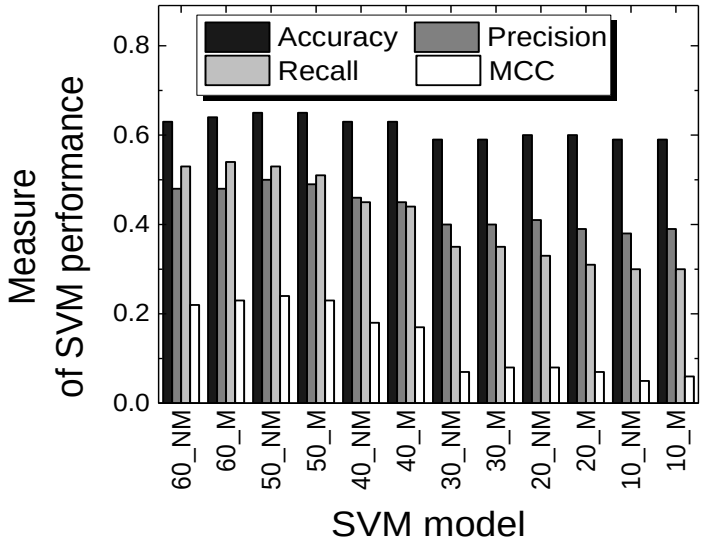


Figure A-4: *SVM performance for manual feature selection using linear kernel. Data with M suffix was obtained on abstracts excluding those with SVM-scores -0.05 to +0.05. Data with NM suffix was obtained on all abstracts.*

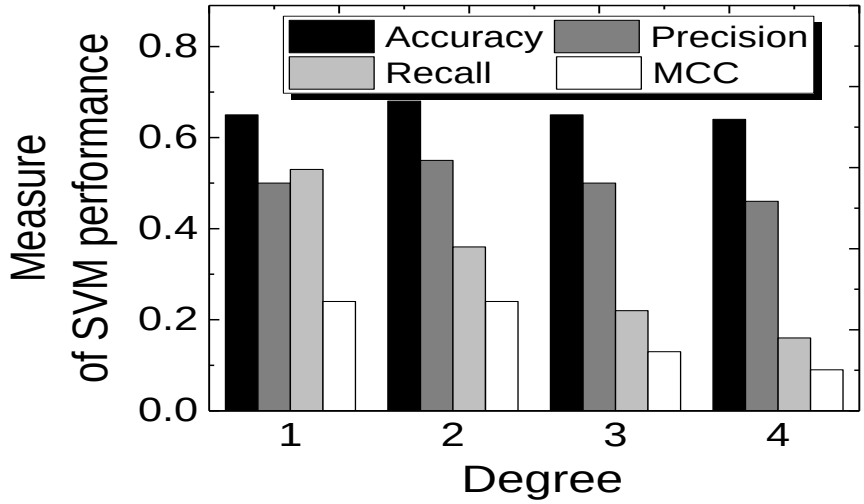


Figure A-5: *SVM performance for manual feature (50_NM) selection using polynomial kernel with different degrees.*

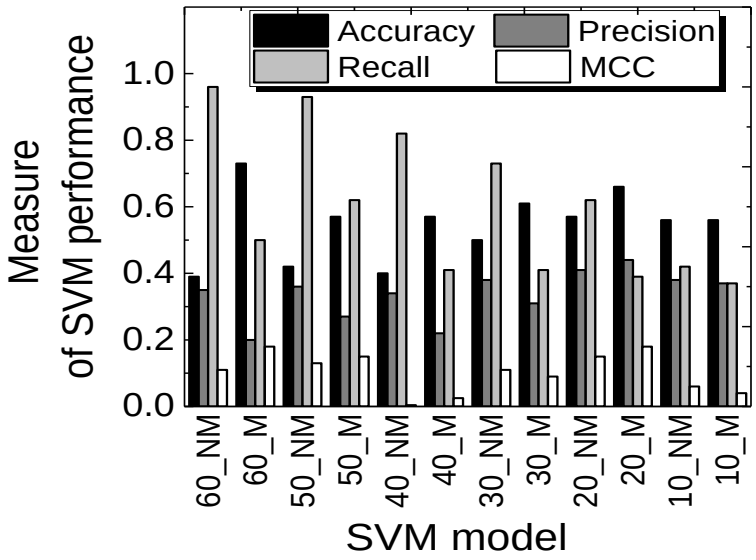


Figure A-6: *SVM performance for manual feature selection using RBF kernel with $\gamma = 1$. Data with M suffix was obtained on abstracts excluding those with SVM-scores -0.05 to +0.05. Data with NM suffix was obtained on all abstracts.*

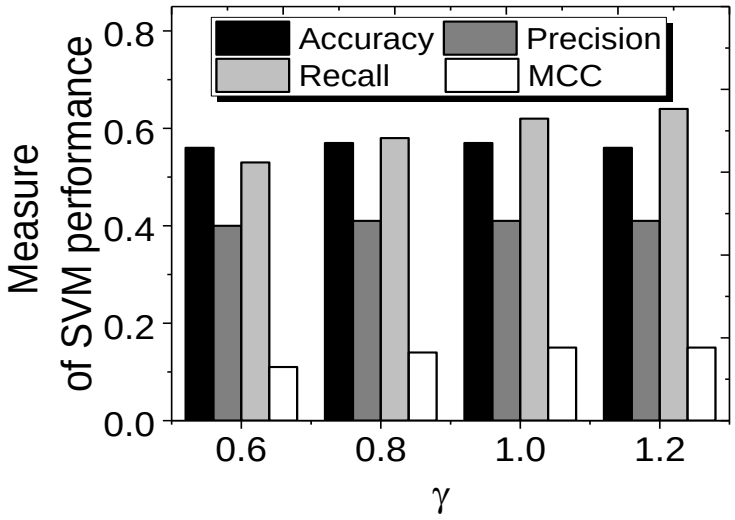


Figure A-7: *SVM performance for manual feature (20_NM) selection using RBF kernel with various γ.*

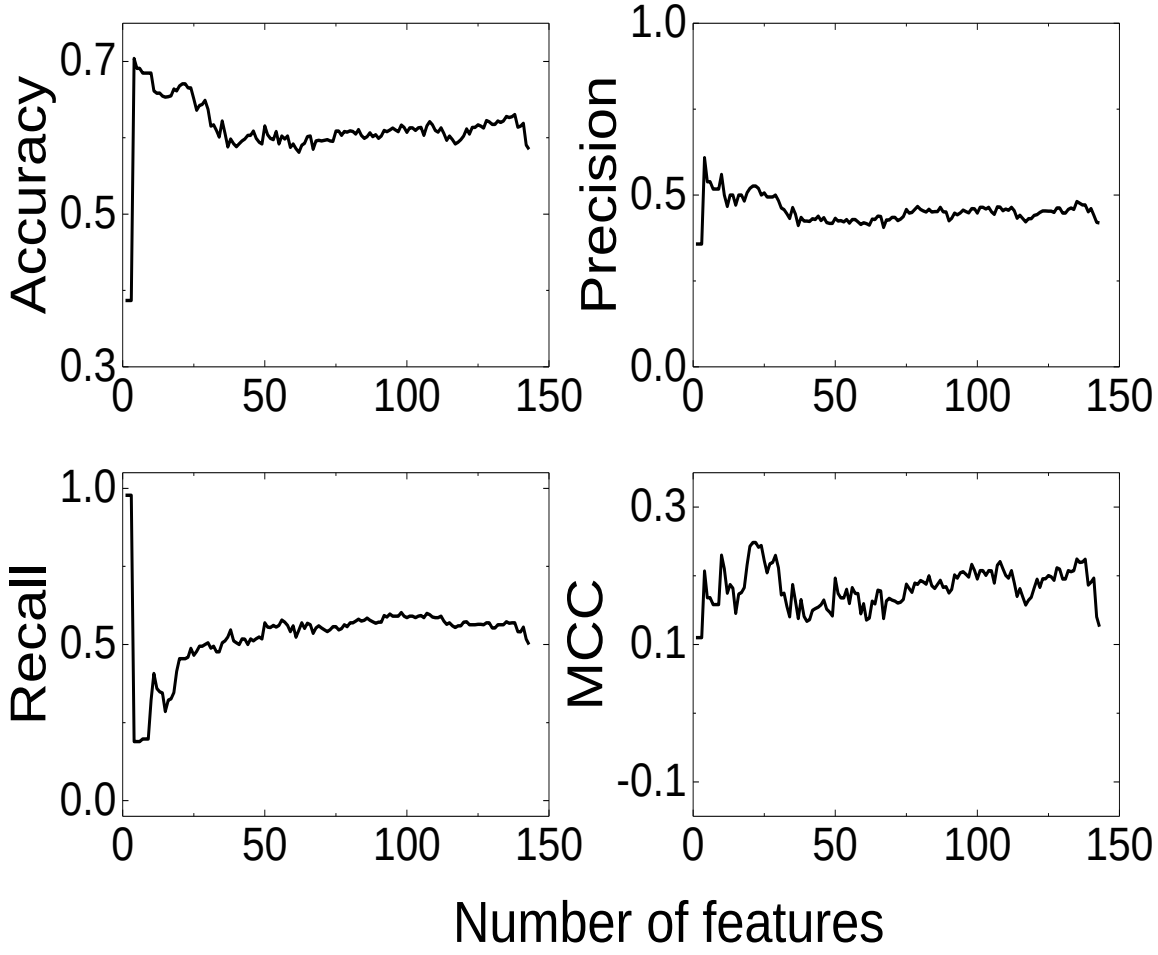


Figure A-8: *SVM performance for automated feature selection using linear kernel and 0.05 margin.*

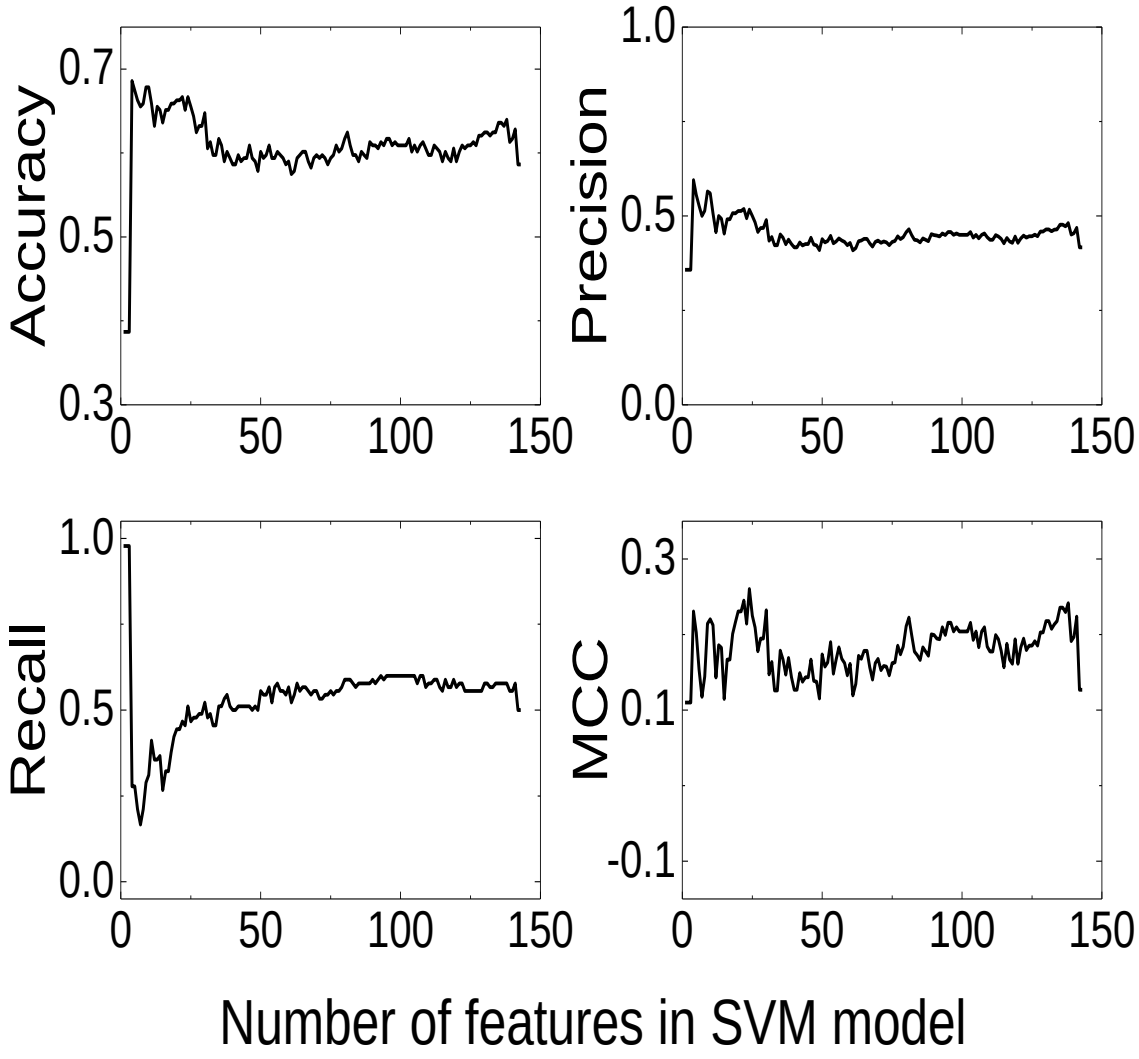


Figure A-9: *SVM performance for automated feature selection using linear kernel without margin.*

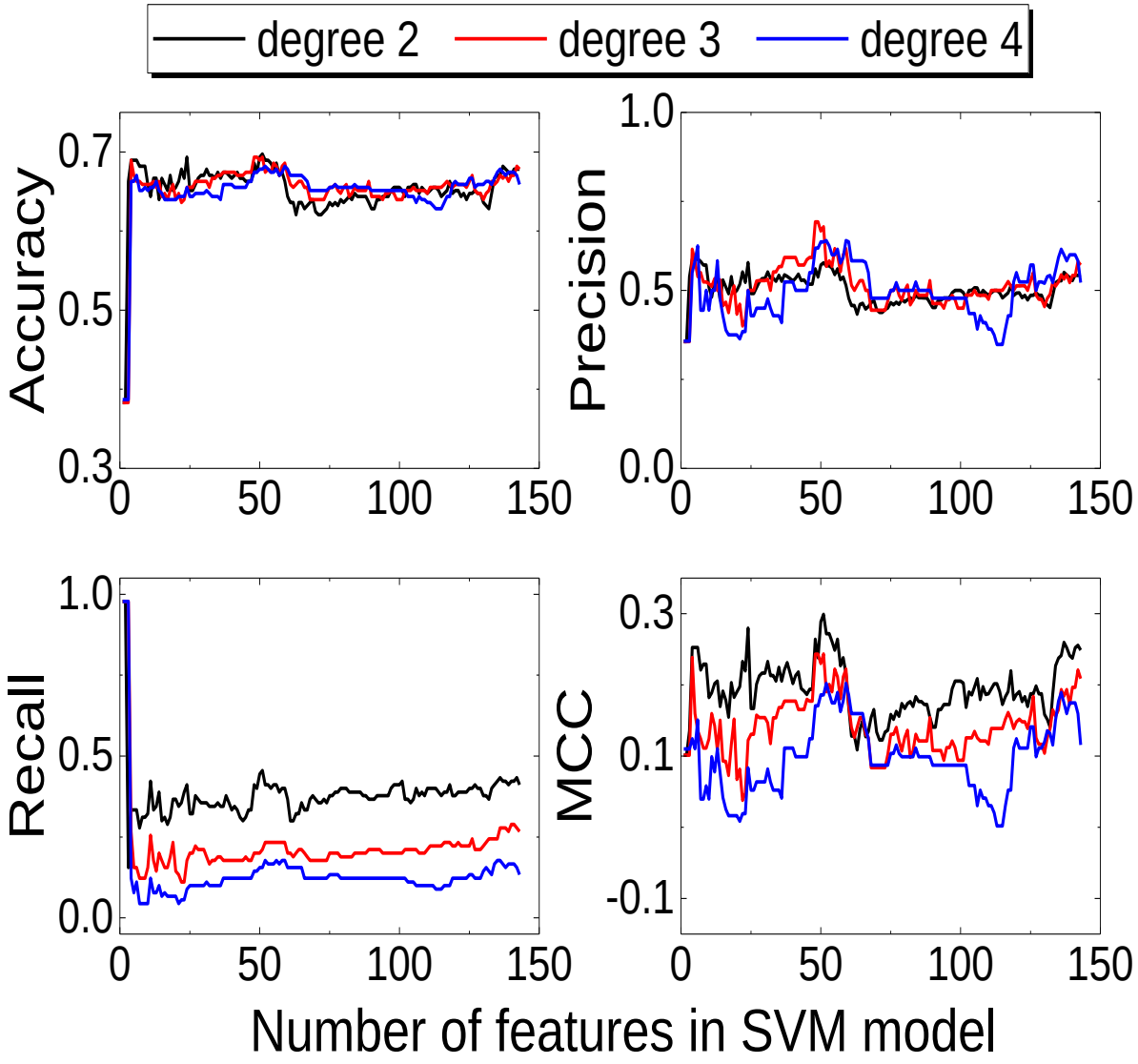


Figure A-10: *SVM performance for automated feature selection using polynomial kernel with different degrees and no margin.*

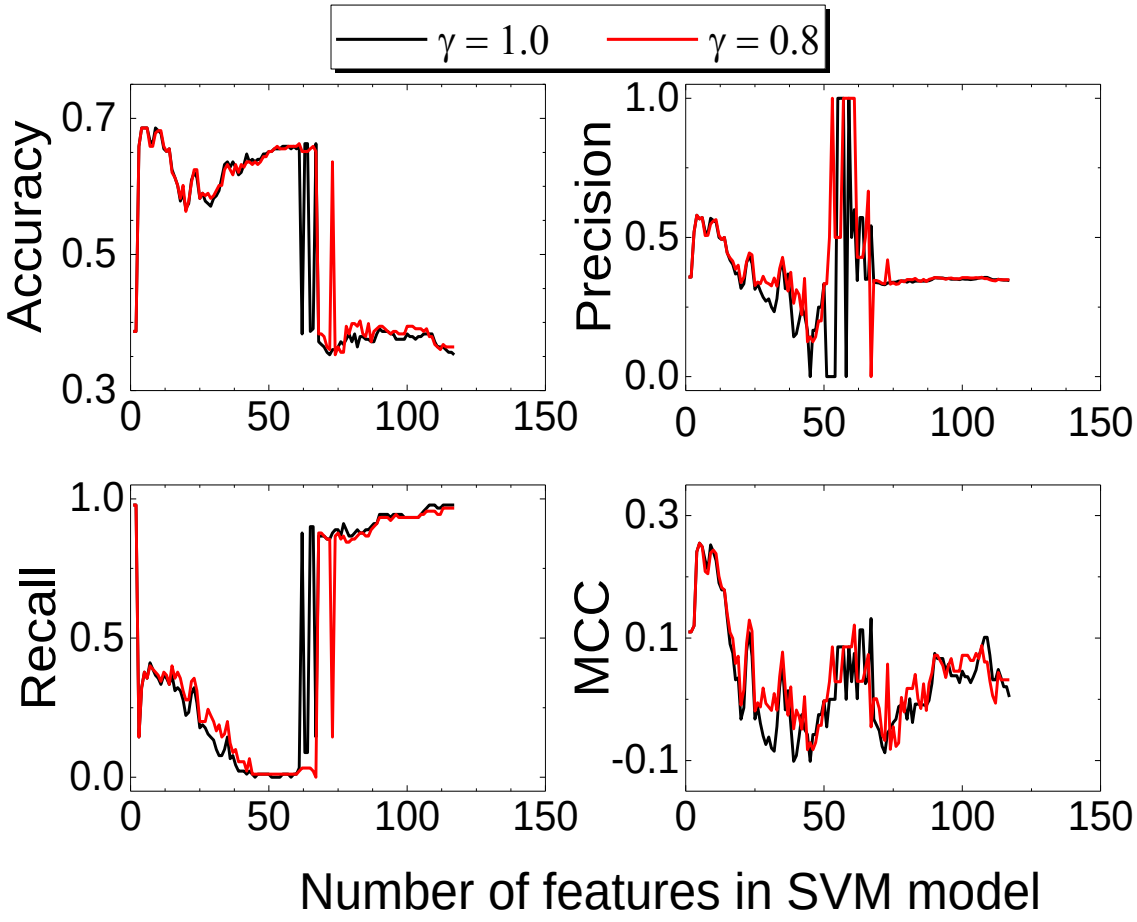


Figure A-11: *SVM performance for automated feature selection using RBF kernel with different γ and no margin.*

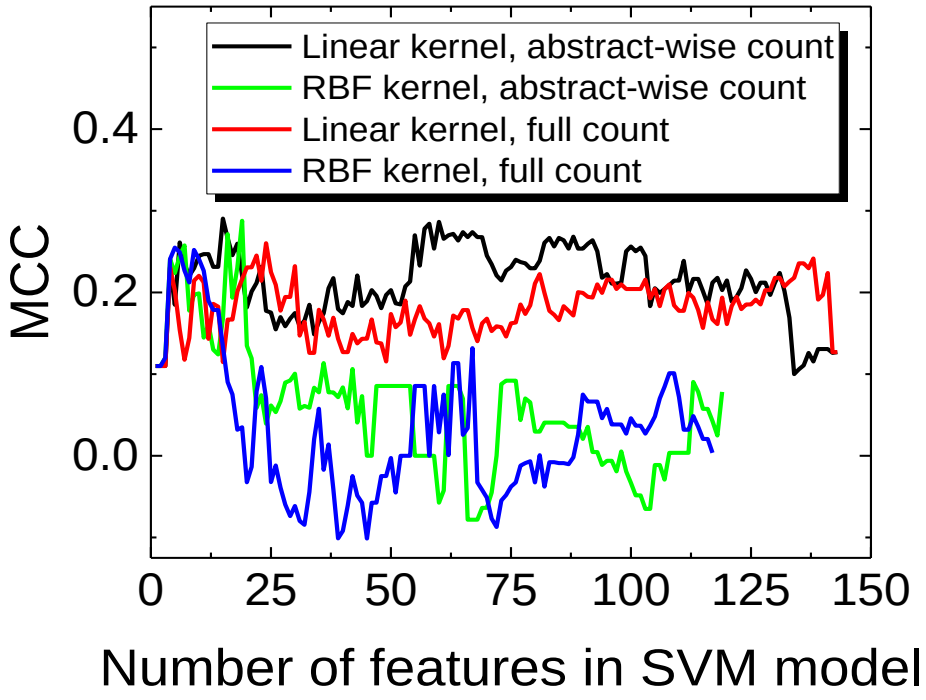


Figure A-12: *Comparison of Matthews correlation coefficient for different approaches to calculate the number of features in abstracts in the training set.*
The abstract-wise feature count (irrespective of the number of times the feature appears in an abstract) was used earlier in the extraction of features for prediction of protein function and localization (Wong & Shatkay, BMC Bioinformatics, 2013,14,Suppl.3: S14; Shatkay et al. Methods, 2015,74:54-64). The full count, used in the current study, accounts for all instances of a feature in an abstract. The data was obtained on the validation set of 261 abstracts. The SVM models were trained on 1,044 abstracts (see Methods).

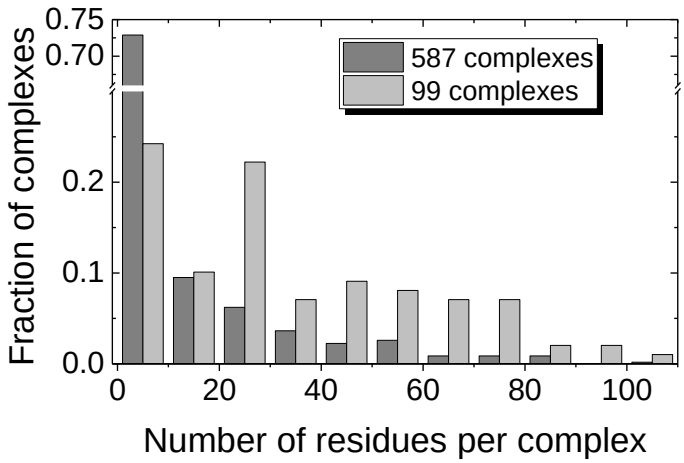


Figure A-13: *Distribution of total number of residues per complex extracted by OR-queries in two sets.*
The data is normalized by the total number of complexes in the corresponding set.

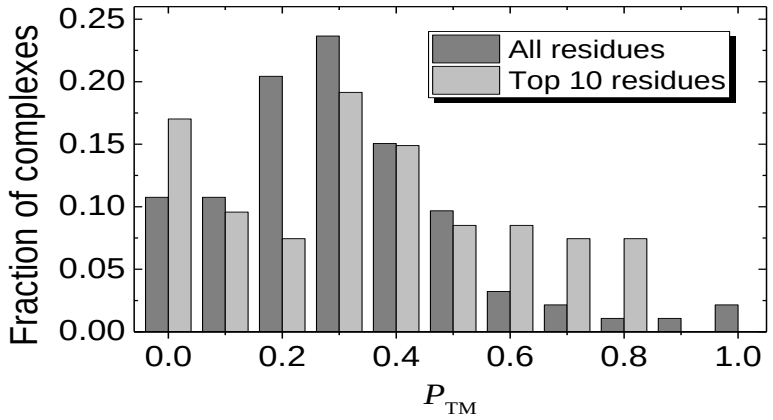


Figure A-14: *Normalized distribution of complexes in the Dockground benchmark set 3 according to TM performance, P_{TM} (Eq.(2-1)).*
*The data is obtained by the basic TM protocol with OR-queries, and is normalized to the total number of complexes, for which residues were predicted (column 3 in **Table 2-3**). Dark and light bars show distributions for all retrieved residues and for the top 10 residues, submitted to docking, respectively.*

Table A-1: *Examples of optimal SVM model impact on TM output.*
*The examples are from **Figure A-1**: Examples of residues extracted by the basic TM. A residue is considered removed if an SVM model filtered out all abstracts mentioning this residue.*

SVM model	PDB 3cki		PDB 3f7p		PDB 1m27		PDB 1zoq	
	CR [a]	IR [b]	CR	IR	CR	IR	CR	IR
MF50L	Gly119B	Pro5B Ala11B	Arg239A	Arg1225C Lys1279C Arg1281C Ser1325C	Thr53A	Tyr132C Trp119C	Ser339A	–
AF138L	Gly119B	Pro5B Val353A	Arg239A	–	–	Tyr132C	Ser339A, Ser385A, Ser386A, Leu238A	–
AF24L	Gly119B	Pro5B Ala11B Val353A	Arg239A	–	–	Tyr132C	Ser339A, Ser385A, Ser386A, Leu238A	–

[a] *Correctly removed (non-interface residues)*
[b] *Incorrectly removed (interface residues)*

Text A-1: *Performance of basic text mining for specific protein-protein complexes*

SH2D1A – p59Fyn complex (1m27). For this complex, AND-query did not retrieve any abstracts. The OR-query identified 6 abstracts with 4 residues, out of which 3 are at the interface (Figure 2-3 in the main text, P_{TM} =0.75). Arg78 of SH2D1A protein (1m27, chain A) was detected in the abstract on the role of tyrosine kinase Fyn and SLAM (synonym of SH2D1A) interaction in the development of natural killer T cells in human and mice [177]. The abstract was not detected by AND-query because p59Fyn names that do not follow UniProt nomenclature. Arg78 was also pinpointed in two mutagenesis studies on the role of the SLAM-SAP-Fyn signaling pathway in mice CD4 T cell function and germinal center development [178, 179]. Trp119 of p59Fyn (1m27, chain C) was detected in the abstract on interactions of SH2 and SH3 domains of p59Fyn [180]. The p59Fyn residue Tyr132 was detected in an abstract of site-directed mutagenesis studies that prove importance of this residue in formation of phosphoprotein pp21 [181]. Non-interface SH2D1A residue Thr53 was found in the abstract of computational study on the role of SLAM mutations in manifestation of an immunodeficiency disease, X-linked lymphoproliferative syndrome [182]. Another residue mentioned in this abstract (Arg32) belongs to the protein core and thus did not pass protocol filters.

TACE – TIMP3 complex (3cki). For human tumor necrosis factor alpha converting enzyme (TACE) co-crystallized with the metalloproteinase inhibitor 3, TIMP3, the AND-query predicted 7 interface and 5 non-interface residues (Figure A-1A, P_{TM} = 0.58), mentioned in the abstracts of 6 publications [183-188], other than the original X-ray study [189]. Redesign of metalloproteinase inhibitor TIMP1, using TIMP3 as a scaffold identified three residues, Ser4, Leu67, and Arg84, important for the TIMP3-TACE binding [183]. Ser4 and Thr2 were found to be functionally important by measuring binding affinities of the mutated TIMP3 [184]. Redesign of

metalloproteinase inhibitor TIMP2, using TIMP3 and TIMP1 as scaffolds [185], identified Ser4 and, additionally, Phe34 residues. Three other residues mentioned in that abstract (Val/Leu69, Thr/Leu98 and Leu/Ile/Met100) did not match residues in the original PDB (Gly69, Val98, and Arg100, respectively). Phe34 was also mentioned in the original X-ray crystallography paper [189]. In the study of TACE cysteine-rich domains role in the TIMP-3 inhibitory potency [186], Lys315 was mentioned as a residue adjacent to the TACE catalytic site, which interacts with TIMP-3 Glu31, which is close to, but not at the TACE-TIMP3 interface. In this abstract, other non-interface residues, Lys26, Lys27, Lys30, and Lys76 were also mentioned along with residues (Glu26, Glu27, Glu30, and Glu76) that either did not match PDB numbering or did not follow residue patterns considered in this study (**Table 2-1**). Finally, Pro5 is identified in the analysis of expression levels of different human prostatic tumor cell lines [187]. However, Pro5 in this publication stands for the name of cell line rather than a residue. Ser4 was also identified by another publication [185] where the binding affinity of the mutant is better than TIMP-3. All the above residues belong to TIMP3 chain in the original PDB, except Lys315, which belongs to the TACE chain.

The OR-query for this complex found three additional abstracts [85, 190, 191] with two additional interface residues (TACE Val353 and TIMP3 Ala11) and one non-interface (TIMP3 Gly119) residue (P_{TM} = 0.60). Val353 was identified as functionally important in the study on stabilization of the TACE autoproteolysis [190]. Ala11 is mentioned in the abstract of paper [85] showing that activation of brain ET(B) receptors causes TIMP-1 and TIMP-3 production. Gly119 is present in the abstract of study [191] on mutated growth hormone (bGH) in transgenic mice. This abstract was picked due to the presence of the TIMP-3 name, however Gly119 mentioned therein belongs to bGH protein and just accidentally coincides with the Gly119 residue in the original PDB file.

The OR-query also found three additional abstracts with five residues already picked up by the AND-query. Study [192] determined that Lys26, Lys27, Lys30, and Lys76 constitute another TIMP3 binding site with the extracellular matrix (ECM). Other residues mentioned in that abstract (Arg163 and Lys165) belong to the C-terminal of TIMP3 not present in the original PDB file. Lys27 was also present in the abstract of study [193] suggesting that EZH2 (Enhancer of zeste homolog 2) accelerates lung cancer cell migration partially through repression of TIMP-3 expression. Thr2 was identified in the paper [194] showing that mutated N-TIMP3 inhibits degradation of ADAMTS-4 and ADAMTS-5 metalloproteinases.

Complex of Plectin-1 and Integrin beta-4 (3f7p). For this complex, AND-query found 3 abstracts with 4 interface residues of Integrin beta 4 (Figure A-1B, P_{TM} = 1.00). Abstract in Ref. [195] states that mutations at Arg1225 and Arg1281 sites inhibit interaction of the integrin beta 4 with plectin while mutation of Lys1279 has no effect to recruit plectin. In addition, Arg1281 was spotted in the abstract in Ref. [196] also showing that mutation at this site affects the interaction between the two proteins. Another mutagenesis study [197] shows that residue Ser1325 is important for recruitment of plectin into hemidesmosomes *in vivo*.

The OR-query for this complex identified one additional abstract with one additional non-interface residue (Figure 2-3B) thus reducing TM performance to P_{TM} = 0.80. Arg239 was found in the abstract in Ref. [198] on interaction between plectin and glial fibrillary acidic protein (GFAP). However, the number of this residue belonging to GFAP protein, just accidentally coincides with the number of one of the arginine residues in the original PDB file. The OR-query has also retrieved 2 additional abstracts of mutagenesis studies containing Arg1281 pinpointed by the AND-query. Mutation at Arg1281 was found to effect severity of epidermolysis bullosa

(genetic skin disease) [199] and to inhibit interaction between plectin and integrin beta 4 with its alpha 6 chain [200].

IRF3 – CBP complex (1zoq). For this complex, the AND-query identified 6 abstracts with 3 non-interface residues in the Interferon regulatory factor 3, IRF3 (Figure A-1D, $P_{TM} = 0.00$). All found residues (Ser339 [201, 202], Ser385 [203-205], and Ser386 [203-206]) were studied in the context of their phosphorylation, which regulates CBP binding allosterically. Interferon interacts with many partners (e.g., BioGrid database [207] lists 44 interactions for IRF-3) and phosphorylation of these three residues was studied in the context of IRF3 binding to other proteins as well. The OR-query thus found Ser336, Ser385 and Ser386 residues in one [208], two [209, 210], and five [210-214] additional abstracts, respectively.

The OR-query for this complex found one more residue, Leu238 of IRF3, in one additional abstract ($P_{TM} = 0.00$) of study on the interferon (IFN) role in resisting evasion of the African swine fever virus into pig immune system [215].

Text A-2: *Performance of SVM-enhanced text mining for specific protein-protein complexes*

For the TACE-TIMP3 complex (3cki, Figure A-1A), all three models removed non-interface Gly119 and one or both interface residues brought up by the OR-query (**Table A-1**). In addition, all models removed one interface residue from the AND-query abstracts (Pro5B) and overlooked all non-interface residues found by the AND-query. This led to slightly deteriorated TM performance (P_{TM} drops by 0.02 for the MF50L and AF138L models and by 0.06 for AF24L model). For the Plectin 1 and Integrin beta 4 complex (3f7p, Figure A-1B), the AF138L and AF24L models removed plectin non-interface Arg239, detected by the OR-query, thus increasing P_{TM} to 1.00. The MF50L model, however, also removed all interface residues detected by the AND-query, thereby erroneously excluding this complex from the consideration (**Table A-1**). Still, for the SH2D1A-p59Fyn complex (1m27, Figure A-1C), the MF50L model raised P_{TM} to 1.00 but, in contrast, the AF138L and AF24L models performed worse, dropping P_{TM} to 0.67 (**Table A-1**). These models purged p59Fyn interface residue Tyr132. Finally, the AF138L and AF24L models correctly excluded IRF3-CBP complex (1zoq, Figure A-1D) from consideration removing all non-interface residues picked up by the OR-query while the MF50L model succeeded only partially, removing only one non-interface residue (**Table A-1**). This model overlooked other non-interface residues.

Appendix B

Table B-1*: Details of residue filtering by the SVM-based approach of the parse-tree analysis for SUMO-conjugating enzyme UBC9 complex with small ubiquitin-related modifier 1 (2uyz, chains A and B).*
For this complex, the AND-query did not retrieve any abstracts. The OR-query identified 5 abstracts and 5 residues have passed the initial filters of the basic TM protocol (see Methods in the main text), out of which 1 is at the interface (Figure 5 in the main text, PTM =0.2). All five residues belong to the small ubiquitin-related modifier 1. Gln69 was detected in the study which pointed out that this residue is responsible for not forming polymers unlike ubiquitin [216]. Glu67 (chain B) was mentioned in the context with interaction of this protein with cytosolic dipeptidyl peptidase 9 (DPP9) [217]. Arg54 (chain B) was noted in the context of fusing sumoylation-site Tec1 mutant to Ubc9 and alteration of transcription activity [218]. Lys37 (chain B) was detected in the mutagenesis studies on ultraspiracle protein (Usp) fragments and its effect on sumoylation [219]. Lys39 (chain B) was named in the context of phosphorylated residues contacting this residue in SUMO1 thus connecting CK2 signaling [220].

PMID of abstract (Residue)	Sentence	+ve/*-ve* words	S_x	Interface?
23152501 (Glu67)	Surprisingly, DPP9 binds to SUMO1 independent of the well-known SUMO interacting motif, but instead interacts with a loop involving Glu(67) of SUMO1.	binds, interacting, interacts/	0.70	Yes
19826484 (Arg54)	In contrast, fusing sumoylation-site mutant Tec1, i.e., Tec1(K54R), to Ubc9 did not significantly alter transcriptional activation and had a less effect on invasive growth.	***/sumoylation***	-0.33	No
22676916 (Lys37)	Mutagenesis studies on the fragments of Usp indicated that sumoylation can occur alternatively on several defined Lys residues, i.e. three (Lys16, Lys20, Lys37) in A/B region, one (Lys424) in E region and one (Lys506) in F region.	***/sumoylation***	-0.16	No
19217413 (Lys39)	We provide evidence that the phosphorylated residues contact lysine 39 and 35 in SUMO1 and SUMO2, respectively.	contact/***phosphorylated***	0.00	No
9654451 (Gln69)	Furthermore, ubiquitin Lys48, required to generate ubiquitin polymers, is substituted in SUMO-1 by Gln69 at the same position, which provides an explanation of why SUMO-1 has not been observed to form polymers.	/***ubiquitin***	-0.33	No

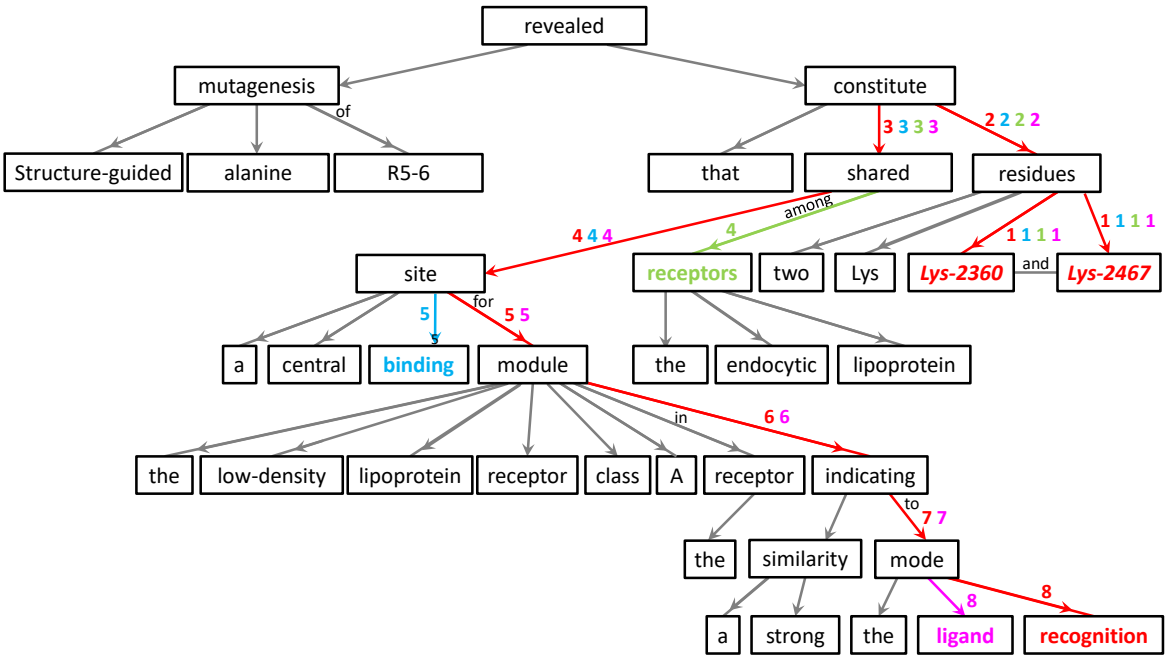


Figure B-1: *Parse tree of sentence "Structure-guided alanine mutagenesis of R5-6 revealed that two Lys residues (Lys-2360 and Lys-2467) constitute a central binding site for the low-density lipoprotein receptor class A module in the receptor, indicating a strong similarity to the ligand recognition mode shared among the endocytic lipoprotein receptors" from the abstract with PMID:17548821. Four PPI+ive words from our own dictionary are in red, magenta, blue and green. Colored numbers are index number of the edge in the path connecting mentioned residues (italic, in red) with the keyword of corresponding color. In this example, the scores of both residues (Equation (3-3) in main text) are* $S_X = \frac{1}{5} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} = 0.7$ *and they both are at the interface of 3a7q chains A and B.*

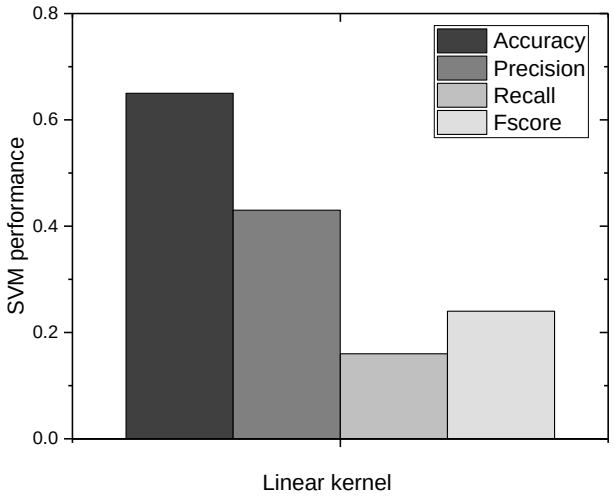


Figure B-2: *SVM performance using linear kernel. Data with no margin was obtained on all sentences.*

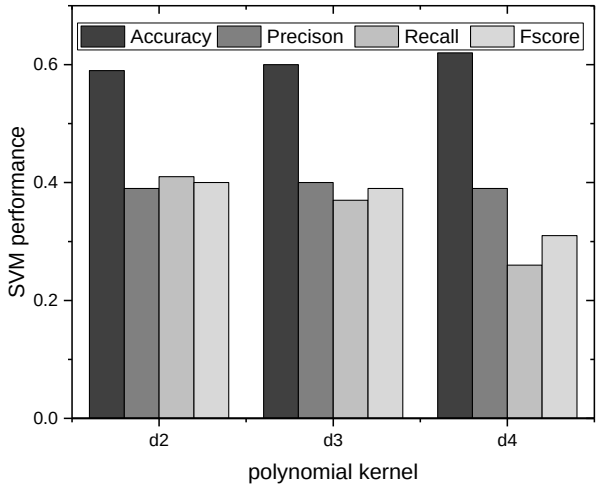


Figure B-3: *SVM performance using polynomial kernel with different degrees and no margin.*

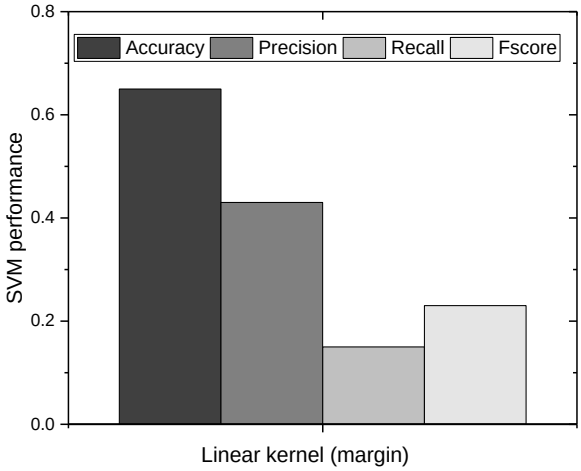


Figure B-4: *SVM performance using linear kernel and 0.05 margin.*
Data with margin was obtained on sentences excluding those with SVM-scores -0.05 to +0.05.

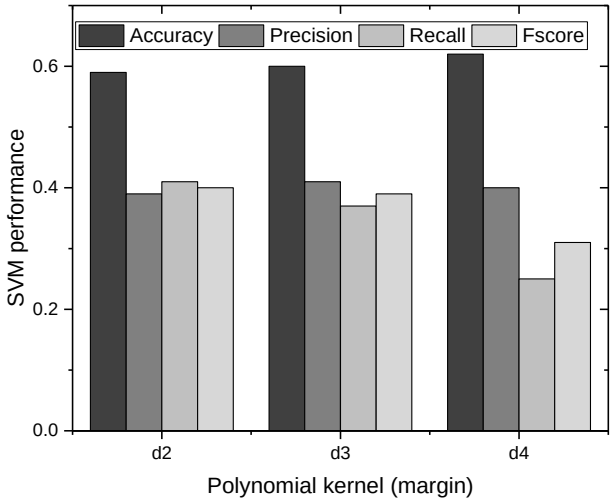


Figure B-5: *SVM performance using polynomial kernel with different degrees and 0.05 margin.*
Data with the margin was obtained on sentences excluding those with SVM-scores -0.05 to +0.05.

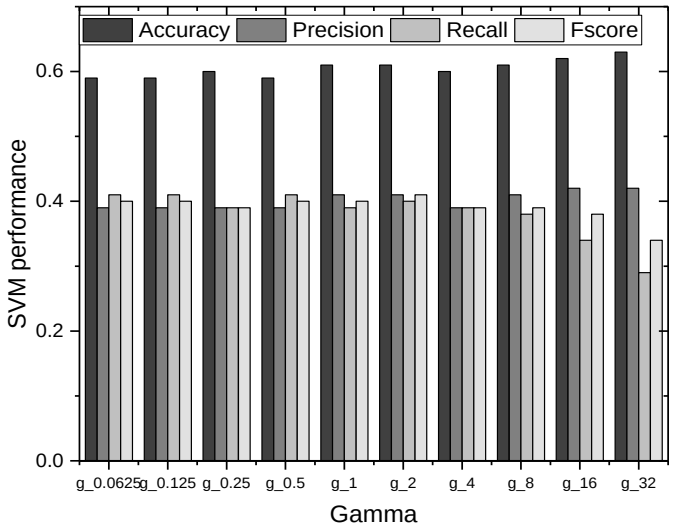


Figure B-6: *SVM performance using RBF kernel with different γ and no margin.*

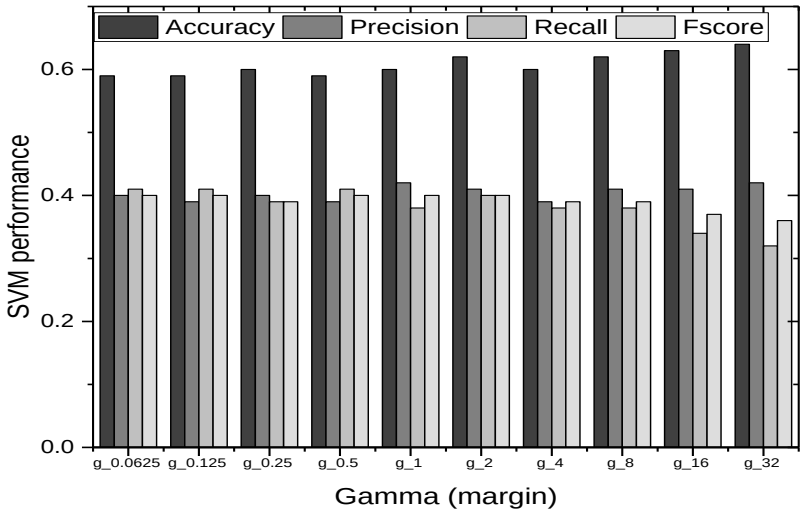


Figure B-7: *SVM performance using RBF kernel with different γ and 0.05 margin.*
Data with the margin was obtained on sentences excluding those with SVM-scores -0.05 to +0.05

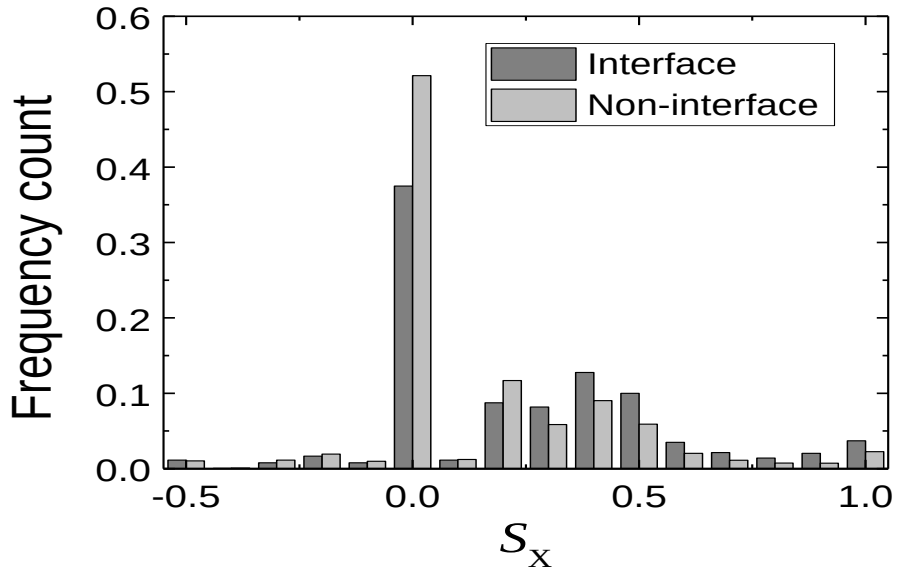


Figure B-8: *Normalized distribution of S_X scores (Eq. (3-3) in the main text) for 1921 interface and 3865 non-interface residues.*
The data was obtained from the parse trees of 5786 sentences of 3109 abstracts on studies of 579 complexes. Residues were spotted for 328 complexes and for 273 of them at least one found residue was at the interface.

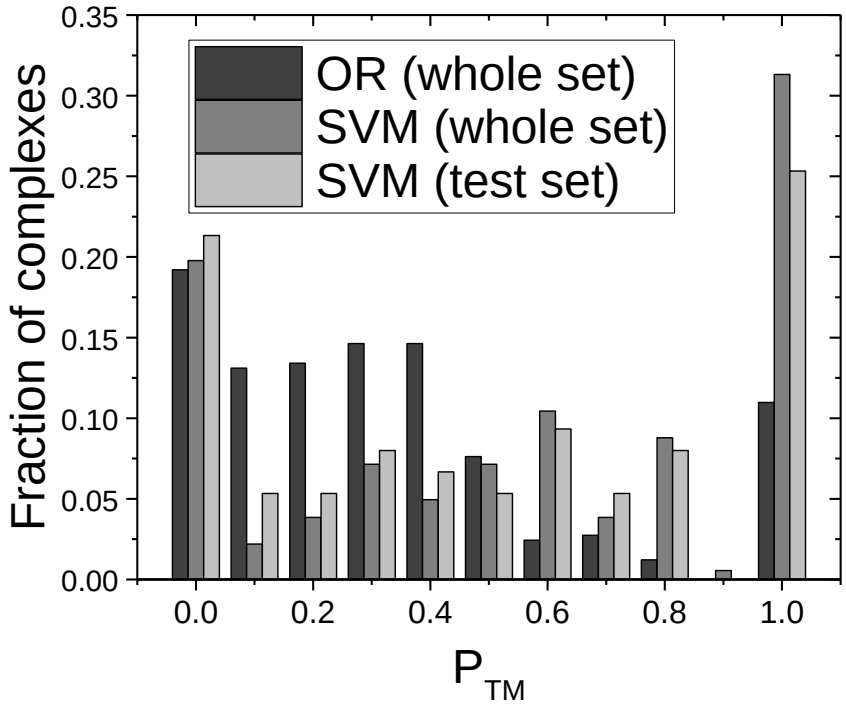


Figure B-9: *Performance of the basic and the advanced text mining protocols.*
Advanced filtering of the residues in the abstracts retrieved by the OR-queries was performed by analysis of sentence parse trees using SVM model on the entire and the reduced sets of abstracts. The TM performance is according to PTM (Eq. (3-1)). The distribution is normalized to the total number of complexes for which residues were identified (328, 182 and 75 for the OR whole set, SVM whole set and SVM test set, respectively).

Appendix C

Text C-1: *for **Figure 4-5** (main text)*

Residue filtering by basic TM for the complex of reelin (chain A) and low-density lipoprotein receptor-related protein 8 (chain B) of 3a7q identified from PubMed abstracts and not from PMC–OA full text. Only AND-query identified 3 PubMed abstracts. Three residues passed the initial filters of the basic TM (see Methods in the main text), of which 2 are at the interface (P_{TM} = 0.66). All three residues belong to reelin (chain A of 3a7q). Direct citation of PDB entry identified Lys2467 and Lys2360 (both in chain A). These residues play an important role in interaction of reelin with apolipoprotein E receptor 2 (ApoER2) [221]. Structure-guided alanine mutagenesis of fifth and sixth reelin repeats (R5-6) identified that residues Lys2467 and Lys2360 (both in chain A), are part of central binding site for low-density lipoprotein receptor [222]. Another residue Ala2101 (chain A) was identified in mutant reelin where this mutant failed to assemble into multimers via disulfide bonds. However, it non-covalently associated high molecular weight oligomeric states in solution. The binding assay (surface plasmon resonance) showed that this mutant retained binding capability towards low density lipoprotein receptor [223].

Text C-2: *for **Figure 4-7** (main text)*

Residue filtering by basic TM for the complex of heat shock protein HSP82 (chain A) and AHA1 (chain B) of 1usu identified one residue from PubMed abstract. PMC (OA) abstracts did not identify any residues, while PMC (OA) full-text identified six residues. The residue identified by PubMed is not at the interface ($P_{TM} = 0$). Four out of six residues identified by PMC (OA) full text are at the interface ($P_{TM} = 0.66$). NLP-hybrid method on PMC (OA) full-text further filtered the residues by identifying three residues, all of which are at the interface ($P_{TM} = 1$). All residues were identified by OR-queries.

TM on abstract from PubMed identified Glu90 in mutant study of Thr90 in Hsp90α when investigating the phosphorylation impact of this residue, showing that Thr90 is involved in the regulation of the Hsp90α chaperone machinery [224] ($P_{TM} = 0$).

TM on PMC (OA) full-text articles identified six residues. The abstract of this full-text article, where the residues Leu66, Ile64 and Phe100 were identified as PPI-related using basic TM, is about a method to photo-cross-link interacting proteins using p-azido-L-phenylalanine (pAzpa). The non-canonical amino acid pAzpa was incorporated into a domain of Aha1 that was known to bind Hsp90 in vitro [225]. The abstract of the full-text article from which residue Asp145 was retrieved using basic TM is on the analysis of SGT1–HSP90 (Suppressor of G2 allele of skp1 and Heat-shock protein 90). The full text mentions this residue as a target for site-directed mutagenesis in HSP90 in wheat [226]. Asp53 was identified in the full-text article on structural study of Aha1 binding with Hsp90 to modulate ATP hydrolysis cycle and client activity in vivo. When this residue is mutated in N-terminal domain in yeast Hch1p it impairs the ability to stimulate Hsp90 [227]. The abstract where the residue Glu22 was identified by basic TM mentions the results of computational study of allosteric regulation in Hsp90 complexes with p23 and Aha1 as the co-

chaperones. NLP-hybrid on full text correctly classified three interface residues (Leu66, Ile64 and Phe100). It rejected 3 residues (Asp145, Asp53 and Glu22) of which one (Asp53) was at the interface. Deep learning on full text yielded results identical to those of NLP-hybrid. DL with 7 words window accepted two residues which are at the interface (Leu66 and Ile64).

Text C-3: *for **Figure 4-10** (main text)*

Residue filtering by basic TM for the complex of Cationic Trypsin (chain A) and Tryptase Inhibitor (chain B) of 2uuy identified 13 residues from 12 PubMed abstracts. Only 4 of 13 residues were correctly identified at the interface ($P_{TM} = 0.30$). DL using whole sentence identified 3 residues, of which only 2 were at the interface ($P_{TM} = 0.66$). DL using window of 7 identified 2 residues, both at the interface ($P_{TM} = 1$). All residues were identified by OR-queries.

Basic TM identified correctly Gly216 in a structural analysis of trypsin-BPTI interfaces [228]. Tyr151 was correctly identified by basic TM in trypsin forming hydrophobic interface in investigation of molecular specificity of Kunitz domain 1 (KD1) of issue factor pathway inhibitor-2 [229]. Tyr151 was also identified in another abstract where crystal structure of trypsin were compared between different organism such as Atlantic salmon, chum salmon and bovine [230]. Tyr151 again was correctly, and Ser146 incorrectly, identified in PubMed abstract where the residue is part of substrate activation binding site of bovine trypsin [231].

Glu79 was incorrectly identified at interface by basic TM in the abstract describing E79K mutation in cationic trypsin causing increase in transactivation of anionic trypsinogen and used in-vitro analysis of recombinant wild and mutant enzymes [232]. Gly65 and Gly23 were incorrectly identified by basic TM from an abstract that describes the specificity of papaya proteinase IV (PPIV) for cleaving glycyl bonds [233]. Of Thr144 and Gly148 identified by TM only the latter is at the interface. The abstract is on the structure of complex of bdellastasin and porcine beta-trypsin [234]. Of Gly174, Gln175 and Gly216 identified by TM only the last two were correctly determined to be at the interface. The abstract is about a comparative study of structures of cyclotheonamide A (CtA) complexes of alpha-thrombin and beta-trypsin [235]. Ser112 was

incorrectly identified by basic TM at interface, from abstract about supercharged mutant (variant) of serine protease human enteropeptidase light chain [236].

Lys145 and Ser146 were incorrectly identified by basic TM at the interface, in abstract about a crystal structure of an active autolysate form of porcine alpha-trypsin (APT) [237]. Lys145 and Ser146 were also identified as an autolysis position in the abstract where the crystal structure of Porcine epsilon-trypsin was studied by molecular replacement [238]. Lys222 was incorrectly identified by basic TM from an abstract about covalent binding of proteinases by human alpha 2-macroglobulin [239].

DL using full sentence identified 3 residues (Tyr151, Thr144 and Gly148), of which 2 (Tyr151, Gly148) were correctly determined at the interface. Thr144 is incorrectly identified at the interface. DL using window size 7 identified 2 residues (Gly216, Tyr151), both at the interface.

Table C-1: *List of automatically generated keywords and associated sentiment labels. Positive and negative scores indicate keywords relevant (PPI+ive) and irrelevant (PPI-ive), respectively, to protein-protein binding sites. Details are in Methods (main text).*

Score	Keyword	Sentiment	Score	Keyword	Sentiment
-6.2086	Mutations	0	1.0936	Spots	3
-5.9771	Mutants	0	1.1334	Compounds	3
-4.6471	Domain	0	1.1391	Active	3
-3.9662	Mutant	0	1.1903	intermolecular	3
-2.9279	Mutation	0	1.2049	Interacting	3
-2.6234	Anti	0	1.2172	Bond	3
-2.5567	Cells	0	1.2236	Extensive	3
-2.1993	Sites	1	1.2634	Defined	3
-2.1408	Observed	1	1.3243	Ubiquitin	3
-2.0669	Position	1	1.3600	Docking	3
-1.9548	Gene	1	1.3602	Structure	3
-1.8865	Additional	1	1.3658	Form	3
-1.6241	Phosphorylation	1	1.3942	Expected	3
-1.4819	Effects	1	1.4608	Stable	3
-1.4340	Substitutions	1	1.5494	Pocket	3
-1.4193	Using	1	1.6241	Complexes	3
-1.4072	Effect	1	1.7257	Salt	3
-1.3690	Previously	1	1.7428	Loops	3
-1.3186	Reduced	1	1.7436	Terminal	3
-1.2951	Values	1	1.7469	Surface	3
-1.2812	Results	1	1.8078	Buried	3
-1.2804	Motif	1	1.8793	Aromatic	3
-1.2780	Substitution	1	2.0572	Sequence	3
-1.2552	Kinase	1	2.0645	Cluster	3
-1.1301	Levels	1	2.2051	Main	3
-1.1065	Reported	1	2.3774	Interface	3
-1.1025	Linker	1	2.4139	Interact	3
-1.0708	Catalytic	1	2.4456	Affinity	3
-1.0642	Amino	1	2.4764	Catenin	3
-1.0562	Specific	1	2.5009	Helix	4
-1.0432	Identified	1	2.5481	Site	4
-1.0416	Direct	1	2.5764	Chains	4
-1.0399	Chemical	1	2.6105	Contacts	4
-1.0107	Factor	1	2.7096	Patch	4
-1.0050	Described	1	2.7569	Interaction	4
-1.0049	Helices	1	2.7836	Complex	4
-1.0042	Core	1	3.2955	Hydrogen	4
1.0270	Burring	3	3.9495	Formed	4
1.0278	Contributing	3	4.1811	Chain	4
1.0603	Protonation	3	4.4273	Interactions	4
1.0855	Forms	3	5.2863	Binding	4
1.0900	Interacts	3	9.1639	Hydrophobic	4

Table C-2: *TM performance on full texts with residue filtering using NLP and two sets of keywords.*
*NLP algorithm includes SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automatically (**Table C-1** and manually [170] selected keywords.*

Keywords	L_{tot}[a]	L_{int}[b]	Coverage (%)[c]	Success (%)[d]	Accuracy (%)[e]
Auto	87	58	33.6	22.4	66.7
Manual	95	62	36.7	23.9	65.3

[a] *Number of complexes for which TM protocol 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}*

Table C-3: *TM performance on the abstracts of PMC-OA test set of full-text articles with simplified residue filtering (basic TM) and with the residue filtering by NLP and DL.*
*NLP algorithm includes SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automatically selected keywords (**Table C-1**). DL consists of Deep Recursive Neural Network model for classifying residues at the entire sentence level. Both SVM model and DRNN were trained on the reduced full-text training set.*

Method	L_{tot} [a]	L_{int} [b]	Coverage (%)[c]	Success (%)[d]	Accuracy (%)[e]	$\Delta N(0)$[f]	$\Delta N(1)$[f]
Basic TM	79	40	30.5	15.4	50.6	–	–
NLP	23	16	8.9	6.2	69.6	–34	–3
DL	20	14	7.7	5.4	70.0	–35	–5

[a] *Number of complexes for which TM 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 (259)*
[d] *Ratio of L_{int} and total number of complexes (259)*
[e] *Ratio of L_{int} and L_{tot}*
[f] *From Eq.(4-6) in the main text with values from basic TM (first row) as X_2*

Table C-4: *Influence of the training set on TM performance with residue filtering by DL. DL consists of Deep Recursive Neural Network model for classifying residues in the PubMed abstracts at the entire sentence level.*

Training dataset	L_{tot} [a]	L_{int} [b]	Coverage (%)[c]	Success (%)[d]	Accuracy (%)[e]	$\Delta N(0)$[f]	$\Delta N(1)$[f]
Full PMC-OA set, (4,982 sentences)	179	120	30.9	20.7	67.0	–3	+6
Reduced PMC-OA set (2,490 sentences)	116	77	20.0	13.3	66.4	–24	–7

[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 Eq.(4-6) in the main text with values from the second row in* **Table 4-1** *as X_2*

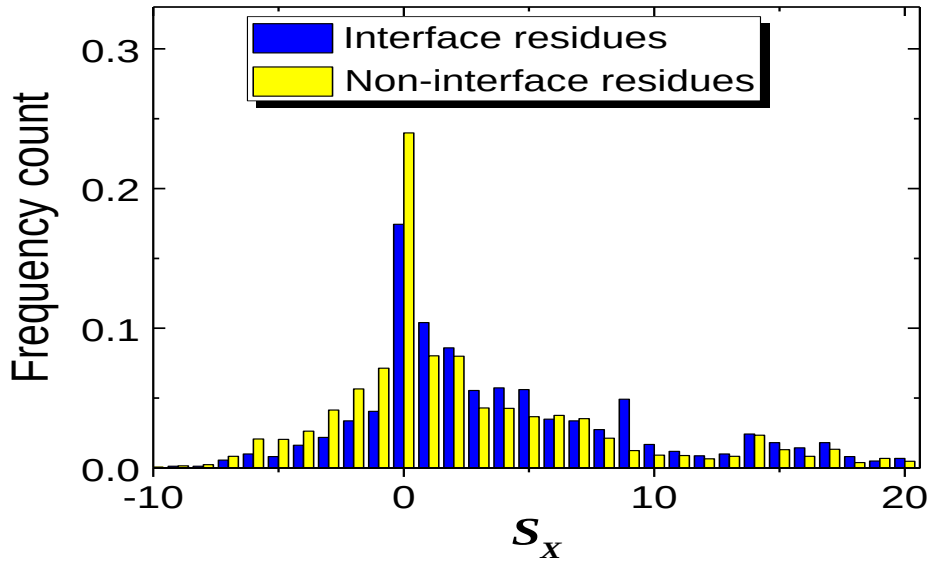


Figure C-1: *Distribution of S_X scores for the mined residues.*
The scores were calculated by Eq.(4-3) (main text) for the 1,605 interface and 3,377 non-interface mined residues from the parse trees of 4,982 sentences of 1,931 articles on studies of 313 complexes (from total of 579 complexes in the set).

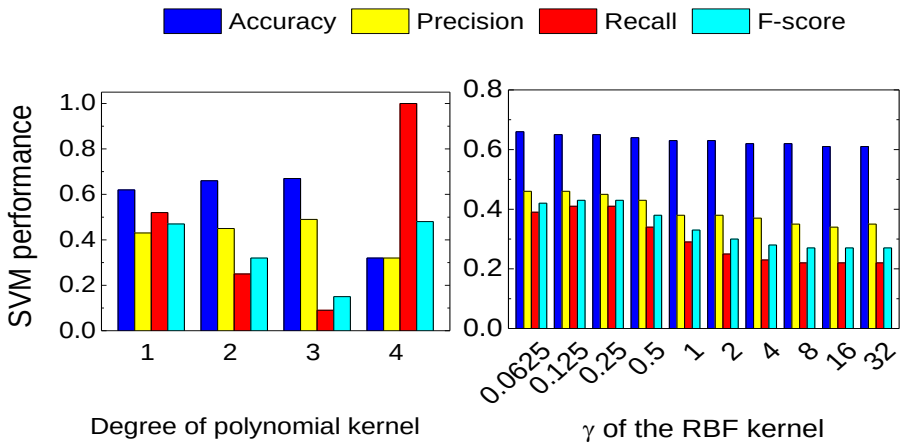


Figure C-2: *Optimization of SVM performance.*
Results for the SVM model with and without 0.05 margin (when sentences with the SVM scores –0.05 to +0.05 are excluded from the evaluation) are similar.

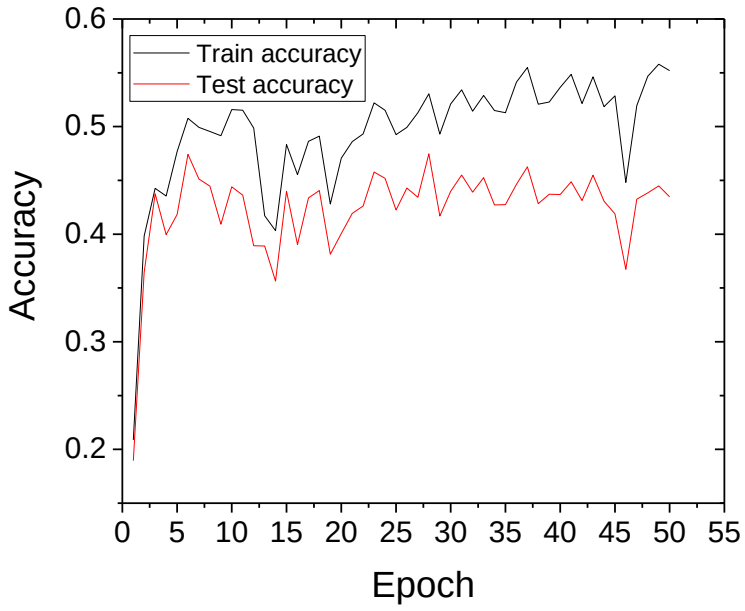


Figure C-3: *Dependence of Deep Recursive Neural Network model accuracy on the training length.*
Training and testing were performed on the 4,982 sentences of the PMC-OA full-text articles and 5,786 sentences of the PubMed abstracts, respectively. The accuracy was defined as the fraction of sentences in the dataset, for which the correct sentiment was assigned. The DRNN learned over ~10 epochs. Beyond that it appeared to be over-trained (accuracy on the training set increases without the corresponding increase in the accuracy on the test set). Sharp falls and rises in the DRNN accuracy can be attributed to the drop-out method used to avoid the exploding gradient issue in DL [165]. Drop-out nodes were chosen at random and at times could correspond to a weight representing important learning, thus causing a sudden drop in test and training accuracies learned in subsequent epochs.

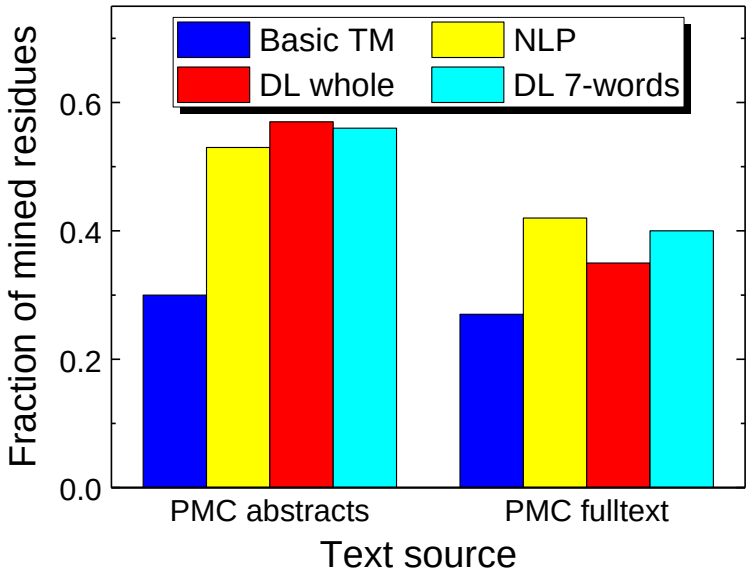


Figure C-4: *Fraction of interface residues among the mined residues obtained by different filtering algorithms.*
Data is obtained on the test set of PMC-OA full text articles (for 259 complexes). Basic TM performed simplified residue filtering on the information retrieved by the OR queries, and was the baseline for further residue filtering. NLP algorithm involved SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automated keywords. DL consisted of Deep Recursive Neural Network model for classifying residues using entire sentences and 7-words window around mined residues.

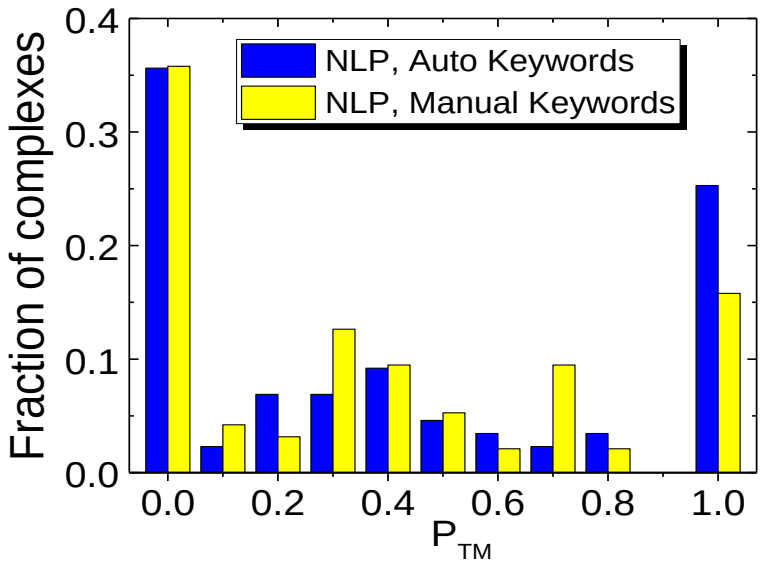


Figure C-5: *Comparison of NLP performance with automatic and manually selected keywords. The NLP algorithm includes SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automatically* ***Table C-1*** *and manually [170] selected keywords. The distribution is normalized by the total number of complexes for which residues were extracted (****Table C-2****).*

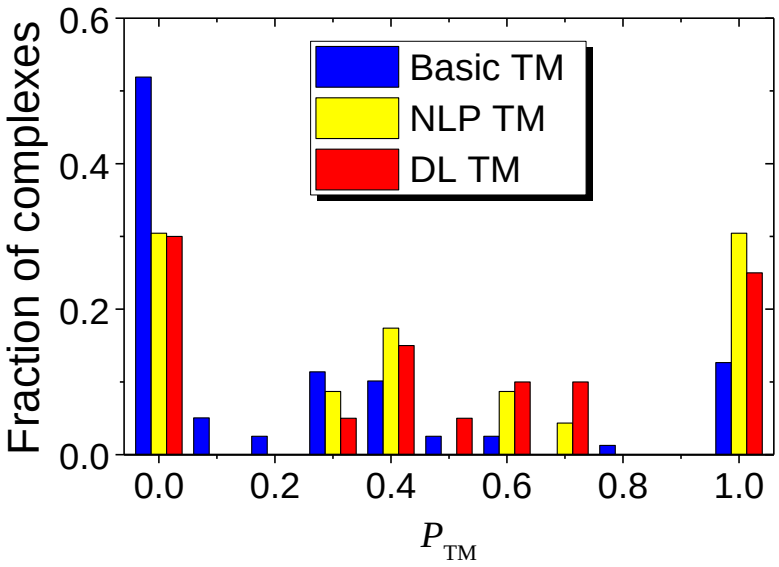


Figure C-6: *TM performance on the abstracts of PMC-OA test set of full-text articles with simplified residue filtering (basic TM) and with residue filtering by NLP and DL.*
*NLP consisted of the SVM model with scores from parse trees of residue-containing and surrounding sentences utilizing automatically selected keywords (**Table C-1**). DL consisted of Deep Recursive Neural Network model for classifying residues in the entire sentences. SVM model and DRNN were trained on the reduced full-text training set. The distribution is normalized to the total number of complexes for which residues were extracted (**Table C-3**).*

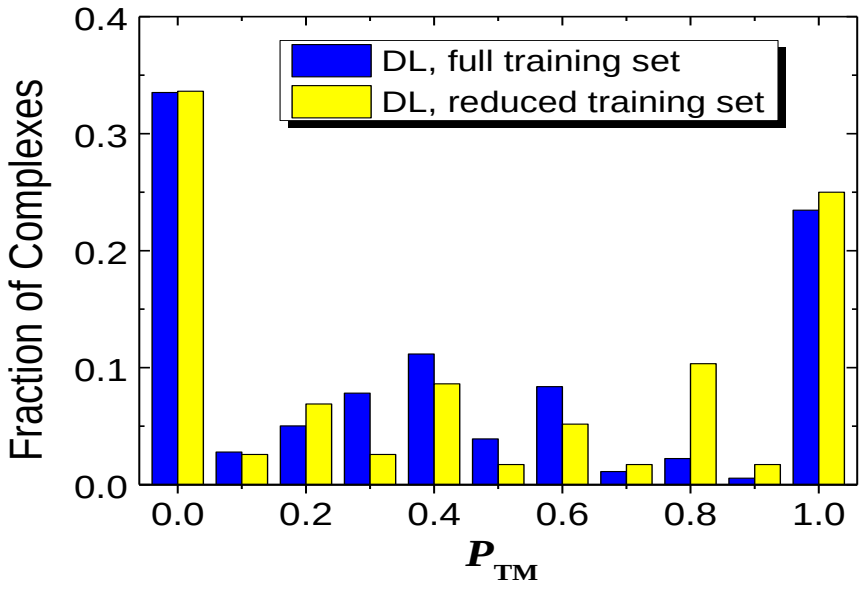


Figure C-7: *Influence of the training set on TM performance with residue filtering by DL. DL consisted of Deep Recursive Neural Network model for classifying residues in the PubMed abstracts at the entire sentence level. Full and reduced training sets consisted of 4,982 and 2,490 residue-containing sentences, respectively, from PMC-OA full-text articles. The TM performance was calculated by Eq.(4-5) in the main text. The distribution is normalized by the total number of complexes for which residues were identified (**Table C-4**).*

Appendix D

List of publications

Reprinted in this thesis

- **Badal VD,** Kundrotas PJ, Vakser IA: (2015) Text Mining for Protein Docking. *PLoS Comput Biol*, 11(12), e1004630.
- **Badal VD,** Kundrotas PJ, Vakser IA: Natural language processing in text mining for structural modeling of protein complexes. *BMC Bioinformatics* 2018, 19(1):84.
- **Badal VD,** Kundrotas PJ, Vakser IA, 2018. Enhanced text mining of biomedical literature for modeling of protein complexes. *Submitted.*

Other related work

- Anishchenko I, **Badal VD**, Dauzhenka T, Das M, Tuzikov AV, Kundrotas PJ, Vakser IA: Genome-Wide Structural Modeling of Protein-Protein Interactions. In: *Bioinformatics Research and Applications: 12th International Symposium, ISBRA 2016, Minsk, Belarus, June 5-8, 2016, Proceedings: 2016*. Springer, 9683: 95.

- Kundrotas PJ, Anishchenko I, **Badal VD**, Das M, Dauzhenka T, Vakser IA: Modeling CAPRI Targets 110–120 by Template-Based and Free Docking Using Contact Potential and Combined Scoring Function. *Proteins: Structure, Function, and Bioinformatics* 2017.

Bibliography

1. Vakser IA: **Low-resolution structural modeling of protein interactome**. *Curr Opin Struct Biol* 2013, **23**:198–205.
2. Vajda S, Hall DR, Kozakov D: **Sampling and scoring: A marriage made in heaven**. *Proteins* 2013, **81**:1874–1884.
3. Krallinger M, Erhardt RAA, Valencia A: **Text-mining approaches in molecular biology and biomedicine**. *Drug Discov Today* 2005, **10**:439-445.
4. Rebholz-Schuhmann D, Oellrich A, Hoehndorf R: **Text-mining solutions for biomedical research: Enabling integrative biology**. *Nature Rev Genetics* 2012, **13**:829-839.
5. Temkin JM, Gilder MR: **Extraction of protein interaction information from unstructured text using a context-free grammar**. *Bioinformatics* 2003, **19**:2046-2053.
6. Blaschke C, Andrade M, Ouzounis CA, Valencia A: **Automatic extraction of biological information from scientific text: protein-protein interactions**. In: *Proc ISMB-99 Conf: 1999; Heidelberg, Germany*. American Association for Artificial Intelligence: 60-67.
7. Kim S, Kwon D, Shin SY, Wilbur WJ: **PIE the search: Searching PubMed literature for protein interaction information**. *Bioinformatics* 2012, **28**:597-598.
8. Wong A, Shatkay H: **Protein function prediction using text-based features extracted from the biomedical literature: The CAFA challenge**. *BMC Bioinformatics* 2013, **14**:1.
9. Verspoor KM, Cohn JD, Ravikumar KE, Wall ME: **Text mining improves prediction of protein functional sites**. *PLoS One* 2012, **7**:e32171.
10. Lensink MF, Velankar S, Wodak SJ: **Modeling protein-protein and protein-peptide complexes: CAPRI 6th edition**. *Proteins* 2017, **85**:359-377.
11. Vakser IA: **Protein-protein docking: From interaction to interactome**. *Biophys J* 2014, **107**:1785-1793.
12. Glaser F, Steinberg D, Vakser IA, Ben-Tal N: **Residue frequencies and pairing preferences at protein-protein interfaces**. *Proteins* 2001, **43**:89-102.
13. Nicola G, Vakser IA: **A simple shape characteristic of protein-protein recognition**. *Bioinformatics* 2007, **23**:789-792.
14. Kundrotas PJ, Zhu Z, Janin J, Vakser IA: **Templates are available to model nearly all complexes of structurally characterized proteins**. *Proc Nat Acad Sci USA* 2012, **109**:9438-9441.
15. Anishchenko I, Kundrotas PJ, Vakser IA: **Structural quality of unrefined models in protein docking**. *Proteins* 2017, **85**(1):39-45.
16. Mandell JG, Roberts VA, Pique ME, Kotlovyi V, Mitchell JC, Nelson E, Tsigelny I, Ten Eyck LF: **Protein docking using continuum electrostatics and geometric fit**. *Protein Eng* 2001, **14**:105-113.
17. Vakser IA, Aflalo C: **Hydrophobic docking: A proposed enhancement to molecular recognition techniques**. *Proteins* 1994, **20**:320-329.
18. Vakser IA, Kundrotas P: **Predicting 3D structures of protein-protein complexes**. *Curr Pharm Biotech* 2008, **9**:57-66.
19. Kozakov D, Brenke R, Comeau SR, Vajda S: **PIPER: An FFT-based protein docking program with pairwise potentials**. *Proteins* 2006, **65**:392-406.
20. Mintseris J, Pierce B, Wiehe K, Anderson R, Chen R, Weng Z: **Integrating statistical pair potentials into protein complex prediction**. *Proteins* 2007, **69**:511-520.
21. Katchalski-Katzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA: **Molecular**

surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. *Proc NatL Acad Sci USA* 1992, **89**:2195-2199.
22. Fischer D, Norel R, Wolfson H, Nussinov R: **Surface motifs by a computer vision technique: Searches, detection, and implications for protein-ligand recognition**. *Proteins* 1993, **16**:278-292.
23. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D: **Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations**. *J Mol Biol* 2003, **331**:281-299.
24. Tovchigrechko A, Vakser IA: **GRAMM-X public web server for protein–protein docking**. *Nucl Acids Res* 2006, **34**:W310-W314.
25. Hwang H, Vreven T, Pierce BG, Hung JH, Weng Z: **Performance of ZDOCK and ZRANK in CAPRI rounds 13-19**. *Proteins* 2010, **78**:3104-3110.
26. Macindoe G, Mavridis L, Venkatraman V, Devignes M-D, Ritchie DW: **HexServer: An FFT-based protein docking server powered by graphics processors**. *Nucl Acids Res* 2010, **38**:W445-W449.
27. Ritchie DW, Kemp GJ: **Protein docking using spherical polar Fourier correlations**. *Proteins* 2000, **39**:178-194.
28. De Vries SJ, Van Dijk M, Bonvin AM: **The HADDOCK web server for data-driven biomolecular docking**. *Nature Prot* 2010, **5**:883.
29. Dominguez C, Boelens R, Bonvin AM: **HADDOCK: A protein− protein docking approach based on biochemical or biophysical information**. *J Amer Chem Soc* 2003, **125**:1731-1737.
30. Kundrotas PJ, Alexov E: **Predicting 3D structures of transient protein-protein complexes by homology**. *Biochim Biophys Acta* 2006, **1764**:1498-1511.
31. Kundrotas PJ, Vakser IA: **Global and local structural similarity in protein-protein complexes: Implications for template-based docking**. *Proteins* 2013, **81**:2137-2142.
32. Sinha R, Kundrotas PJ, Vakser IA: **Docking by structural similarity at protein-protein interfaces**. *Proteins* 2010, **78**:3235-3241.
33. Sinha R, Kundrotas PJ, Vakser IA: **Protein docking by the interface structure similarity: How much structure is needed?** *PLoS One* 2012, **7**:e31349.
34. Venkatraman V, Ritchie DW: **Flexible protein docking refinement using pose-dependent normal mode analysis**. *Proteins* 2012, **80**:2262-2274.
35. Moal IH, Bates PA: **SwarmDock and the use of normal modes in protein-protein docking**. *Int J Mol Sci* 2010, **11**:3623-3648.
36. Wang C, Bradley P, Baker D: **Protein-protein docking with backbone flexibility**. *J Mol Biol* 2007, **373**:503-519.
37. Badal VD, Kundrotas PJ, Vakser IA: **Text mining for protein docking**. *PLoS Comp Biol* 2015, **11**:e1004630.
38. Sanchez R, Sali A: **Advances in comparative protein-structure modeling**. *Curr Opin Struct Biol* 1997, **7**:206-214.
39. Aloy P, Ceulemans H, Stark A, Russell RB: **The relationship between sequence and interaction divergence in proteins**. *J Mol Biol* 2003, **332**:989-998.
40. Lu L, Lu H, Skolnick J: **MULTIPROSPECTOR: An algorithm for the prediction of protein-protein interactions by multimeric threading**. *Proteins* 2002, **49**:350-364.
41. Kundrotas PJ, Zhu Z, Janin J, Vakser IA: **Templates are available to model nearly all complexes of structurally characterized proteins**. *Proc Natl Acad Sci USA* 2012,

109:9438-9441.

42. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The Protein Data Bank**. *Nucleic Acids Res* 2000, **28**:235-242.
43. Dominguez C, Boelens R, Bonvin AMJJ: **HADDOCK: A protein-protein docking approach based on biochemical or biophysical information**. *J Am Chem Soc* 2003, **125**:1731-1737.
44. Moal IH, Moretti R, Baker D, Fernandez-Recio J: **Scoring functions for protein–protein interactions**. *Curr Opin Struct Biol* 2013, **23**:862–867.
45. Lensink MF, Wodak SJ: **Docking, scoring, and affinity prediction in CAPRI**. *Proteins* 2013, **81**:2082-2095.
46. 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.
47. Krallinger M, Valencia A: **Text-mining and information-retrieval services for molecular biology**. *Genome Biol* 2005, **6**:224.
48. Seoud AA, Solouma NH, Youssef AM, Kadah YM: **Extraction of protein interaction information from unstructured text using a link grammar parser**. In: *ICCES '07 International Conference on Computer Engineering & Systems 2007: 2007*. 70-75.
49. 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.
50. Niu Y, Otasek D, Jurisica I: **Evaluation of linguistic features useful in extraction of interactions from PubMed; Application to annotating known, high-throughput and predicted interactions in I2D**. *Bioinformatics* 2010, **26**:111-119.
51. 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**:867–875.
52. Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, Zhang S, Baskin B, Bader GD, Michalickova K *et al*: **PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine**. *BMC Bioinformatics* 2003, **4**:11.
53. 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-D400.
54. Czarnecki J, Nobeli I, Smith AM, Shepherd AJ: **A text-mining system for extracting metabolic reactions from full-text articles**. *BMC Bioinformatics* 2012, **13**:172.
55. Naderi N, Witte R: **Automated extraction and semantic analysis of mutation impacts from the biomedical literature**. *BMC Genom* 2012, **13**:S10.
56. Shatkay H, Brady S, Wong A: **Text as data: Using text-based features for proteins representation and for computational prediction of their characteristics**. *Methods* 2015, **74**:54-64.
57. Verspoor KM, Cohn JD, Ravikumar KE, Wal ME: **Text mining improves prediction of protein functional sites**. *PloS One* 2012, **7**:e32171.
58. Papanikolaou N, Pavlopoulos GA, Theodosiou T, Iliopoulos I: **Protein-protein interaction predictions using text mining methods**. *Methods* 2014, **74**:47-53.
59. Kim JD, Ohta T, Tateisi Y, Tsujii J: **GENIA corpus—a semantically annotated corpus for bio-textmining**. *Bioinformatics* 2003, **19**:i180-i182.

60. Barbosa-Silva A, Fontaine JF, Donnard ER, Stussi F, Ortega JM, Andrade-Navarro MA: **PESCADOR, a web-based tool to assist textmining of biointeractions extracted from PubMed queries**. *BMC Bioinformatics* 2011, **12**:435.
61. Barbosa-Silva A, Soldatos TG, Magalhaes IL, Pavlopoulos GA, Fontaine JF, Andrade-Navarro MA, Schneider R, Ortega JM: **LAITOR--Literature Assistant for Identification of Terms co-Occurrences and Relationships**. *BMC Bioinformatics* 2010, **11**:70.
62. Korhonen A, Seaghdha DO, Silins I, Sun L, Hogberg J, Stenius U: **Text mining for literature review and knowledge discovery in cancer risk assessment and research**. *PloS One* 2012, **7**:e33427.
63. Kim S, Wilbur WJ: **Classifying protein-protein interaction articles using word and syntactic features**. *BMC Bioinformatics* 2011, **12(Suppl 8)**:S9.
64. Tudor CO, Arighi CN, Wang Q, Wu CH, Vijay-Shanker K: **The eFIP system for text mining of protein interaction networks of phosphorylated proteins**. *Database* 2012:bas044.
65. Raja K, Subramani S, Natarajan J: **PPInterFinder—a mining tool for extracting causal relations on human proteins from literature**. *Database* 2013:bas052.
66. Kwon D, Kim S, Shin SY, Chatr-Aryamontri A, Wilbur WJ: **Assisting manual literature curation for protein-protein interactions using BioQRator**. *Database* 2014:bau067.
67. Consortium U: **Activities at the Universal Protein Resource (UniProt)**. *Nucl Acid Res* 2014, **42**:D191-D198.
68. Knecht LWS, Nelson SJ: **Mapping in PubMed**. *J Med Lib Assoc* 2002, **90**:475-476.
69. Sayers E: **A General Introduction to the E-utilities**. In: *Entrez Programming Utilities Help [Internet]*. 2010.
70. Rose GD, Geselowitz AR, Lesser GJ, Lee RH, Zehfus MH: **Hydrophobicity of amino acid residues in globular proteins**. *Science* 1985, **229**:834-838.
71. Nicholls A, Sharp KA, B. H: **Protein folding and association: Insights from the interfacial and thermodynamic properties of hydrocarbons**. *Proteins* 1991, **11**:281-296.
72. Martin ACR: **Mapping PDB chains to UniProtKB entries**. *Bioinformatics* 2005, **21**:4297-4301.
73. Jiang J, Zhai C: **An empirical study of tokenization strategies for biomedical information retrieval**. *Inform Retrieval* 2007, **10**:341-363.
74. Porter MF: **An algorithm for suffix stripping**. *Program-Electon Lib* 1980, **14**:130-137.
75. Joachims T: **Making large-scale support vector machine learning practical**. In: *Advances in Kernel Methods.* MIT Press; 1999: 169-184.
76. Morik K, Brockhausen P, Joachims T: **Combining statistical learning with a knowledge-based approach: A case study in intensive care monitoring**. Universität Dortmund; 1999.
77. Shatkay H, Feldman R: **Mining the biomedical literature in the genomic era: An overview**. *J Comp Biol* 2003, **10**:821–855.
78. Byvatov E, Fechner U, Sadowski J, Schneider G: **Comparison of support vector machine and artificial neural network systems for drug/nondrug classification**. *J Chem Inf Comput Sci* 2003, **43**:1882-1889.
79. Gao Y, Douguet D, Tovchigrechko A, Vakser IA: **DOCKGROUND system of databases for protein recognition studies: Unbound structures for docking**. *Proteins* 2007, **69**:845-851.

80. Vakser IA: **Protein docking for low-resolution structures**. *Protein Eng* 1995, **8**:371-377.
81. Joachims T: **Text categorization with Support Vector Machines: Learning with many relevant features**. In: *Machine Learning: ECML-98.* Edited by Nedellec C, Rouveirol C. Berlin Heidelberg: Springer; 1998: 137-142.
82. Ozgur A, Ozgur L, Gungor T: **Text categorization with class-based and corpus-based keyword selection**. In: *Computer and Information Sciences-ISCIS 2005.* Springer; 2005: 606-615.
83. Jamal N, Mohd M, Noah SA: **Poetry classification using support vector machines**. 2012, **8**:1441.
84. Wong A, Shatkay H: **Protein function prediction using text-based features extracted from the biomedical literature: The CAFA challenge**. *BMC Bioinformatics* 2013, **14(Suppl 3)**:S14.
85. Koyama Y, Baba A, Matsuda T: **Intracerebroventricular administration of an endothelin ETB receptor agonist increases expression of tissue inhibitor of matrix metalloproteinase-1 and -3 in rat brain**. *Neuroscience* 2007, **147**:620-630.
86. Liu S, Gao Y, Vakser IA: **DOCKGROUND protein-protein docking decoy set**. *Bioinformatics* 2008, **24**:2634-2635.
87. Moal IH, Moretti R, Baker D, Fernandez-Recio J: **Scoring functions for protein–protein interactions**. *Curr Opin Struc Biol* 2013, **23**:862-867.
88. de Vries SJ, van Dijk ADJ, Bonvin AMJJ: **WHISCY: What information does surface conservation yield? Application to data-driven docking**. *Proteins* 2006, **63**:479-489.
89. 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**(suppl 1):S74-S82.
90. 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.
91. Fundel K, Kuffner R, Zimmer R: **RelEx—Relation extraction using dependency parse trees**. *Bioinformatics* 2007, **23**:365-371.
92. Califf ME, Mooney RJ: **Relational learning of pattern-match rules for information extraction**. In: *Proc 16th Natl Conf Artificial Intelligence: 1999; Orlando, Florida*. The AAAI Press, Menlo Park, California: 328.
93. 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: 408-419.
94. 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: 2012; Arlington, Virginia*. 32-37.
95. Liu H, Hunter L, Keselj V, Verspoor K: **Approximate subgraph matching-based literature mining for biomedical events and relations**. *PloS One* 2013, **8**:e60954.
96. 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: 2015; Beijing, China*. 21-30.
97. 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: 724-731.
98. Mooney RJ, Bunescu RC: **Subsequence kernels for relation extraction**. In: *Proc 2005 Conf (NIPS): 2005; Vancouver, Canada*. MIT Press: 171-178.

99. Moschitti A: **Making tree kernels practical for natural language learning**. In: *Proc 11th Conf Eur Ch Associ Comput Linguistics: 2006; Trento, Italy*. 113-120.
100. Moschitti A: **A study on convolution kernels for shallow semantic parsing**. In: *Proc 42nd Ann Meeting Assoc Comput Linguistics: 2004; Barcelona, Spain*. Association for Computational Linguistics: 335-342.
101. Culotta A, Sorensen J: **Dependency tree kernels for relation extraction**. In: *Proc 42nd Annual Meeting Association for Comput Linguistics: 2004; Barcelona, Spain*. Association for Computational Linguistics: 423-429.
102. Quan C, Wang M, Ren F: **An unsupervised text mining method for relation extraction from biomedical literature**. *PloS One* 2014, **9**:e102039.
103. Blaschke C, Valencia A: **The frame-based module of the SUISEKI information extraction system**. *IEEE Intell Syst* 2002:14-20.
104. Raja K, Subramani S, Natarajan J: **PPInterFinder—a mining tool for extracting causal relations on human proteins from literature**. *Database* 2013, **2013**:bas052.
105. 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.
106. He M, Wang Y, Li W: **PPI finder: A mining tool for human protein-protein interactions**. *PloS One* 2009, **4**:e4554.
107. 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.
108. 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 2015; Sevilla, Spain*. 30-35.
109. Koyabu S, Phan TT, Ohkawa T: **Extraction of protein-protein interaction from scientific articles by predicting dominant keywords**. *BioMed Res Int* 2015, **2015**:928531.
110. 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: 2007; Prague, Czech Republic*. Association for Computational Linguistics: 228-237.
111. 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.
112. Zhou D, He Y: **Extracting interactions between proteins from the literature**. *J Biomed Inform* 2008, **41**:393-407.
113. Banerjee S, Pedersen T: **An adapted Lesk algorithm for word sense disambiguation using WordNet**. In: *Proc 3rd Int Conf CompLinguistics Intelligent Text Processing: 2002; Mexico City, Mexico*. Springer-Verlag London: 136-145.
114. 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.
115. Lin D: **An information-theoretic definition of similarity**. In: *Proc 15th Int Conf Machine Learning: 1998; Madison, Wisconsin, USA*. Morgan Kaufmann Publishers Inc. San

Francisco, CA, USA: 296-304.
116. Meng L, Huang R, Gu J: **A review of semantic similarity measures in wordnet**. *Int JHybrid Inf Technol* 2013, **6**:1-12.
117. 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.
118. Miller GA: **WordNet: A lexical database for English**. *Commun ACM* 1995, **38**:39-41.
119. Fellbaum C: **WordNet: An electronic lexical database**: MIT Press, Cambridge MA; 1998.
120. De Marneffe MC, Manning CD: **Stanford typed dependencies manual**. In.: Technical report, Stanford University; 2008: 338-345.
121. De Marneffe MC, Manning CD: **The Stanford typed dependencies representation**. In: *Proc Workshop Cross-Framework Cross-Domain Parser Evaluation: 2008; Manchester, UK*. Association for Computational Linguistics: 1-8.
122. Joachims T: **Text categorization with Support Vector Machines: Learning with many relevant features**. In: *Machine Learning: ECML-98.* Edited by Nedellec C, Rouveirol C, vol. 1398: Springer Berlin Heidelberg; 1998: 137-142.
123. 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.
124. Shatkay H, Feldman R: **Mining the biomedical literature in the genomic era: An overview**. *J Comput Biol* 2003, **10**:821–855.
125. Vakser IA: **Low-resolution docking: Prediction of complexes for underdetermined structures**. *Biopolymers* 1996, **39**:455-464.
126. Zervanou K, McNaught J: **A term-based methodology for template creation in information extraction**. In: *Proc 2nd Int Conf Natural Language Processing: 2000; Patras, Greece*. Springer: 418-423.
127. Pucher M: **Performance evaluation of WordNet-based semantic relatedness measures for word prediction in conversational speech**. In: *Proc 6th Int Workshop Comput Semantics: 2005; Tilburg, Netherlands*.
128. 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**:141-155.
129. Knecht LWS, Nelson SJ: **Mapping in PubMed**. *J Med Libr Assoc* 2002, **90**:475-476.
130. 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**:200-207.
131. Chowdhary R, Zhang J, Liu JS: **Bayesian inference of protein–protein interactions from biological literature**. *Bioinformatics* 2009, **25**:1536-1542.
132. 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**:481-494.
133. Plake C, Hakenberg J, Leser U: **Optimizing syntax patterns for discovering protein-protein interactions**. In: *Proc 2005 ACM Symp Applied Computing: 2005; Santa Fe, New Mexico* ACM: 195-201.
134. Martin EP, Bremer EG, Guerin M-C, DeSesa C, Jouve O: **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; 2004: 96-108.

135. Schuemie MJ, Weeber M, Schijvenaars BJ, van Mulligen EM, van der Eijk CC, Jelier R, Mons B, Kors JA: **Distribution of information in biomedical abstracts and full-text publications**. *Bioinformatics* 2004, **20**:2597-2604.
136. Lan M, Su J: **Empirical investigations into full-text protein interaction Article Categorization Task (ACT) in the BioCreative II. 5 Challenge**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 2010, **7**:421-427.
137. Cohen KB, Johnson HL, Verspoor K, Roeder C, Hunter LE: **The structural and content aspects of abstracts versus bodies of full text journal articles are different**. *BMC Bbioinformatics* 2010, **11**:492.
138. Shah PK, Perez-Iratxeta C, Bork P, Andrade MA: **Information extraction from full text scientific articles: Where are the keywords?** *BMC Bioinformatics* 2003, **4**:20.
139. Caporaso JG, Deshpande N, Fink JL, Bourne PE, Cohen KB, Hunter L: **Intrinsic evaluation of text mining tools may not predict performance on realistic tasks**. In: *Pacific Sympos Biocomputing: 2008*. NIH Public Access: 640.
140. Cohen AM, Hersh WR: **A survey of current work in biomedical text mining**. *Brief Bioinf* 2005, **6**:57-71.
141. Rodriguez-Esteban R: **Biomedical text mining and its applications**. *PLoS Comput Biol* 2009, **5**:e1000597.
142. Lin J: **Is searching full text more effective than searching abstracts?** *BMC Bioinformatics* 2009, **10**:46.
143. Corney DP, Buxton BF, Langdon WB, Jones DT: **BioRAT: Extracting biological information from full-length papers**. *Bioinformatics* 2004, **20**:3206-3213.
144. McIntosh T, Curran JR: **Challenges for automatically extracting molecular interactions from full-text articles**. *BMC Bioinformatics* 2009, **10**:311.
145. Mallory EK, Zhang C, Re C, Altman RB: **Large-scale extraction of gene interactions from full text literature using deepdive**. *Bioinformatics* 2015:btv476.
146. Fink JL, Kushch S, Williams PR, Bourne PE: **BioLit: Integrating biological literature with databases**. *Nucl Acids Res* 2008, **36**(suppl 2):W385-W389.
147. Gerner M, Nenadic G, Bergman CM: **An exploration of mining gene expression mentions and their anatomical locations from biomedical text**. In: *Proc 2010 Workshop Biomed Natural Language Processing: 2010*. Association for Computational Linguistics: 72-80.
148. Gerner M, Sarafraz F, Bergman CM, Nenadic G: **BioContext: An integrated text mining system for large-scale extraction and contextualization of biomolecular events**. *Bioinformatics* 2012, **28**:2154-2161.
149. Huang M, Zhu X, Hao Y, Payan DG, Qu K, Li M: **Discovering patterns to extract protein-protein interactions from full texts**. *Bioinformatics* 2004, **20**:3604-3612.
150. Peng Y, Arighi C, Wu CH, Vijay-Shanker K: **BioC-compatible full-text passage detection for protein–protein interactions using extended dependency graph**. *Database* 2016, **2016**:baw072.
151. Krallinger M, Leitner F, Rodriguez-Penagos C, Valencia A: **Overview of the protein-protein interaction annotation extraction task of BioCreative II**. *Genome Biol* 2008, **9**(Suppl 2):S4.

152. Dogan RI, Kim S, Chatr-Aryamontri A, Chang CS, Oughtred R, Rust J, Wilbur WJ, Comeau DC, Dolinski K, Tyers M: **The BioC-BioGRID corpus: Full text articles annotated for curation of protein–protein and genetic interactions**. *Database* 2017, **2017**.
153. LeCun Y, Bengio Y, Hinton G: **Deep learning**. *Nature* 2015, **521**:436-444.
154. Bengio Y, Courville A, Vincent P: **Representation learning: A review and new perspectives**. *IEEE transactions on pattern analysis and machine intelligence* 2013, **35**:1798-1828.
155. Rumelhart D, Hinton G, Williams R: **Learning representations by back-propagating errors**. *Nature* 1986, **323**:533-538.
156. Schwenk H: **Continuous space language models**. *Comp Speech & Language* 2007, **21**:492-518.
157. Mikolov T: **Statistical language models based on neural networks**. *PhD Thesis, Brno University of Technology* 2012.
158. Turney PD: **Distributional semantics beyond words: Supervised learning of analogy and paraphrase**. *Trans Assoc Comput Linguistics (TACL)* 2013, **1**:353-366.
159. Weston J, Bengio S, Usunier N: **Wsabie: Scaling up to large vocabulary image annotation**. In: *IJCAI: 2011*. 2764-2770.
160. Socher R, Lin CC, Manning C, Ng AY: **Parsing natural scenes and natural language with recursive neural networks**. In: *Proc 28th Int Conf Machine Learning (ICML-11): 2011*. 129-136.
161. Mikolov T, Chen K, Corrado G, Dean J: **Efficient estimation of word representations in vector space**. *arXiv:13013781* 2013.
162. Mikolov T, Yih W-t, Zweig G: **Linguistic Regularities in Continuous Space Word Representations**. In: *Hlt-naacl: 2013*. 746-751.
163. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J: **Distributed representations of words and phrases and their compositionality**. In: *Adv Neural Information Processing Systems: 2013*. 3111-3119.
164. Collobert R, Weston J: **A unified architecture for natural language processing: Deep neural networks with multitask learning**. In: *Proc 25th Int Conf Machine Learning: 2008*. ACM: 160-167.
165. Irsoy O, Cardie C: **Deep recursive neural networks for compositionality in language**. In: *Adv Neural Information Processing Systems: 2014*. 2096-2104.
166. Irsoy O, Cardie C: **Modeling compositionality with multiplicative recurrent neural networks**. *arXiv:14126577* 2014.
167. Brants T, Popat AC, Xu P, Och FJ, Dean J: **Large language models in machine translation**. In: *Proc Joint Conf Empirical Methods in Natural Language Processing and Computational Natural Language Learning: 2007*. Citeseer.
168. Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C: **Recursive deep models for semantic compositionality over a sentiment treebank**. In: *Proc Conf Empirical Methods in Natural Language Processing (EMNLP): 2013*. Citeseer: 1642.
169. Socher R, Pennington J, Huang EH, Ng AY, Manning CD: **Semi-supervised recursive autoencoders for predicting sentiment distributions**. In: *Proc Conf Empirical Methods in Natural Language Processing: 2011*. Association for Computational Linguistics: 151-161.
170. Badal VD, Kundrotas PJ, Vakser IA: **Natural language processing in text mining for**

structural modeling of protein complexes. *BMC Bioinformatics* 2018, **19**:84.
171. NCBI RC: **Database resources of the National Center for Biotechnology Information**. *Nucl Acids Res* 2013, **41**:D8.
172. Kundrotas PJ, Anishchenko I, Dauzhenka T, Kotthoff I, Mnevets D, Copeland MM, Vakser IA: **Dockground: A comprehensive data resource for modeling of protein complexes**. *Protein Sci* 2018, **27**:172-181.
173. Pennington J, Socher R, Manning C: **Glove: Global vectors for word representation**. In: *Proc 2014 Conf Empirical Methods in Natural Language Processing (EMNLP): 2014*. 1532-1543.
174. Jurafsky D, Martin JH: **Semantics with Dense Vectors** In: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 2017.
175. van der Maaten L, Hinton G: **Visualizing data using t-SNE**. *J Machine Learning Res* 2008, **9**:2579-2605.
176. De Marneffe M-C, Manning CD: **Stanford typed dependencies manual**. In.: Technical report, Stanford University; 2008: 338-345.
177. Nunez-Cruz S, Yeo WC, Rothman J, Ojha P, Bassiri H, al. e: **Differential requirement for the SAP-Fyn interaction during NK T cell development and function**. *J Immunol* 2008, **181**:2311-2320.
178. McCausland MM, Yusuf I, Tran H, Ono N, Yanagi Y, al. e: **SAP regulation of follicular helper CD4 T cell development and humoral immunity is independent of SLAM and Fyn kinase**. *J Immunol* 2007, **178**:817-828.
179. Cannons JL, Yu LJ, Jankovic D, Crotty S, Horai R, al. e: **SAP regulates T cell-mediated help for humoral immunity by a mechanism distinct from cytokine regulation**. *J Exp Med* 2006, **203**:1551-1565.
180. Panchamoorthy G, Fukazawa T, Stolz L, Payne G, Reedquist K, al. e: **Physical and functional interactions between SH2 and SH3 domains of the Src family protein tyrosine kinase p59fyn**. *Mol Cell Biol* 1994, **14**:6372-6385.
181. Koyasu S, McConkey DJ, Clayton LK, Abraham S, Yandava B, al. e: **Phosphorylation of multiple CD3 zeta tyrosine residues leads to formation of pp21 in vitro and in vivo. Structural changes upon T cell receptor stimulation**. *J Biol Chem* 1992, **267**:3375-3381.
182. Chandrasekaran P, Rajasekaran R: **Structural characterization of disease-causing mutations on SAP and the functional impact on the SLAM peptide: A molecular dynamics approach**. *Mol Biosyst* 2014, **10**:1869-1880.
183. Lee MH, Maskos K, Knauper V, Dodds P, Murphy G: **Mapping and characterization of the functional epitopes of tissue inhibitor of metalloproteinases (TIMP)-3 using TIMP-1 as the scaffold: A new frontier in TIMP engineering**. *Protein Sci* 2002, **11**(10):2493-2503.
184. Lee MH, Verma V, Maskos K, Nath D, Knauper V, al. e: **Engineering N-terminal domain of tissue inhibitor of metalloproteinase (TIMP)-3 to be a better inhibitor against tumour necrosis factor-α-converting enzyme**. *Biochem J* 2002, **364**:227-234.
185. Lee MH, Rapti M, Murphy G: **Delineating the molecular basis of the inactivity of tissue inhibitor of metalloproteinase-2 against tumor necrosis factor-α-converting enzyme**. *J Biol Chem* 2004, **279**:45121-45129.
186. Lee MH, Dodds P, Verma V, Maskos K, Knauper V, al. e: **Tailoring tissue inhibitor of metalloproteinases-3 to overcome the weakening effects of the cysteine-rich domains**

of tumour necrosis factor-alpha converting enzyme. *Biochem J* 2003, **371**:369-376.
187. Karan D, Lin FC, Bryan M, Ringel J, Moniaux N, al. e: **Expression of ADAMs (a disintegrin and metalloproteases) and TIMP-3 (tissue inhibitor of metalloproteinase-3) in human prostatic adenocarcinomas**. *Int J Oncol* 2003, **23**:1365-1371.
188. Lee MH, Rapti M, Knauper V, Murphy G: **Threonine 98, the pivotal residue of tissue inhibitor of metalloproteinases (TIMP)-1 in metalloproteinase recognition**. *J Biol Chem* 2004, **279**:17562-17569.
189. Wisniewska M, Goettig P, Maskos K, Belouski E, Winters D, al. e: **Structural determinants of the ADAM inhibition by TIMP-3: crystal structure of the TACE-N-TIMP-3 complex**. *J Mol Biol* 2008, **381**:1307-1319.
190. Ingram RN, Orth P, Strickland CL, Le HV, Madison V, al. e: **Stabilization of the autoproteolysis of TNF-alpha converting enzyme (TACE) results in a novel crystal form suitable for structure-based drug design studies**. *Protein Eng* 2006, **19**:155-161.
191. Esposito C, Liu ZH, Striker GE, Phillips C, Chen NY, al. e: **Inhibition of diabetic nephropathy by a GH antagonist: a molecular analysis**. *Kidney Int* 1996, **50**:506-514.
192. Lee MH, Atkinson S, Murphy G: **Identification of the extracellular matrix (ECM) binding motifs of tissue inhibitor of metalloproteinases (TIMP)-3 and effective transfer to TIMP-1**. *J Biol Chem* 2007, **282**:6887-6898.
193. Xu C, Hou Z, Zhan P, Zhao W, Chang C, al. e: **EZH2 regulates cancer cell migration through repressing TIMP-3 in non-small cell lung cancer**. *Med Oncol* 2013, **30**:1-8.
194. Lim N, Kashiwagi M, Visse R, Jones J, Enghild J, al. e: **Reactive-site mutants of N-TIMP-3 that selectively inhibit ADAMTS-4 and ADAMTS-5: Biological and structural implications**. *Biochem J* 2010, **431**:113-122.
195. Koster J, Kuikman I, Kreft M, Sonnenberg A: **Two different mutations in the cytoplasmic domain of the integrin beta 4 subunit in nonlethal forms of epidermolysis bullosa prevent interaction of beta 4 with plectin**. *The Journal of investigative dermatology* 2001, **117**(6):1405-1411.
196. Nievers MG, Kuikman I, Geerts D, Leigh IM, Sonnenberg A: **Formation of hemidesmosome-like structures in the absence of ligand binding by the (alpha)6(beta)4 integrin requires binding of HD1/plectin to the cytoplasmic domain of the (beta)4 integrin subunit**. *Journal of cell science* 2000, **113 (Pt 6)**:963-973.
197. Litjens SH, Wilhelmsen K, de Pereda JM, Perrakis A, Sonnenberg A: **Modeling and experimental validation of the binary complex of the plectin actin-binding domain and the first pair of fibronectin type III (FNIII) domains of the beta4 integrin**. *The Journal of biological chemistry* 2005, **280**(23):22270-22277.
198. Tian R, Gregor M, Wiche G, Goldman JE: **Plectin regulates the organization of glial fibrillary acidic protein in Alexander disease**. *The American journal of pathology* 2006, **168**(3):888-897.
199. Pulkkinen L, Rouan F, Bruckner-Tuderman L, Wallerstein R, Garzon M, Brown T, Smith L, Carter W, Uitto J: **Novel ITGB4 mutations in lethal and nonlethal variants of epidermolysis bullosa with pyloric atresia: missense versus nonsense**. *American journal of human genetics* 1998, **63**(5):1376-1387.
200. Kambham N, Tanji N, Seigle RL, Markowitz GS, Pulkkinen L, Uitto J, D'Agati VD: **Congenital focal segmental glomerulosclerosis associated with beta4 integrin mutation and epidermolysis bullosa**. *American journal of kidney diseases : the official journal of the National Kidney Foundation* 2000, **36**(1):190-196.

201. Wang JT, Doong SL, Teng SC, Lee CP, Tsai CH, al. e: **Epstein-Barr virus BGLF4 kinase suppresses the interferon regulatory factor 3 signaling pathway**. *J Virol* 2009, **83**:1856-1869.
202. Clement JF, Bibeau-Poirier A, Gravel SP, Grandvaux N, Bonneil E, al. e: **Phosphorylation of IRF-3 on Ser 339 generates a hyperactive form of IRF-3 through regulation of dimerization and CBP association**. *J Virol* 2008, **82**:3984-3996.
203. Chen W, Srinath H, Lam SS, Schiffer CA, Royer WE, al. e: **Contribution of Ser386 and Ser396 to activation of interferon regulatory factor 3**. *J Mol Biol* 2008, **379**:251-260.
204. Panne D, McWhirter SM, Maniatis T, Harrison SC: **Interferon regulatory factor 3 is regulated by a dual phosphorylation-dependent switch**. *J Biol Chem* 2007, **282**:22816-22822.
205. Mori M, Yoneyama M, Ito T, Takahashi K, Inagaki F, al. e: **Identification of Ser-386 of interferon regulatory factor 3 as critical target for inducible phosphorylation that determines activation**. *J Biol Chem* 2004, **279**:9698-9702.
206. Takahasi K, Horiuchi M, Fujii K, Nakamura S, Noda NN, al. e: **Ser386 phosphorylation of transcription factor IRF-3 induces dimerization and association with CBP/p300 without overall conformational change**. *Genes Cells* 2010, **15**:901-910.
207. Chatr-Aryamontri A, Breitkreutz BJ, Oughtred R, Boucher L, Heinicke S, al. e: **The BioGRID interaction database: 2015 update**. *Nucl Acid Res* 2015, **43**:D470-478.
208. Saitoh T, Tun-Kyi A, Ryo A, Yamamoto M, Finn G, al. e: **Negative regulation of interferon-regulatory factor 3-dependent innate antiviral response by the prolyl isomerase Pin1**. *Nat Immunol* 2006, **7**:598-605.
209. Gu M, Zhang T, lin W, Liu Z, Lai R, al. e: **Protein phosphatase PP1 negatively regulates the Toll-like receptor- and RIG-I-like receptor-triggered production of type I interferon by inhibiting IRF3 phosphorylation at serines 396 and 385 in macrophage**. *Cell Signal* 2014, **26**:2930-2939.
210. Bergstroem B, Johnsen IB, Nguyen TT, Hagen L, Slupphaug G, al. e: **Identification of a novel in vivo virus-targeted phosphorylation site in interferon regulatory factor-3 (IRF3)**. *J Biol Chem* 2010, **285**:24904-24914.
211. Anglero-Rodriguez YI, Pantoja P, Sariol CA: **Dengue virus subverts the interferon induction pathway via NS2B/3 protease-IkappaB kinase epsilon interaction**. *Clin Vaccine Immunol* 2014, **21**:29-38.
212. Shu C, Sankaran B, Chaton CT, Herr AB, Mishra A, al. e: **Structural insights into the functions of TBK1 in innate antimicrobial immunity**. *Structure* 2013, **21**:1137-1148.
213. Inoue K, Tsukiyama-Kohara K, Matsuda C, Yoneyama M, Fujita T, al. e: **Impairment of interferon regulatory factor-3 activation by hepatitis C virus core protein basic amino acid region 1**. *Bioch Biophys Res Comm* 2012, **428**:494-499.
214. Fujii K, Nakamura S, Takahashi K, Inagaki F: **Systematic characterization by mass spectrometric analysis of phosphorylation sites in IRF-3 regulatory domain activated by IKK-i**. *J Proteomics* 2010, **73**:1196-1203.
215. Correia S, Ventura S, Parkhouse RM: **Identification and utility of innate immune system evasion mechanisms of ASFV**. *Virus Res* 2013, **173**:87-100.
216. Bayer P, Arndt A, Metzger S, Mahajan R, Melchior F, Jaenicke R, Becker J: **Structure determination of the small ubiquitin-related modifier SUMO-1**. *Journal of molecular biology* 1998, **280**(2):275-286.
217. Pilla E, Moller U, Sauer G, Mattiroli F, Melchior F, Geiss-Friedlander R: **A novel SUMO1-**

specific interacting motif in dipeptidyl peptidase 9 (DPP9) that is important for enzymatic regulation. *J Biol Chem* 2012, **287**(53):44320-44329.
218. Wang Y, Irqeba AA, Ayalew M, Suntay K: **Sumoylation of transcription factor Tec1 regulates signaling of mitogen-activated protein kinase pathways in yeast**. *PloS one* 2009, **4**(10):e7456.
219. Bielska K, Seliga J, Wieczorek E, Kedracka-Krok S, Niedenthal R, Ozyhar A: **Alternative sumoylation sites in the Drosophila nuclear receptor Usp**. *J Steroid Biochem Mol Biol* 2012, **132**(3-5):227-238.
220. Stehmeier P, Muller S: **Phospho-regulated SUMO interaction modules connect the SUMO system to CK2 signaling**. *Mol Cell* 2009, **33**(3):400-409.
221. Yasui N, Nogi T, Takagi J: **Structural basis for specific recognition of reelin by its receptors**. *Structure* 2010, **18**:320-331.
222. Yasui N, Nogi T, Kitao T, Nakano Y, Hattori M, Takagi J: **Structure of a receptor-binding fragment of reelin and mutational analysis reveal a recognition mechanism similar to endocytic receptors**. *Proc Natl Acad Sci USA* 2007, **104**:9988-9993.
223. Yasui N, Kitago Y, Beppu A, Kohno T, Morishita S, Gomi H, Nagae M, Hattori M, Takagi J: **Functional importance of covalent homodimer of reelin protein linked via its central region**. *J Biol Chem* 2011, **286**:35247-35256.
224. Wang X, Lu XA, Song X, Zhuo W, Jia L, Jiang Y, Luo Y: **Thr90 phosphorylation of Hsp90alpha by protein kinase A regulates its chaperone machinery**. *Biochem J* 2012, **441**:387-397.
225. Berg M, Michalowski A, Palzer S, Rupp S, Sohn K: **An in vivo photo-cross-linking approach reveals a homodimerization domain of Aha1 in S. cerevisiae**. *PLoS One* 2014, **9**:e89436.
226. Kadota Y, Amigues B, Ducassou L, Madaoui H, Ochsenbein F, Guerois R, Shirasu K: **Structural and functional analysis of SGT1-HSP90 core complex required for innate immunity in plants**. *EMBO Rep* 2008, **9**:1209-1215.
227. Koulov AV, LaPointe P, Lu B, Razvi A, Coppinger J, Dong MQ, Matteson J, Laister R, Arrowsmith C, Yates JR, 3rd *et al*: **Biological and structural basis for Aha1 regulation of Hsp90 ATPase activity in maintaining proteostasis in the human disease cystic fibrosis**. *Mol Biol Cell* 2010, **21**:871-884.
228. Kawamura K, Yamada T, Kurihara K, Tamada T, Kuroki R, Tanaka I, Takahashi H, Niimura N: **X-ray and neutron protein crystallographic analysis of the trypsin-BPTI complex**. *Acta Crystallogr D Biol Crystallogr* 2011, **67**:140-148.
229. Schmidt AE, Chand HS, Cascio D, Kisiel W, Bajaj SP: **Crystal structure of Kunitz domain 1 (KD1) of tissue factor pathway inhibitor-2 in complex with trypsin. Implications for KD1 specificity of inhibition**. *J Biol Chem* 2005, **280**:27832-27838.
230. Toyota E, Ng KK, Kuninaga S, Sekizaki H, Itoh K, Tanizawa K, James MN: **Crystal structure and nucleotide sequence of an anionic trypsin from chum salmon (Oncorhynchus keta) in comparison with Atlantic salmon (Salmo salar) and bovine trypsin**. *J Mol Biol* 2002, **324**:391-397.
231. Oliveira MG, Rogana E, Rosa JC, Reinhold BB, Andrade MH, Greene LJ, Mares-Guia M: **Tyrosine 151 is part of the substrate activation binding site of bovine trypsin. Identification by covalent labeling with p-diazoniumbenzamidine and kinetic characterization of Tyr-151-(p-benzamidino)-azo-beta-trypsin**. *J Biol Chem* 1993, **268**:26893-26903.

232. Teich N, Le Marechal C, Kukor Z, Caca K, Witzigmann H, Chen JM, Toth M, Mossner J, Keim V, Ferec C *et al*: **Interaction between trypsinogen isoforms in genetically determined pancreatitis: mutation E79K in cationic trypsin (PRSS1) causes increased transactivation of anionic trypsinogen (PRSS2)**. *Hum Mutat* 2004, **23**:22-31.
233. Buttle DJ, Ritonja A, Pearl LH, Turk V, Barrett AJ: **Selective cleavage of glycyl bonds by papaya proteinase IV**. *FEBS Lett* 1990, **260**:195-197.
234. Rester U, Moser M, Huber R, Bode W: **L-Isoaspartate 115 of porcine beta-trypsin promotes crystallization of its complex with bdellastasin**. *Acta Crystallogr D Biol Crystallogr* 2000, **56**:581-588.
235. Ganesh V, Lee AY, Clardy J, Tulinsky A: **Comparison of the structures of the cyclotheonamide A complexes of human alpha-thrombin and bovine beta-trypsin**. *Protein Sci* 1996, **5**:825-835.
236. Simeonov P, Zahn M, Strater N, Zuchner T: **Crystal structure of a supercharged variant of the human enteropeptidase light chain**. *Proteins* 2012, **80**:1907-1910.
237. Johnson A, Krishnaswamy S, Sundaram PV, Pattabhi V: **The first structure at 1.8 A resolution of an active autolysate form of porcine alpha-trysoin**. *Acta Crystallogr D Biol Crystallogr* 1997, **53**:311-315.
238. Huang Q, Wang Z, Li Y, Liu S, Tang Y: **Refined 1.8 A resolution crystal structure of the porcine epsilon-trypsin**. *Biochim Biophys Acta* 1994, **1209**:77-82.
239. Sottrup-Jensen L, Hansen HF, Pedersen HS, Kristensen L: **Localization of epsilon-lysyl-gamma-glutamyl cross-links in five human alpha 2-macroglobulin-proteinase complexes. Nature of the high molecular weight cross-linked products**. *J Biol Chem* 1990, **265**:17727-17737.