KinDER: A Biocuration Tool for Extracting Kinase Knowledge from Biomedical Literature

Daniel Dopp[1*], Adam Morrone[2*], Indika Kahanda[3†]
Affiliation: [1]University of Kentucky, [2]Liberty University, [3]Montana State University
*equal contribution, [†]corresponding author

Abstract—Kinases are enzymes that mediate phosphate transfer. Extracting information on kinases from biomedical literature is an important task which has direct implications for applications such as drug design. In this work, we develop KinDER, Kinase Document Extractor and Ranker, a biomedical natural language processing tool for extracting functional and disease related information on kinases. This tool combines information retrieval and machine learning techniques to automatically extract information about protein kinases. First, it uses several bio-ontologies to retrieve documents related to kinases and then uses a supervised classification model to rank them according to their relevance. This was developed to participate in the *Text-mining services for Human Kinome Curation* Track of the BioCreative VI challenge. According to the official BioCreative evaluation results, KinDER provides state-of-the-art performance for extracting functional information on kinases from abstracts.

Keywords—kinase; proteins; machine learning; biomedical natural language processing; BioCreative; text classification; supervised learning

I. Introduction

With the steady advancement of computing power and decline in memory cost over the years has come the ability to work with increasingly larger data sets in more complex ways. These opportunities have opened up the relatively new fields of computer-driven bioinformatics and natural language processing. These two areas, when in conjunction, can allow for automatic extraction of important information from biomedical literature written in plain, unstructured text. An example of where this is advantageous could be having the ability to intelligently search through all existing journal articles about a specific cellular structure in order to aggregate current knowledge about that structure. This process is currently done by hand via human curators. As there are literally millions of journal articles published each year, there is much room for improvement. One such group of bio-entities of high interest are the human protein kinases, a specific type of enzyme that can phosphorylate (add a phosphate to) other proteins. This process can activate or inhibit various other proteins, and plays an important role in cellular communication and hormone action (1). An automated, intelligent search tool for protein kinases could dramatically improve the curation process and potentially assist the scientific community in better understanding these important proteins. This report describes the development, implementation, and testing of a pipeline to do just this. In particular, the KinDER (Kinase Document Extractor and Ranker) pipeline allows users to enter a specific human protein kinase in addition to an axis (be it disease, or biological function), and returns predictions of which documents from either a collection of PubMed database journal articles or MEDLINE database abstracts contain relevant information to those criteria. Furthermore, KinDER can be used to predict ~500 character snippets of text which contain relevant information to the search criteria. This tool was developed in order to participate in the Track 2 (Text-mining services for Human Kinome Curation) of BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology) challenge held in 2017.

II. Methodology

KinDER is composed of two main components: 1) Document Retrieval component which retrieves documents annotated with kinases and axis terms using dictionaries and 2) Document Ranking and Information Extraction component which uses machine learning to rank those documents based on relevancy, as depicted in the figure below. There are a significant number of data processing steps that occur inside these two components that make up the full KinDER pipeline (Fig. 1). The following subsections will describe those steps in more detail.

A. Data

We use the BioCreative Track 2 official dataset as the input data for KinDER. Included in this were PubMed articles (approx. 260,000) and MEDLINE abstracts (approx. 4.4 million) in BioC format (2), lists of kinase names and synonyms, and a gold standard dataset of kinase names and associated relevant documents. This challenge has three subtasks: Abstract Triage, FullText Triage, and Snippet Selection. In order to annotate documents based on their relevancy to the disease axis (DIS), we considered the HPO (3), ORDO (4), NCITd (5) (hand culled subset comprised of only disease related subsections of NCIT), PDO (6), OAE (7), IDO (8), ICD10 (9), MeSH (10) and DOID (11) bio-ontology annotation dictionaries available from the NCBO annotator website (12). For annotating the biological process axis (BP) we considered the GO (13) dictionary from NCBO as well as a concept recognition dictionary developed by Funk et al. (14-

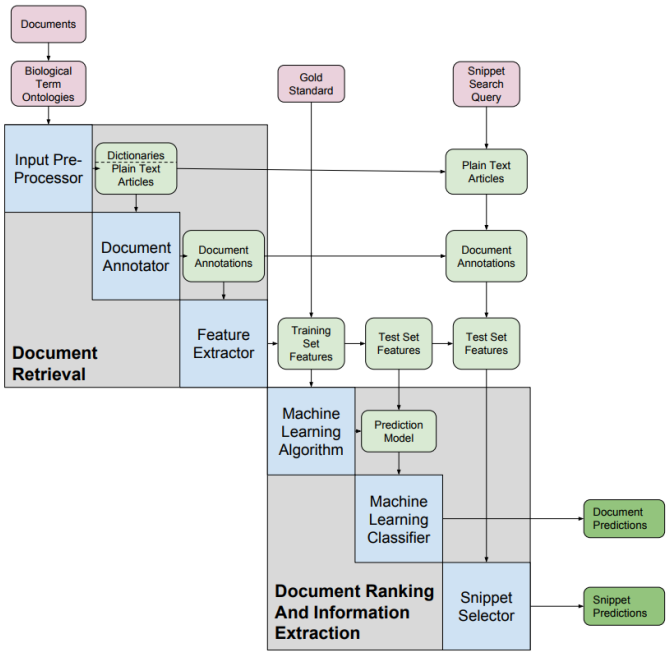

Fig. 1. N2 (N-squared) Diagram of KinDER Pipeline.

15), which we refer to as GO2. According to our preliminary results (see Fig. 2 and Fig. 3), we chose HPO and NCITd for DIS annotation and GO for BP annotation.

B. Input Pre-Processing

This first stage takes the input data described previously and converts it into formats useful for the annotation stage. The document annotator chosen for this pipeline was ConceptMapper (CM) (16), an annotation engine for the Apache UIMA framework (17). This tool is discussed in more detail in the next subsection. Input processing has two main steps: article extraction and dictionary creation. In order to handle the large collection of documents provided in BioC format, a custom python library was developed using lxml's iterparse object (18), which significantly improves memory efficiency compared to existing libraries. This step also consisted of writing scripts which converted BioCreative's XML lists of kinase synonyms, NCBO's CSV dictionaries, and the GO.obo dictionaries into XML formatting for ConceptMapper. Original BioCreative kinase dictionaries were also enhanced by using kinase synonyms from UniProt (19), adding any new synonyms that were not already provided. These appended dictionaries were then run through string-processing scripts to convert Roman numerals to Arabic numerals and remove unnecessary spaces between words. These variants were added as additional synonyms, as opposed to replacing old ones.

C. Document Annotation

As mentioned previously, CM was chosen as the primary dictionary look up tool for the KinDER pipeline. This tool takes a directory of text files to annotate, as well as a dictionary to annotate with. By default, it automatically handles all stemming and stop word removal. For improving efficiency, the CM output files were compressed into custom summary structs containing basic information on each annotation made including the term that was a "hit" as well as its position in the document and the canonical term which it refers to. In addition, these structs stored metadata about the documents such as the number of total hits, the number of unique terms, and sets/counts of matched terms.

D. Feature Extraction

The main goal of this stage is to provide meaningful information to enable successful downstream classification of documents as relevant or irrelevant. This task is broken down into three main processes: 1) cross-reference validation, which attempts to filter out obviously irrelevant documents (i.e. documents that do not contain both a kinase and an axis term), 2) feature vector generation, which creates vectors of meaningful metadata that the downstream machine learning algorithm can use to learn from, and finally 3) creation of corresponding binary labels for the training subset of feature vectors based on the BioCreative gold standard.

In generating feature vectors to train the machine learning based Document Ranking portion of the KinDER pipeline, two types of features were generated, both using the set of documents that made it through the initial round of cross referencing based selection. The first approach is the standard Bag of Words (BOW) feature model which uses TFIDF features values (20). In our model, two and three-gram term combinations were also included in the vocabulary.

For the second approach, six metrics were chosen as features (referred to as the "engineered feature set" or ENG). They are, Kinase Score: the number of kinase annotations normalized by total words, Axis Score: the number of axis term annotations normalized by total words, Relevancy Score: The product of the kinase score and axis score, Proximity Score: The minimum number of words separating a kinase and axis annotation, and Proximity 10-Count and Proximity 50-Count: the number of pairs of kinase and axis annotations that are within 10 and 50 words of one another. We apply standard pre- and post-processing techniques including stemming, and the removal of standard English stop words before constructing both types of above features.

E. Machine Learning Model Selection and Training

We model this problem as a binary classification problem in which we distinguish between *relevant* vs *irrelevant* articles. We used the Scikit-learn (21) Python machine learning library for implementing the machine learning models. An initial model selection phase was conducted comparing three supervised classification algorithms and it was determined that Support Vector Machines (SVMs) were the most promising avenue. For the BP FullText and BP Abstract subtasks, eight SVM models were evaluated based on SVM kernel (linear vs gaussian) and feature type (BOW vs ENG). For the DIS FullText and DIS Abstract subtasks, sixteen models were

evaluated based on kernel (linear vs gaussian), features (BOW vs ENG) and the ontology (HPO vs NCITd).

Each classifier model was trained using the full set of gold standard relevant documents and only a 10-20% random sample of the total irrelevant documents. Several classifiers utilizing BOW were also restricted to feature vectors of total length 100,000 (only the 100,000 most common terms). It was found through preliminary experiments that restrictions on training set size and feature vector length did not significantly impact model scoring (data not shown). Computational efforts were performed on the Hyalite High-Performance Computing System, which is operated and supported by University Information Technology Research Cyberinfrastructure at Montana State University.

F. Test Kinase Classification and Ranking Paradigm

Saved classification models were used for ranking the documents based on their relevance. Document subsets for the test data created in the document annotation stage were fed into their respective classifiers and assigned a classification and confidence score. All documents within a subset were sorted based on the classifier confidence score.

G. Snippet Selection

In order to extract a snippet of text 500 characters or less which contained sufficient relevant information to make an accurate annotation for the article, we used the following method. First, the two annotated terms, kinase and axis, that were in closest proximity in the article was identified. Next an approximate 500-character excerpt encapsulating the two terms as close to the middle as possible was captured. Finally, the excerpt length was rounded down in order to begin and end at the start and end of sentences.

III. Experimental Setup

A. Document Retreival

To determine the best ontology for document retrieval, standard metrics of precision, recall, and F-1 Score were utilized. To calculate these, two sets of articles of equal size, one containing gold standard positives and the other containing gold standard negatives were created. The default settings of ConceptMapper were used.

B. Document Classification and Ranking

Three classifier models were compared: K-Nearest Neighbors, Support Vector Machines, and Naïve Bayes. For evaluating the machine learning models, in both comparing models in the initial model selection phase and selecting hyperparameters when tuning models for the ranking phase, a 3-fold stratified cross validation technique (22) was used. We used AUROC (23) as our evaluation measure. In training models for ranking, a grid search with nested cross validation (24) approach was used.

IV. Results

A. Document Retrieval

According to the results of ontology comparison for both the Abstract and the FullText sets depicted in Fig. 2 and Fig. 3, we chose NCITd and HPO for the disease axis, and GO for the function axis.

B. Ranking and Information Extraction

Initial SVM scoring results were promising for the BOW model which significantly outscored the engineered feature set as seen in Fig. 4 and Fig. 5. Furthermore, the linear kernel SVM performed best across all subtasks and ontologies, slightly beating out gaussian kernel models, likely due to a larger hyperparameter search space used because of more efficient training times.

C. Official BioCreative Track 2 Results

In addition to prediction made by our machine learning models described above, we made a set of submissions based on several *rule-based* models that were each using the six ENG feature types. In this method, each document was assigned a relevancy score or an aggregate score of all calculated feature vectors, and the predictions were made purely on this basis without any machine learning. According to Table I which shows the MAP (mean average precision –

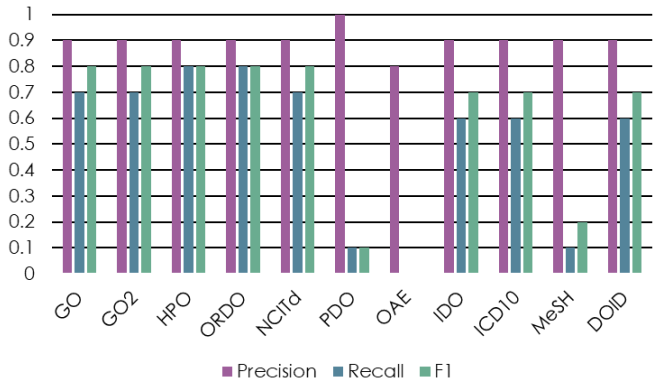

Fig. 2. Bio-ontology comparison for Abstract subtask.

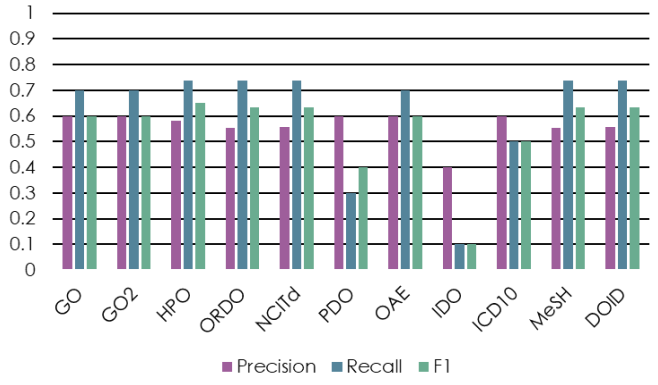

Fig. 3. Bio-ontology comparison for FullText subtask.

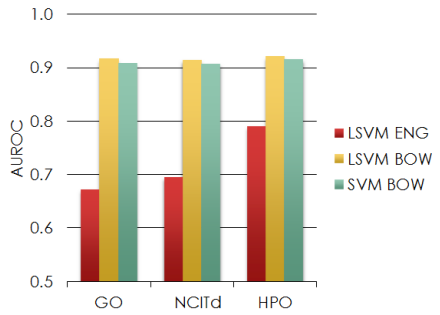

Fig. 4. SVM performance on Abstract subtask. LSVM/ SVM: SVMs using linear/gaussian kernel, BOW: bag-of-words features, ENG: engineered features.

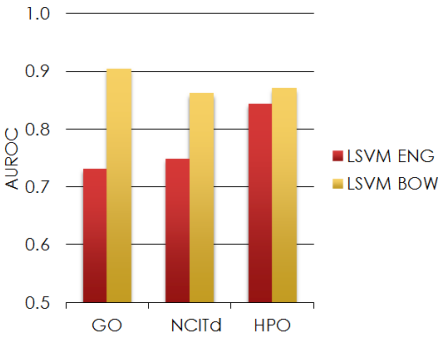

Fig. 5. SVM performance on FullText subtask. LSVM/ SVM: SVMs using linear/gaussian kernel, BOW: bag-of-words features, ENG: engineered features.

the higher is better) scores for our top three submissions in each of the subtasks, and as predicted by our preliminary tests, using a BOW machine learning model outperformed engineered feature sets. In addition, however, we observe that the rule-based methods using engineered features outperformed the machine learning methods for subtask 2 (FullText). It is important to note that, for the Abstract BP subtask, KinDER provides state-of-the-art performance among all submissions.

V. Conclusion and Future Work

To conclude, KinDER has been shown to have the potential to become an effective tool for automating biocuration efforts, particularly in the functional domain. However, much work is still necessary to improve both the recall of document annotation and the ranking paradigms.

The creation and experimentation with KinDER revealed several additional avenues of necessary research in the field. Perhaps the most glaring problem was the lack of a comprehensive list of synonyms for proteins kinases. Though several synonym lists exist, we found through experimentation that none are exhaustive when it comes to the various ways that authors notate kinases. We also found that manually expanding the synonym lists (for instance changing roman numerals to numbers and vice versa) increased our recall. A more comprehensive list would improve results for computer-driven biocuration.

TABLE I. Official BioCreative results.

Subtask	Model	MAP
Abstract - DIS	HPO - LSVM BOW	0.098
	NCITd - LSVM BOW	0.096
	NCITd - Relevancy Score[a]	0.080
Abstract - BP	GO - LSVM BOW	0.201
	GO - Relevancy Score[a]	0.197
	GO – LSVM BOW[b]	0.187
FullText - DIS	NCITd - Aggregate Score[a]	0.118
	NCITd - Relevancy Score[a]	0.112
	HPO - Relevancy Score[a]	0.100
FullText - BP	GO - Aggregate Score[a]	0.293
	GO - Kinase Score[a]	0.278
	GO - Proximity Score[a]	0.271

a. Rule-based methods.

b. Using only 20% of the training data to train the classifier.

Within the machine learning portion of our work, many improvements could be made in the comprehensiveness of model selection and training. The extent of model selection was fairly limited due to resource constraints for this study and examining further models e.g. random forests or neural networks may lead to improved predictions over SVMs. Training SVMs is a highly resource intensive process for larger datasets, making it difficult to test more than a handful of hyperparameters. A more extensive parameter sweep trained on a larger, more balanced dataset would likely improve KinDER's performance. In addition, should a golden standard data set containing examples for every kinase we are interested in ranking be released this problem could be rethought as a multiclass classification problem which would simplify many aspects of the problem.

In addition to triage improvements, our process for snippet selection did not incorporate any machine learning techniques. If we were able to incorporate ML, we believe our snippet selection process would improve as well. Lastly, though KinDER is a fully functional standalone pipeline, its current web user interface is very limited. The evolution of KinDER into an end-to-end tool for biocuration could lend itself well to future bio-curation projects.

Acknowledgement

The authors would like to thank Karen Stengel of Montana State University for her input as a domain expert for improving Kinase dictionaries. We would also like to thank the National Science Foundation for funding the REU (Research Experience for Undergraduates) program at Montana State University within which this research was conducted.

References

1. Marieb,E., Hoehn,K., (2013) The Endocrine System. *Human Anatomy & Physiology*, **9**, 606-619.

2. Comeau,D., Doğan,R., Ciccarese et al. (2013) BioC: A minimalist approach to interoperability for biomedical text processing. *Database*, 15.

3. Robinson,P., Köhler,S., Bauer et al. (2008) The Human Phenotype Ontology: A Tool for Annotating and Analyzing Human Hereditary Disease. *The American Journal of Human Genetics*, **83**, 610-615.

4. Vasent, D., Chanas,L Malone et al. (2014) ORDO: An Ontology Connecting Rare Disease, Epidemiology and Genetic Data. *Phenotype data at ISMB2014*.

5. Sioutos,N., Coronado,S., Haber et al. (2007) NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, **40**, 30-43.

6. (2016) Pathogenic Disease Ontology *http://bioportal.bioontology.org/ontologies/PDO*

7. He,Y., Sarntivijai, S., Lin et al. (2014) OAE: The Ontology of Adverse Events. *Journal of Biomedical Semantics*, **5**, 29.

8. Cowell,L., Smith,B. (2011) Infectious Disease Ontology. *Infectious Disease Informatics*, **27**, 373-396.

9. National Center for Health Statistics. (2016) *https://www.cdc.gov/nchs/icd/icd10.htm*

10. U.S. National Library of Medicine. Medical Subject Headings – Home Page. *https://www.nlm.nih.gov/mesh/*

11. Schriml,L., Arze,C., Nadendla et al. (2012) Disease ontology: A backbone for disease semantic integration. *Nucleic Acids Research*, **40**.

12. Jonquet,C., Shah,N., Cherie et al. (2009) NCBO Annotator: Semantic Annotation of Biomedical Data. *Iswc*, 2-3.

13. Ashburner,M., Ball,C., Blake et al. (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, **25**, 25-29.

14. Funk,C., Baumgartner,W., Garcia et al. (2014) Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. BMC Bioinformatics, **15**, 59.

15. Funk,C., Kahanda,I., Ben-Hur et al. (2015) Evaluating a variety of text-mined features for automatic protein function prediction with GOstruct. *Jounral of biomedical semantics*, **6**, 9.

16. Tanenblatt,M., Coden,A., Sominsky, I. (2010) The ConceptMapper Approach to Named Entity Recognition. *Proceedings of the Seventh conference on International Language Resources and Evaluation LREC10*, 546-551.

17. ASF (2013) Apache UIMA. *http://uima.apache.org/*

18. Richter,S. (2015) lxml - XML and HTML with Python. http://lxml.de/.

19. Bateman,A., Martin,M., O'Donovan et al. (2017) UniProt: The universal protein knowledgebase. *Nucleic Acids Research*, **45**, D158-D169.

20. Manning,C., Raghavan,P., Shutze,H. (2009) An Introduction to Information Retrieval. 569.

21. Pedragosa,F., Varoquaux,G., Gramfote et al. (2012) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830.

22. Witten,I., Frank,E., Hall, M. (2016) Data Mining: Practical Machine Learning Tools and Techniques 3rd Edition.

23. Bewick,V., Cheek,L., Ball,J. Statistics review 13: Receiver operating characteristic curves. *Critical Care*, **8**, 508-512.

24. Cawley,G., Talbot,N. (2010) On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, **11**, 2079-2107.