Large Scale Extraction of Protein Protein Interactions Using Data Programming

Aleksandar Dimitriev
Faculty of Computer and Information Science
University of Ljubljana
Večna pot 113, Ljubljana, Slovenia
ad7414@student.uni-lj.si

Stephen Bach
Stanford University
353 Serra Mall, Stanford, CA 94305
bach@cs.stanford.edu

Rok Sosič
Stanford University
353 Serra Mall, Stanford, CA 94305
rok@cs.stanford.edu

Jure Leskovec
Stanford University
353 Serra Mall, Stanford, CA 94305
jure@cs.stanford.edu

Abstract

We present a large scale analysis of protein protein interactions of PubMed, the medical repository of scientific articles with 26 million publicly available abstracts. We found a set of 15 million protein pair sentence candidates and 1.5 million protein pairs and trained a classifier to predict an interaction given its sentences. To obtain labeled data for such a large data set, we used data programming, a weak supervision technique, where users write labeling functions (simple rules) that label large parts of the data set simultaneously. We evaluated our predictions on two well known protein interaction databases. The results show promise for data programming as a way to bypass the infeasible collection of ground truth labels for a data set of this size.

1 Introduction

Protein-protein interaction, henceforth PPI, is a broad term used for almost any kind of physical interaction of a pair of proteins in the cell. They can interact in many ways, such as (in)activation, inhibition, up(down)regulation etc. Similarly, the way interaction is stated in the literature can be very specific, like *phosphorylation* at a specific site, or as broad as just saying that pair *interacts*. This further compounds the difficulty of finding PPIs because it is non-trivial to define whether a sentence constitutes a PPI, as evidenced by the annotator inter-agreement number of 80% [7].

Unfortunately, the literature has grown so large that human annotation is no longer feasible [2]; we must resort to automated techniques and machine learning to find these interactions. Most automatic PPI extraction has focused on small data sets. We attempt to remedy this by conducting PPI extraction on the full PubMed literature, which contains over 26 million abstracts and 138 million sentences. Because there is no labeled ground truth for a data set of this size, machine learning models have not been applied. However, by using data programming, we were able to provide labels for most of the data and trained a supervised classifier.

Data programming [13] is a novel weak supervision approach. It allows us to noisily label data where no ground truth is available, or is too costly to obtain. Instead of going through each example by hand, users write labeling functions that do this for them. For example, in the case of protein-protein interactions, we can write a labeling function that looks for the word *activate* or *inhibit* anywhere in the sentence (or possibly just between the two proteins) and labels the example as positive (a PPI) if

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

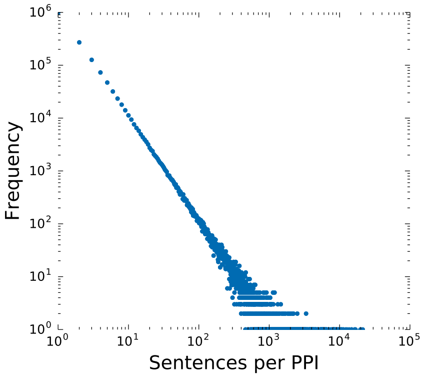

Figure 1: Frequency distribution of the number of sentences containing a given pair of proteins.

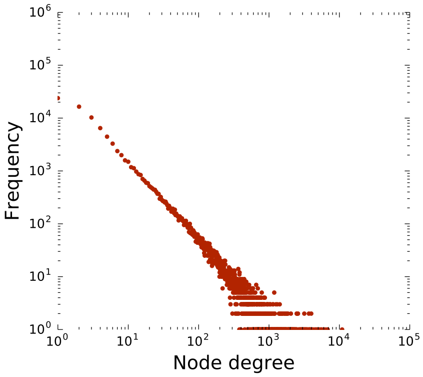

Figure 2: Node degree distribution of the PPI network.

such an interaction keyword is present. The accuracy of these functions is estimated through their overlap, conflict and agreement for the data points. In essence, in the data programming paradigm instead of hand-designing features that may subsequently be useful, users write the aforementioned labeling functions.

2 Related work

Work on protein-protein interaction extraction has, in the beginning, focused on the simpler task of tagging protein and/or gene names in biomedical texts [15, 6, 8]. Although an important step in the PPI extraction pipeline, more recently protein pair interactions are directly extracted by the protein name dictionaries and ad-hoc rules that were previously developed. To bypass the feature stage, kernel methods are commonly employed [11, 1, 5, 10] with support vector machines [16], especially on dependency parse trees. Miwa et al. [10] find that multiple kernel learning outperforms the state-of-the-art on a few commonly small data sets, whereas Yang et al. [17] perform cross-corpora learning to evaluate the similarity between the aforementioned data sets.

There has also been focus on specific interaction types [18, 19], however this can only be performed after an interaction has been found, which is the non-trivial part. Bayesian techniques have also been attempted: Polajnar et al. [11] use Gaussian Processes [11] to predict whether a sentence contains a PPI, but extract neither the protein pair nor the interaction keyword. Chowdhary et al. [4] train a Bayesian network on extracted protein pair and interaction keyword triplets using a dictionary, but this results in many unknown false negatives due to inexact dictionary matches. To the best of our knowledge, PPI detection on this scale has not been tackled. Almost all of the aforementioned work uses a few well-known small corpora like AIMed [3] and there hasn't been focus on scalability. The largest training set is only a few thousand protein pairs, with most containing a few hundred abstracts, whereas the largest evaluation is 1 million MEDLINE abstracts [7], but their approach does not entail any classifier training, and is still orders of magnitude lower than the PubMed literature.

3 Methods

Our approach was the following. First, we split the PubMed abstracts into sentences and parsed them using coreNLP [9]. Due to the size of the data set, we performed this action in parallel using Bazaar[1], which is a wrapper for coreNLP that allows parallelization. We obtain part-of-speech tags, lemmas, dependency tree parse, and other information which is subsequently used. For the subsequent steps, we used Snorkel[2], a Python-based data programming tool. Next, we extracted potential protein interactions by finding pairs of proteins according to a combination of dictionaries and regex patterns.

[1] https://github.com/HazyResearch/bazaar

[2] https://github.com/HazyResearch/snorkel

Protein pair		Frequency
MAPK	p38	29105
CD25	CD4	27199
IL-6	TNF-alpha	22405
IFN-gamma	IL-4	15652
IL-6	IL-8	15442
IFN-gamma	TNF-alpha	14473
IL-10	IL-6	13769
IL-10	IL-4	12274
IL-2	IL-4	11867
MMP-2	MMP-9	11584

Table 1: The ten most common pairs of proteins found in PubMed.

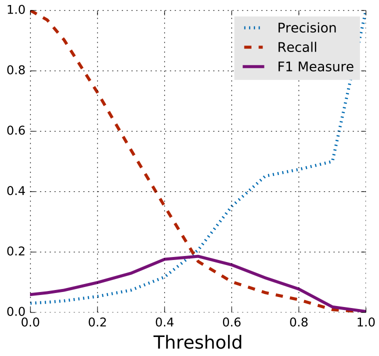

Figure 3: Test set results evaluated on HPRD.

We combined several dictionaries of human protein[3] and gene names[4]. These were augmented with several regex patterns because words might not be exact matches, but can have suffixes like *-alpha*, *-beta*, or a dash followed by a number. This results in a large pool of potential protein interaction pairs and an upper bound on recall, as well as many false positives. We then group these sentences by the protein pair they describe and obtain a set of unique PPI candidates, each with one or more sentences describing it. To obtain features, we perform TF-IDF, taking the collection of sentences for a given pair as the document. Lastly, to improve precision we trained a model to predict whether the pair is actually an interaction according to the sentences. The labeling functions come into play here by automatically annotating most of the data set so that the supervised classifier can be trained. We used logistic regression for scalability and ease of interpretation.

4 Results

The data set consisted of 26,723,368 abstracts containing 137,132,555 sentences, from which we extracted 15,441,834 candidate interaction sentences and 1,704,484 unique protein-protein candidate pairs, thus having approximately 9 sentences per putative interaction. However, the distribution of sentences shown in Fig. 1 looks scale-free and a few protein pairs account for many of the sentences, as can be seen in Table 1. The network constructed from these interactions (where nodes are proteins and edges are formed for each pair found in our data set) appears to be scale-free as well, as shown in Fig. 2.

The labeling functions we used were simple word occurrence indicators from a dictionary of protein interactions words like *association*, *activation*, *upregulation*, etc. Each labeling function outputs 1 if that particular word (or its lemmatizations like *activates*, *activated*, *activating* etc.) is used in any of the sentences and 0 otherwise. For negative labels we looked for words like *tumor*, *gene*, and *DNA*, such that each function outputs -1 if they are found. Consequently, the TF-IDF features with the highest absolute coefficients in the logistic regression model were words like *expression*, *association*, *genes*, and *nucleotide*, with high positive and negative coefficients, respectively.

To evaluate the performance of our model, we compared our predictions to protein interactions found in two databases: HPRD [12] and STRING [14]. Note that only $\sim 25\%$ of the interactions in HPRD were extracted as candidates by our framework (similarly for STRING). One possibility is that the majority of PPIs occur not in the abstract but only in the full text. We trained our model on 90% of the interactions found (randomly chosen). The results from testing on 10% (not seen before) interactions against HPRD are shown in Fig. 3. Recall and precision are normalized by looking only at the $\sim 25\%$ interactions found (otherwise the upper bound on the two measures would be 0.25).

[3] STRING database of human protein aliases: http://string-db.org/download/protein.aliases.v10/9606.protein.aliases.v10.txt.gz

[4] HUGO protein and gene names: *gene_with_protein_product.txt* at http://www.genenames.org/cgi-bin/statistics

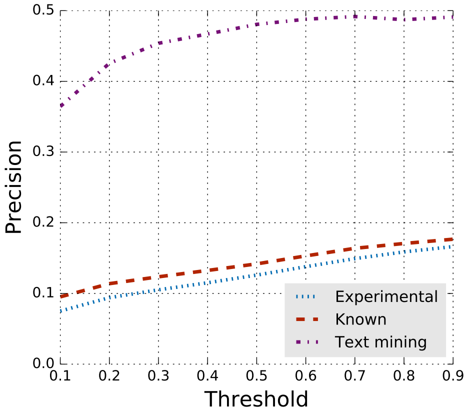

Figure 4: Precision for each STRING ground truth.

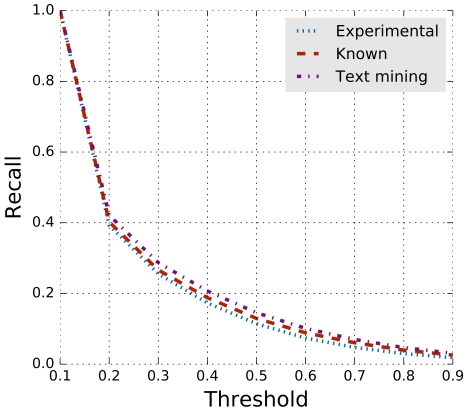

Figure 5: Recall for each STRING ground truth.

On the other hand, STRING offers several types of ground truth for an interaction. We chose 3 measures: experimental, known (combining experimental, fusion, co-occurrence, co-expression and neighborhood information), and text mining. Precision and recall for each of the 3 ground truths can be seen in Fig. 4 and Fig. 5. Text mining PPIs are the one that are most likely to show up in abstracts, and thus our model finds them much more frequently. Moreover, the number of experimental and known PPI ground truth is much lower than text mining, which means that many of the pairs found by text mining are not experimentally validated and this brings down the baseline precision of the other two measures. We can also see that the model is actually gaining useful knowledge from the noisy labels, because precision increases for every ground truth measure – if the labels were uninformative or too noisy, the precision line would be flat.

Thus, our approach shows that the data programming paradigm is a viable approach when constructing large data sets and can be used to bypass costly human annotation. In the future, we intend to go going beyond the sentence level, as well as use pronoun resolution to improve protein pair recall.

Acknowledgments

We thank and Marinka Žitnik for discussion and insightful comments. We are also grateful to Angela Wilkins, Stephen Wilson, Chih-Hsu Lin, and Olivier Lichtarge for providing protein dictionaries and offering helpful feedback on labeling functions.

References

[1] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. A graph kernel for protein-protein interaction extraction. In *Proceedings of the workshop on current trends in biomedical natural language processing*, pages 1–9. Association for Computational Linguistics, 2008.

[2] W. A. Baumgartner, K. B. Cohen, L. M. Fox, G. Acquaah-Mensah, and L. Hunter. Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics*, 23(13):i41–i48, 2007.

[3] R. Bunescu, R. Ge, R. J. Kate, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155, 2005.

[4] R. Chowdhary, J. Zhang, and J. S. Liu. Bayesian inference of protein–protein interactions from biological literature. *Bioinformatics*, 25(12):1536–1542, 2009.

[5] F. M. Chowdhury, A. Lavelli, and A. Moschitti. A study on dependency tree kernels for automatic extraction of protein-protein interaction. In *Proceedings of BioNLP 2011 Workshop*, pages 124–133. Association for Computational Linguistics, 2011.

[6] K.-i. Fukuda, T. Tsunoda, A. Tamura, T. Takagi, et al. Toward information extraction: identifying protein names from biological papers. In *Pac symp biocomput*, volume 707, pages 707–718. Citeseer, 1998.

[7] K. Fundel, R. Küffner, and R. Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[8] A. Koike and T. Takagi. Gene/protein/family name recognition in biomedical literature. In *Proceedings of BioLink 2004 Workshop: Linking Biological Literature, Ontologies and Databases: Tools for Users*, volume 42, page 56, 2004.

[9] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[10] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. Protein–protein interaction extraction by leveraging multiple kernels and parsers. *International journal of medical informatics*, 78(12):e39–e46, 2009.

[11] T. Polajnar, T. Damoulas, and M. Girolami. Protein interaction sentence detection using multiple semantic kernels. *Journal of biomedical semantics*, 2(1):1, 2011.

[12] T. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, et al. Human protein reference database—2009 update. *Nucleic acids research*, 37(suppl 1):D767–D772, 2009.

[13] A. Ratner, C. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. *arXiv preprint arXiv:1605.07723*, 2016.

[14] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, et al. String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, page gku1003, 2014.

[15] L. Tanabe and W. J. Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.

[16] Z. Yang, H. Lin, and Y. Li. Bioppisvmextractor: A protein–protein interaction extractor for biomedical literature using svm and rich feature sets. *Journal of biomedical informatics*, 43(1):88–96, 2010.

[17] Z. Yang, N. Tang, X. Zhang, H. Lin, Y. Li, and Z. Yang. Multiple kernel learning in protein–protein interaction extraction from biomedical literature. *Artificial intelligence in medicine*, 51(3):163–173, 2011.

[18] L. Zhang, D. Berleant, J. Ding, and E. S. Wurtele. Automatic extraction of biomolecular interactions: an empirical approach. *BMC bioinformatics*, 14(1):234, 2013.

[19] S. Žitnik, M. Žitnik, B. Zupan, and M. Bajec. Sieve-based relation extraction of gene regulatory networks from biological literature. *BMC bioinformatics*, 16(16):1, 2015.