USING TEXT MINING TECHNIQUES TO ASSIST GENE RELATED ANNOTATION

by

Ruoyao Ding

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Fall 2017

USING TEXT MINING TECHNIQUES TO ASSIST GENE RELATED ANNOTATION

by

Ruoyao Ding

Approved: _____

Kathleen F. McCoy, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: _____

Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____

Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Vijay K. Shanker, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Cathy H. Wu, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Ben Carterette, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Hongfang Liu, Ph.D.
Member of dissertation committee

ACKNOWLEDGMENTS

It is my pleasure to thank those who made this dissertation possible.

Foremost, I would like to express my deepest gratitude to my advisors Dr. Cathy H. Wu and Dr. Vijay K. Shanker for the continuous support of my Ph.D. study. I am very grateful for their patience when things were moving slowly, and I thank them for always being available for guidance, encouragement and advice. It would not be possible for me to reach this point without their generous help.

Besides my advisors, I would like to thank the rest of my dissertation committee, Dr. Ben Carterette and Dr. Hongfang Liu, for sharing their time and expertise, and providing me with insightful comments and suggestions.

I also would like to thank my lab mates in the University of Delaware Text mining & Natural Language Processing lab: Yifan Peng, Gang Li, Samir Gupta, and Ashique Mahmood, for the days we were working together, and for all the fun we have had in the last six years.

Special thanks go to Cecilia N. Arighi, Qinghua Wang and Hongzhan Huang of Delaware Biotechnology Institute, for providing annotations and feedbacks for my tools.

Lastly, and most importantly, I would like to thank my parents for raising me and loving me unconditionally. Their love and understanding will support me spiritually throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Biomedical researchers usually describe their experimental results in research publications. With the rapid growth of biomedical publications, the information of interest needs to be extracted automatically to avoid the time consuming and labor intensive process.

In this dissertation, we seek to help in the process of gene related annotations, by extracting the unstructured information buried in the literature and providing means to structure the extracted information. We start by recognizing gene names in the literature and linking them to database records. Based on this work, we develop a system which automatically selects articles that are about a specific UniProt protein entry. Next, we describe our approach for mining information related to another important bio-named entity, protein complex. Finally, we present our work on assisting the curation of gene annotation.

In each of these tasks, we conduct experiments to evaluate the efficacy of our approaches. The results show that our systems achieve good performances, and can be used to assist the annotation process related to genes.

Chapter 1

INTRODUCTION

1.1 Motivations and Contributions

Most biological experimental results are described in published literature. Researchers need to find the information of their interest from the research literature in order to conduct and interpret their own experiments. However, with the rapid growth of biomedical publications, molecular biology has become an information-saturated field. Manually extracting information from the literature usually is a time consuming and labor-intensive process. As a result, a major focus of bioinformatics research is to automatically extract information from published literature, using text mining techniques.

Named entity recognition (NER, a task to recognize entities mentioned in natural language text) and relation extraction (a task to identify relations between the recognized entities) are two common tasks in the biomedical text mining field. They both extract the unstructured information buried in the literature, and provide a means to structure the extracted information, such as putting it into databases. In my dissertation, I will present my work on the recognition and normalization for two important bio-named entities, genes and protein complexes. I will also describe my work on assisting the annotations of these two entities, which is based on relation extraction.

This dissertation's focus reflects the primary importance genes and their products play and their importance in biological and medical studies. In order to comprehensively annotate gene/protein records and to support queries from biologists from a variety of backgrounds who may use different names to refer to a gene/protein of interest, curators of knowledge bases, such as UniProt Knowledgebase (UniProtKB) [1], need to capture the full range of names and symbols by which a protein/gene is known. Automatic detection of gene/protein names in the literature and their linkage to database records, also known as gene normalization (GN), is being developed as an alternative to the time-consuming practice of manual extraction of names. GN has become an essential component of many text mining systems and database pipelines. For example, our group uses GN of kinase and substrate mentions to integrate phosphorylation information from the text mining tool RLIMS-P [2] into iPTMnet [3], a bioinformatics resource for protein post-translational modifications.

Despite the large body of work [4, 5, 6, 7, 8, 9, 10, 11, 12] conducted in gene normalization, it remains inadequate for certain taxonomic groups, particularly plants. This is an initial motivation to work on this task. We developed a gene normalization system, called pGenN (pivot-based Gene Normalization) [13]. The system consists of three steps: dictionary-based gene mention detection, species assignment, and intra species normalization. It accounts for the naming conventions for genes in plant species, however, the notions and approaches developed in pGenN is generalizable to any species. For evaluation purposes, we had to develop our own curated literature corpora due to the non-availability of annotated plant literature in existing corpora.

Evaluation using these data sets, shows that our system achieved state-of-art performance on abstract level plant gene normalization. Additionally, we applied pGenN on all plant-related Medline abstracts. The results, which are updated monthly in sync with PubMed, are stored in a local database called pGenN_DB. To make these results accessible to the community, we developed a web interface (http://biotm.cis.udel.edu/gn) for multiple modes of querying and downloading of results. In addition to developing the best performing GN system on plant literature, our work also contributes in two aspects: (1) it introduces the notion of pivot that can be used in other named entity recognition and normalization tasks; and (2) the development of a method to automatically generate a large corpus can be generalized for other types of entities as well as for other specific sub-domains. Techniques here may also be applicable for distant supervision for development of NER tools.

We have continued our work to GN in full length articles and focus on the species assignment process of genes in the results section. Based on our study, this process is different from normalizing genes in the abstract, while other techniques that have been introduced in abstract level gene normalization are still directly applicable to full length articles. We showed that our approach on species assignment of genes in the results section yields good performance. Our approach can be distinguished from previous approaches because it treats different full length article sections differently, especially the results section from the abstract. We believe that this idea can be fruitful for other text mining tasks applied on full length articles.

A second aspect of this dissertation concerns extending computationally mapped bibliography for UniProtKB. UniProtKB is a publicly available database with access to vast amount of protein sequence and functional information. Articles used in the annotation of information for proteins are associated from the protein's entry. Rather than curating all articles about a given protein, UniProt chooses a representative set that maximizes the information content. Thus, an article related to one UniProt entry with potential useful information content may not have been included in that entry. Furthermore, UniProt focuses the curation effort on the most widely studied species, therefore some organisms for which experimental data may be available are not actively annotated. To complement the curated literature set in UniProtKB/Swiss-Prot with additional publications and to add relevant literature to UniProtKB/TrEMBL entries that have not yet been curated, UniProt compiles additional bibliography from external sources.

We developed a system to automatically select articles that are about a specific UniProt protein entry, and suggest these articles to the entry as additional bibliographies. The system employs a 2-stage process: (1) using pGenN to detect plant gene/protein mentions in a given abstract and normalize them to UniProt entries, and (2) utilizing a trained model, to predict whether the abstract is about a gene and hence should be linked to the UniProt entry for that gene. Evaluation results showed that our system obtained very high precision, thus can be used to meet the need of suggesting additional bibliographies for UniProt. Our notion of aboutness can be extended for other types of entities and in general be used for other tasks, such as ranking articles in the information retrieval tasks. We have conducted the full-scale processing of

PubMed for 8 common plant species for suggesting UniProt accession-PubMed ID pairs to UniProt. The literature collected by our system has already been integrated in the UniProt production of computationally mapped literature. It is updated every three months and can be accessed via the UniProtKB protein entries enhanced view of publications. So far, over nine thousand UniProt accession-PubMed ID pairs have been suggested of which nearly four thousand were not previously found in UniProt.

So far, we have introduced our work related to genes/proteins (as in most gene related text mining tasks, we do not distinguish gene from protein). Since proteins often function as components of larger complexes to perform a specific function [14], and some molecules exist only in certain types of complex (e.g. collagen type I, EBI-2325312) [15], it is important to apply text mining techniques to mine protein complex related information. The next part of this dissertation focuses on this aspect.

Protein Ontology [16], Complex Portal at IntAct [15] and CORUM [17] are some well-known resources which containing information about protein complexes. Entries for individual complex in these resources commonly include the protein complex name and its synonyms, the species and the component proteins of the complex. These types of information are usually obtained from individual experiments published in scientific articles, rather than from the high-throughput experiments data [17]. Currently, the coverage of protein complex in these resources is limited, and most of the included information focuses on a few species. For example, there are only 1905 protein complex entries in Complex Portal (07/2017 release), 570 entries are from Homo sapiens (NCBI Taxonomy ID: 9606), 531 from Mus musculus (10090),

424 from Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (559292), with the remaining 380 from 15 other different species.

We conducted our work on three subtasks that are intended to assist in the improvement of the coverage in these resources: (I) protein complex mention recognition, (II) protein complex-component relation extraction, and (III) protein complex component-component relation extraction.

Although several systems have been developed for recognizing bio-named entities such as genes and diseases, as far as we are aware, there is no system publicly available for protein complex mention recognition. Due to the limit of coverage in the current protein complex resources, we did not work on the normalization task, but instead try to assist in the improvement of the coverage in these resources. Since the annotation of a protein complex requires its components information, detecting the relation between protein complex and protein will be helpful for complex annotation (Of course, this information will also help with the annotation of the protein). Frequently, complex formation can be found without the name of the protein complex. In these cases, we will extract the protein complex component and component relation. Similar to the complex-component relation, extraction of the component-component relation will also assist the annotation for both the protein complex and the protein.

Evaluation results show our system for protein complex mention recognition achieves good performance. Since there is no system publicly available for protein

complex recognition, our system will be the first one to serve the community. In this task, we create a dictionary of protein complex names by extracting protein complex names from the literature using high confidence. This idea can also be used in other situations when a dictionary of names needs to be created. For the two complex related relation extraction tasks, we have shown that our approaches achieves very high precision. Thus, we believe our work on the three subtasks can provide high confidence text evidence for the protein complex resources, and can be used to assist in the improvement of the coverage in these resources.

The final part of the dissertation focusses on assisting the curation of gene annotation. The first part of the work on GN is an essential step for curation of gene annotation since allow us to identify the places where a gene is mentioned in the literature. Further work on the mining of protein complex related information can be used for the curation of the proteins for complex formation information. Extending this part, we consider the annotation for one of the domains of gene annotations.

The Gene Ontology (GO) [18] is a resource that supplies information about gene product function using ontologies to represent biological knowledge. These ontologies cover three domains: (I) Cellular Component (CC), the parts of a cell or its extracellular environment; (II) Molecular Function (MF), the elemental activities of a gene product at the molecular level, such as binding or catalysis; and (III) Biological Process (BP), operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units (cells, tissues, organs, and organisms). GO annotation is a process which assigns gene functional information

using GO terms to relevant genes in the literature. It is a common task among the Model Organism Database (MOD) groups. Manual GO annotation relies on human curators assigning gene functional information using GO terms by reading the biomedical literature. This process is very time-consuming and labor-intensive. As a result, many MODs can afford to curate only a fraction of relevant articles [19].

Many systems have been developed for automatic GO annotation [20, 21, 22, 23, 24, 25, 26]. Despite the fact that there are three types of GO terms (CC, MF, and BP) and the kind of textual evidences for each type are different, current approaches for automatic GO annotation tend to use a single approach for annotations of all types of GO terms. In contrast, we treat annotation with different types of GO terms as individual subtasks, where we cast each subtask as relation extraction between gene and other entities. Since our previous work covers the mining of protein complex related information, and protein complex falls into the CC domain, in this work, we consider annotating genes using GO terms from the CC domain.

GO terms from the CC domain can be essentially divided into two sub-hierarchies: subcellular location terms, and protein complex terms. We believe that we can assist in the curation of the annotation process using GO terms from these two sub-hierarchies by considering two relation extraction tasks: (1) extract cases where a protein is found to be in a subcellular location, and (2) extract cases where a protein is a subunit of a protein complex. We tested our approach on the BC4GO test set [27], and evaluation results show that our approach achieves very high precision. Thereby

we believe our system can be used as a useful tool for the bio-annotators to accelerate the process of GO annotation.

1.2 Dissertation Overview

In Chapter 2, we introduce related work and concepts necessary for understanding the work presented in this dissertation. Chapter 3 discusses our approach for gene normalization. In Chapter 4, we present our system automatic identify additional bibliography for given UniProt protein entries. Chapter 5 reports our work for mining protein complex related information. Both protein complex mention recognition and two protein complex-associated relations (complex-component relation, component-component relation) extraction are discussed. Chapter 6 discusses our approaches for gene annotation using GO term from one particular domain, Cellular Component. In Chapter 7 we summarize this dissertation and outline directions for future work.

Chapter 2

RELATED WORK

This chapter introduces related work and concepts necessary for understanding the works presented in this dissertation. Section 2.1-2.3 are concerned with named entity recognition and normalization, describing related works in the gene mention task, the gene normalization task and other biomedical named entity recognition and normalization tasks respectively. Section 2.4 introduces works on biomedical relation extraction. Finally, works related to application of text mining to Gene Ontology annotation are discussed in Section 2.5.

2.1 Gene Mention Recognition

The task of gene mention (GM) recognition is to automatically recognize gene/protein names mentioned in text. This task has received wide attention, and has been used in several challenge evaluations such as BioCreative I [28] and BioCreative II [29]. Other annotated corpora have also been constructed for system development and evaluation purpose.

There are several challenges of the gene mention recognition task. (1) the number of gene names are in the millions and new names are created continuously. (2)

Name variations: Authors usually do not use proposed standardized gene names. (3) Polysemy: gene names often also refer to other entities such as disease names.

As is typical in gene mention/normalization literature, we do not differentiate between names of genes or their protein products.

2.1.1 Gene Mention Recognition systems

Approaches to gene mention recognition can be categorized into two major classes: (1) rule-based approaches and (2) machine learning-based approaches.

While rule-based gene mention recognition approaches do not require annotated data to train a system, they do require domain experts to be closely involved in developing the rules. The following three systems are examples of gene mention detectors that rely on manually developed rules.

Hanisch et al. [30] presented a dictionary matching based system that detects fly, mouse and yeast gene names from biomedical text. Fukuda et al. [31] proposed a method which incorporates two new concepts called c-term (a concept based on orthography) and f-term (a concept that is based on terms that correspond to types of biological entities) (details about those two terms will be introduced later in the Gene Normalization chapter). Narayanaswamy et al. [32] developed a system which extracts multiple types of named entities including gene names. Their system is based on a manually developed set of rules that rely upon some crucial lexical information,

linguistic constraints of English, and contextual information and develop the notion of c-term and f-term in named entity recognition.

The machine learning-based gene mention recognition approaches require annotated data to train a system. Thus, domain expertise is now required in the development of the data annotation and less during the system training.

In the machine learning-based gene mention recognition approaches, the gene mention recognition task is often treated as a sequence labelling problem (label the tokens in the text using the tags). **BIO** (or **IO**) tags for the text are commonly used to represent the boundaries of gene mentions where **B** represents the beginning of the gene name in text, **I** is assigned to a token inside the gene mention and **O** is assigned to token that are outside the gene mentions.

Among the machine-learning based systems, BANNER [33] is widely used for recognizing biomedical named entities including gene mentions. It is based on conditional random fields and applied orthographic, morphological and shallow syntax features. Liu et al. [34] trained a classification system using Conditional random field (CRF) [35] to classify each word in the literature to the BIO tags. They applied BioThesaurus [36], a comprehensive collection of gene names to entries in the UniProt Knowledgebase, for dictionary lookup and used the matching information as a feature. Huang et al. [37] considered the gene mention task as a classification problem and applied support vector machine (SVM) [38] to solve it. Chen et al. [39] proposed a gene mention recognition system for biomedical literature using a dictionary and

Support Vector Machine. Zhou et al. [40] proposed an ensemble of classifiers for gene mention recognition. They combined three classifiers, one Support Vector Machine and two discriminative Hidden Markov Models using a simple majority voting strategy. Other machine learning based gene mention systems can be found in [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51].

2.1.2 Gene Mention Recognition corpora

High quality gene mention corpora are important for the development of any type of gene mention recognition system. Even for the rule-based system, more accurate rules can be made by analyzing the instances in the corpora.

The GENIA corpus [52] is a collection of 2000 abstracts extracted from Medline database. Multiple biomedical named entities, including gene names, are annotated. It is focused on a subset of human hematology. The PennBioIE corpus [53] consists of 1414 Medline abstracts on cancer. 24 types of biomedical named entities, including gene names, are annotated. The BioCreative 1 GM corpus [28] contains 15,000 sentences from Medline abstracts. Genes and related entities mentions are annotated. The BioCreative 2 GM corpus [29] contains 20,000 sentences from Medline abstracts (15,000 of which were used previously in BioCreative 1).

2.2 Gene Normalization

The task of gene normalization (GN) is to automatically link a gene mention to a database entry for the gene (product).

Other than the challenges stated in the gene mention recognition task, the challenges for the gene normalization task also include: (1) identifying the species for the gene mentions since most gene (product) knowledge bases contain species-specific entries, and (2) disambiguation since multiple gene entries may share the same short name (symbol).

2.2.1 Gene Normalization systems

The following were the top performing systems in the BioCreative I [54] and BioCreative II [55] Challenge GN Tasks. ProMiner [5] is a dictionary-based GN system which is characterized by the inclusion of different biomedical dictionaries and manual clean-up of a dictionary. BioTagger [4] tackles the GN problem with the steps: (1) dictionary lookup to obtain a list of mapping pairs of gene mention and database identifier, (2) machine learning that considers features such as the gene mention recognition, name ambiguity, and token shape information, and (3) a similarity based method to associate Entrez gene records with phrases detected by the gene mention tagger. GNAT [6] is a GN system encompassing four steps: named entity recognition for genes and species, validation of gene mentions, correlating gene mentions with species, and finally gene mention disambiguation. GeNo [7] tackles the GN problem by employing a carefully crafted suite of symbolic and statistical methods.

In BioCreative III [56], the GN task was further extended to cover genes of all relevant species in the literature corpora. Among the systems, Bhattacharya et al. [8] tried to associate a species name with a gene name by considering their proximity to the gene mention. Dai et al. [9] employed a multistage GN procedure and selected dictionary entries from only the top 22 most common species in NCBI (from 7283 species) to speed up the GN process. A document-level gene normalization system, called GeneTUKit [10], employed features from the local context as well as the global context of the whole full-text article. GenNorm [11] follows three steps: gene name recognition, species assignment, and species-specific gene normalization, and uses SR4GN [57] for assigning species to gene mentions. GenNorm has been widely used in text mining systems that require GN, such as in PubTator [58] and in an event extraction pipeline [59]. GNormPlus [12], as an updated version of GenNorm, refined the gene mention process by training the mention recognizer on a new corpus with gene, gene family and protein domain annotations. It also integrates several advanced text mining techniques, including SimConcept for resolving composite gene names.

2.2.2 Gene Normalization corpora

High quality gene normalization corpora are important for the development of any type of gene normalization system.

The BioCreative I gene normalization corpus [54] and the BioCreative II gene normalization corpus [55] focused on the GN task for yeast, fly, and mouse genes and human genes respectively. Both of these corpora annotate gene mentions found in

abstracts. In contrast, the BioCreative III gene normalization corpus [56] annotates full length articles and is not limited to specific species.

The BioCreative I gene normalization corpus consists 15,000 abstracts for training, 468 abstracts for developing, and 750 abstracts for testing. All these abstracts are annotated in abstract level, not mention level (only a list of database identifiers is given for each abstract, without any location information). No corresponding gene name in the abstracts for the database identifier is provided in this corpus. The BioCreative II gene normalization corpus consists 281 abstracts for training, and 262 abstracts for testing. All these abstracts are also annotated in abstract level, but the corresponding gene names in the abstracts are given for each database identifier. The BioCreative III gene normalization corpus consists 32 fully annotated articles and 500 partially annotated articles for training. For testing, it provides 50 articles as gold standard and 507 articles as silver standard.

2.3 Other Biomedical Named Entity Recognition and Normalization

There has been considerable interest in the detection and normalization of other types of biomedical entities such as diseases, chemical compounds and drugs. The issues tackled and approaches taken in these fields are similar because of the nature of the tasks and some ideas and concepts found in these works may be helpful to adopt for our work on gene mention detection and normalization.

2.3.1 Other Biomedical Named Entity Recognition

tmChem [60] is a chemical named entity recognition system created by combining two Conditional random field (CRF) models in an ensemble. The two models in the system used different tokenization methods, feature sets, CRF implementations, CRF parameters. Lu et al. [61] developed a chemical named entity recognition system based on mixed CRFs with word clustering. Lowe et al. [62] proposed a system for chemical entity recognition based on grammar and dictionary. Their system uses a mixture of expertly curated grammars and dictionaries, as well as dictionaries automatically derived from public resources.

Chowdhury et al. [63] presented a CRF based approach for disease mention recognition. The features they used include disease specific contextual features, orthographic features, general linguistic features, syntactic dependency features and dictionary lookup features. Kaewphan et al. [64] developed a system for disease mention recognition. Their system was based on an existing named entity system, NERsuite, supplemented with UMLS dictionary features.

Other biomedical named entity recognition systems can be found in [65, 66, 67, 68, 69].

2.3.2 Other Biomedical Named Entity Normalization

Leaman et al. [60] paired their chemical named entity recognition system with a dictionary approach for normalization. They used a dictionary of chemical entities and their names that was collected from MeSH and ChEBI. DNorm [70] is a disease normalization system, which uses a linear model to score the similarity between mentions and concept names. DNorm has an interesting approach of learning term variation directly from training data. Kaewphan et al. [64] developed a disease normalization system, which was based on their disease mention system. They combined compositional word vector representations with CRF to map the recognized mentions to the UMLS concepts. Other biomedical named entity normalization works can be found in [71, 72, 73, 74].

2.4 Biomedical Relation Extraction

Biomedical relation extraction is a task to identify relations between entities mentioned in natural language texts. Significant research efforts have been made in various tasks on this topic, e.g., extraction of protein subcellular localization in BioNLP shared task 2009, 2011, and 2013 in the Genia track [75, 76, 77], protein-protein interaction in BioCreative II and II [78, 79]. Rule based and machine learning based methods are two kinds of commonly used methods for biomedical relation extraction.

2.4.1 Rule based Relation Extraction Systems

BioSEM is a rule based system for extraction of protein subcellular localization event, it generates patterns automatically from training data [80]. RLIMS-P is a system specifically designed to extract protein phosphorylation information on protein kinase, substrate and phosphorylation sites from biomedical literature [2]. Nebhi developed a system which uses linguistic information provided by syntactic parsers [81]. miRTex is a text mining system that uses lexico-syntactic rules to extract miRNA-target, miRNA-gene and gene-miRNA regulation relations [82]. Peng et al. [83] developed a system for the detection the protein-protein interaction. The system utilizes extended dependency graph (EDG), an intermediate level of representation that attempts to abstract away syntactic variations in text. Other rule based relation extraction systems can be found in [84, 85, 86, 87, 88].

2.4.2 Machine Learning based Relation Extraction Systems

EVEX uses support vector machine with various lexical and syntactic features to extract the Protein subcellular localization event [89]. Liu et al. developed a system for biomedical event extraction. The system is based on a pairwise model that transforms the problem of trigger classification to a multi-label problem [90]. TEES 2.1 is a SVM based system for the extraction of events and relations. It uses an automated annotation scheme, which derives task-specific event rules and constraints from the training data, and uses these to automatically adapt the system for new corpora [91]. Miwa et al. [92] propose a method to combine kernels based on several syntactic parsers for the extraction of PPI. Other machine learning based relation extraction systems can be found in [93, 94, 95, 96, 97].

2.5 Gene Ontology Annotation

The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products across databases. It has developed three structured ontologies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner [18].

Manual Gene Ontology annotation is very time-consuming and labor-intensive [98]. Hence there has been interest in developing automatic systems to assist curators. BioCreative IV [19] defines the Gene Ontology Annotation task as automatically assigning gene functional information using GO terms to relevant genes in the literature.

2.5.1 Gene Ontology Annotation systems

Automatic Gene Ontology Annotation involves three main steps: (1) location of text passages in the literature that can be used as evidence for GO term assignment for a gene (product), (2) gene normalization, and (3) prediction of GO terms for relevant genes. Since gene normalization can be treated as an independent task, the discussion of GO annotation systems can focus on the remaining two tasks.

Zhu et al. [20] presented a system for GO Annotation. They trained a logistic regression model to detect GO evidence sentences, and developed a search-based

approach to predict GO terms based on the GO evidence sentences. Li et al. [21] built a binary classifier to identify evidence sentences, and developed an information retrieval-based method to retrieve the GO term which is most relevant to each evidence sentence. This method was based on a ranking function that combined cosine similarity and the frequency of GO terms in documents, and also used a filtering method based on high-level GO classes. Gaudan et al. [22] introduced a method for automatic identification of GO terms in natural language text. Their work was based on considering the proximity between the words occurring in text and the information content of the GO terms. Couto et al. [23] presented a system called GOAnnotator that automatically identifies evidence text in literature for GO annotation of UniProt/Swiss-Prot proteins. Emadzadeh et al. [24] proposed an approach for finding GO terms for different genes in an article. Their approach was based on distributional semantic similarity over the GO terms. Tuan et al. [25] introduced an approach using the GO cross products as the GO term representation to recognize GO terms in text. Chen et al. [26] proposed a rule based GO evidence sentence retrieval systems based on a set of rule patterns.

2.5.2 Gene Ontology Annotation corpus

The BC4GO corpus [27] is a publicly available corpus for the Gene Ontology Annotation task, it provides GO annotations for each full text article, and evidence sentences for each GO annotation. This corpus contains 200 full-text articles for 1356 distinct GO terms. Among these 200 full-text articles, 100 are used for training, 50 are used for development, and the rest 50 are used for testing.

Chapter 3

GENE NORMALIZATION

3.1 Introduction

A major focus of modern biological research is to link big data to knowledge [99], which requires tools to provide structure to information in unstructured sources such as the scientific literature. One barrier to structured representation of information in the literature is the highly complex nature of the nomenclature of genes and proteins. Multiple names and symbols are frequently used to refer to the same entity, and conversely, a given name or symbol can refer to multiple entities.

In order to comprehensively annotate gene/protein records and to support queries from biologists from a variety of backgrounds who may use different names to refer to a gene/protein of interest, curators of knowledge bases, such as UniProt [1], need to capture the full range of names and symbols by which a protein/gene is known. Automatic detection of gene/protein names in the literature and their linkage to database records, also known as gene normalization (GN), is being developed as an alternative to the current time-consuming practice of manual extraction of names and has become an essential component of many text mining systems and database pipelines. For example, PubTator, which uses GNormPlus, incorporates GN to assist curation of genes in PubMed abstracts. Our group also uses GN of kinase and substrate mentions to integrate phosphorylation information from the text mining tool

RLIMS-P [2] into iPTMnet [3], a bioinformatics resource for protein post-translational modifications. GN also enables semantically refined literature searches [100] (e.g. retrieval of literature for a given protein in a particular taxon group), Thus, efficient and reliable GN systems can play a key role in the effort to link big data to knowledge.

Despite large body of work has been done, as shown in the related work chapter, gene normalization continues to be inadequate for certain taxonomic groups, particularly plants, where (i) there is a lack of common standard nomenclature across species [101-106], (ii) locus or ORF species-specific names are frequently used, and (iii) there is a high frequency of ambiguity in gene names because of the high number of paralogs in multigene families. Thus, we first conducted our experiments on plant related literatures. We developed a gene normalization system, called pGenN (pivot-based Gene Normalization) [13]. Although it incorporates notions that are not specific to any species, the system was initially tailored to normalizing plant genes. Additionally, the system was designed to work on abstracts. For evaluation purposes, we had to develop our own curated literature corpora due to the non-availability of annotated plant literature in existing corpora. Evaluation using these data sets, shows that our system achieved state-of-art performance on abstract level plant gene normalization.

We applied pGenN to all plant-related Medline abstracts to test its scalability. The results, which are updated monthly in sync with PubMed, are stored in a local database called pGenN_DB. The pre-processed results are publicly available via a web interface for multiple modes of querying and downloading of results.

Finally, pGenN was extended to full length article. The focus of this extension was on the species assignment process of genes in the results section. We show this component yields good performance.

3.2 Methodology

In this section, I will describe our approaches used in pGenN for plant gene normalization. The overall architecture of pGenN is shown in Figure 3.1. The input text in Medline abstract format is broken up into sentences and then into individual tokens. Potential gene name candidates are first identified using a dictionary lookup. Next, a context-based disambiguation component decides which of these candidates correspond to actual gene mentions. Then, a species is associated with each detected gene mention. Using different features gathered from text and dictionary, pGenN completes the normalization process by assigning an identifier for a species-specific gene in the dictionary among those that matched the name found in the text.

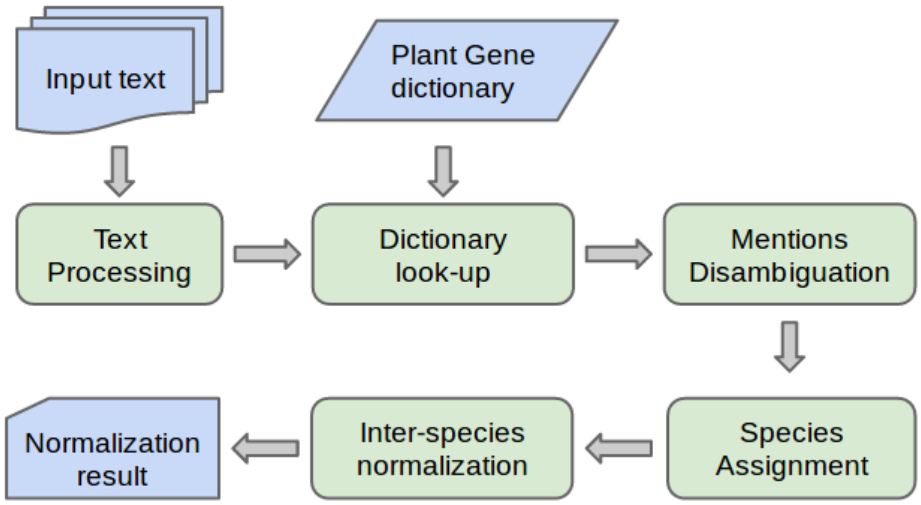

Figure 3.1 pGenN system architecture

3.2.1 Text Preprocessing

Given a list of PMIDs, the titles, abstract text, and the MeSH terms are extracted for each PMID. We use an in-house developed tool to split the abstract text into sentences and then tokenize the sentences. This tokenization is based entirely on orthographic features such as the combination of lowercase followed by uppercase letters or presence of numerals and symbols. For example, terms such as AtAur1 and Mn(2+) are tokenized as “At Aur 1” and “Mn (2 +)” respectively. We then use an in-house developed tool to tag noun phrases in each sentence.

3.2.2 Using Dictionary to Determine Gene Mention Candidates

Many existing gene mention recognizers (e.g., BANNER [33]) do not use a dictionary-based approach. For example, BANNER uses conditional random fields (CRF) [35] with orthographic, morphological and shallow syntax features. In contrast, we use a dictionary-based approach for the gene mention task. Dictionary matching is an essential part in the gene normalization task since the task of gene normalization is to link gene mentions with identifiers included in the dictionary. So even if a GN system uses a system such as BANNER to detect gene mentions, it would still need to match the detected mentions with a gene dictionary for normalization. Additionally, using a dictionary for gene mention detection avoids the problem of false negatives produced by the gene mention systems. Similar points have been observed earlier by others, e.g., Verspoor et al. [107].

We create a plant gene dictionary based on the UniProt database. UniProt protein entries contain protein accessions (ACs), protein names and gene names. Our dictionary is created by downloading all entries for all plant proteins in UniProt (latest update based on the 2017_05 release). Similarly to others who have built gene mention detection or normalization tools, we do not distinguish between genes and their protein products. Hence, we collected both gene and protein names from UniProt for all the plant gene/proteins. Each entry in the dictionary contains additional information, such as its species and synonyms (alternate names), along with the name.

While gene names from UniProt are taken directly, protein names are further processed to extract parts that are likely to be found in text. Consider the protein name “Pre-mRNA-splicing factor SLU7” (UniProt AC A2YQU8). In addition to the original

name, we also include the shorter version “SLU7” in the dictionary as they are often found in the abstract text without the preceding “descriptive” part. In order to split the full name into these two parts, we look for protein names in a specific pattern: “word1 word2 … wordN f-term c-term”. The notions of c-terms and f-terms were introduced in [31] and further developed in [32]. An f-term comes from a small list of words such as gene, protein, factor and enzyme, which indicate that the entity is a gene or its product. Table 3.1 shows the regular expression we used for identifying gene f-terms. A c-term, on the other hand, is characterized by the presence of orthographic features such as capital letters, numerals and special prefix symbols, which indicates that the term is not a typical English word but likely to be a name.

Table 3.1: Regular expression for identifying gene f-terms

/(gene|protein|factor|kinase|[^abehiou]ase|oncogene?|binder|globulin|tubulin|lectin|galectin|globin|tinin|matin|ietin|tropin|zyme|kine|leukin|nogen|receptor|enzyme|hormone|protease|permease|nuclease|oncogene)$/

UniProt contains two types of entries: reviewed (Swiss-Prot) and unreviewed (TrEMBL). The reviewed entries are curated by domain experts with information extracted from literature and curator-evaluated computational analysis and are assumed to be of high quality. Proteins and gene names in the reviewed entries include recommended name and synonyms from the literature and nomenclature standards, plus locus names and open reading frame names (ORFs) for gene names. The unreviewed entries, on the other hand, contain protein sequences (e.g., from translation of sequences in GenBank [108]) associated with computationally generated

annotation and large-scale functional characterization. These entries have not been curated by human annotators. Protein and gene names in unreviewed entries come from direct submissions, propagation of annotation rules developed by UniProt, or other external sources. UniProt coverage for plant proteins in the reviewed section is limited; as an example, there are only 446 entries for tomato proteins in the reviewed part, but 37,386 entries from this species in the unreviewed part. Thus, we also considered including gene and protein names from unreviewed entries in UniProt. A quick analysis of some unreviewed entries suggested that while the gene names were acceptable, the protein names should be used with caution because they are often very general. For example, in the unreviewed entry UniProt AC B6SS10, the Submitted (protein) full name is "Receptor kinase". Using such names would lead to too many non-specific matches. As discussed above, the fact that the name ends with an f-term indicates that it is likely to be a generic name description rather than a specific name. Thus, from the unreviewed entries, we extract all gene names and only those protein names that include c-terms.

To discuss the organization of the names in the dictionary, we first consider how a name like AtAur1 is stored. Every name extracted from UniProt is tokenized into three parts that we call the prefix, the pivot, and the suffix. The prefix represents the species, in this case, 'At' for Arabidopsis thaliana. A number of plant species follow a similar convention to indicate the species (two-letter abbreviation with the first letter uppercase and the second letter lowercase), but it is not universal. Several alternatives can be found in Table 3.2. The suffix includes numbers or Greek alphabet letters (or single uppercase alphabet letters corresponding to common Greek alphabet

letters) that are found at the end of the names. We call the remaining part between the prefix and the suffix the pivot. Thus, in this case, ‘Aur’ is the pivot.

Table 3.2: Plant species prefix conventions

2 letter-prefix	First letter is upper case and second is lower case. e.g., “At” for “Arabidopsis thaliana”, “Os” for “Oryza sativa”.
3 letter-prefix	Only for Brassica species. First letter must be upper case “B”, which is short for “Brassica”. Second and third letters are lower case. e.g., “Bra” for “Brassica rapa”, “Bni” for “Brassica nigra”.
4 letter-prefix	For Latin binomial. The symbol for a binomial consists of the first two letters of the genus, plus the first two letters of the specific epithet. e.g., “PASM” for “Pascopyrum smithii”.
5 letter-prefix	All the letters must be upper case, and the first three letters must be “VIT”. e.g., “VITVI” for “Vitis vinifera”.

The dictionary is organized hierarchically into three layers, as shown in Figure 3.2. At the top are nodes that are labeled with pivots. These pivot nodes have multiple child nodes where each child node corresponds to a different suffix. Each suffix node (together with its parent pivot node) corresponds to a specific gene name and has as children UniProt AC nodes.

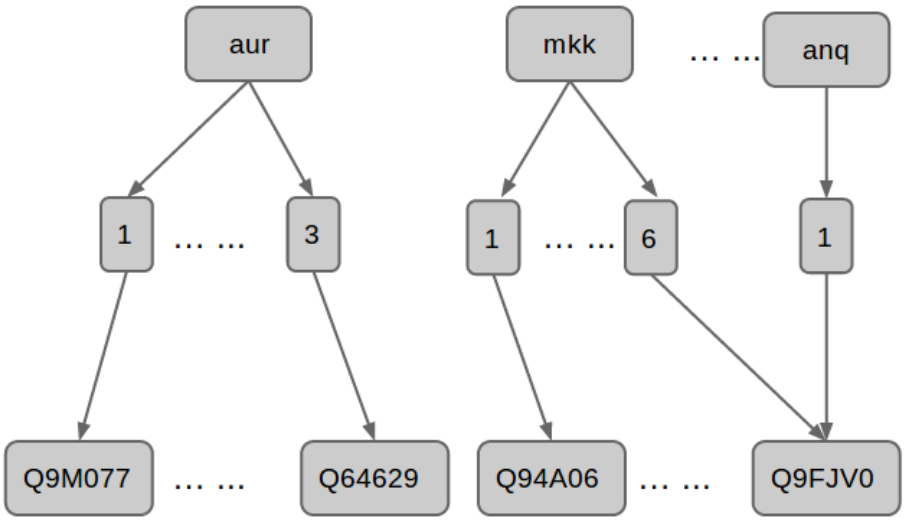

Figure 3.2 Pivot based plant gene dictionary structure

Consider 'AtAur1' appearing in text. It will be tokenized into prefix, pivot, and suffix ('At Aur 1') as stated in the dictionary organization. The occurrence of 'Aur' in text will match a corresponding pivot node in the dictionary. The next token in text ‘1’ matches one of the dictionary entry suffixes under that pivot node. Hence the tokens “Aur 1” can now be considered a candidate gene mention, and several UniProt ID nodes for different species are linked to the suffix node with this candidate gene mention.

Some extensions are added to this simple matching approach to enhance the performance. The first extension is a longest match strategy: if one match occurred inside another, only the longer match will be kept. For example, ‘Bcl-2’ in 'Bcl-2-associated athanogene 7' can be matched to a name in the dictionary, but it will not be kept as candidate gene mention because it is nested in another longer match ('Bcl-2-

associated athanogene 7'). Another extension involves creation of multiple candidates from a sequence of tokens, using three rules. The first rule tackles the case where full name and short name appear together as pivot, and share the same suffix. E.g., “interleukin (IL)-4”. In this case, we create two names from the sequence of tokens: full name plus suffix, and short name plus suffix. E.g., from “interleukin (IL)-4”, we create “interleukin-4” and “IL-4”. Another rule is regarding the case where multiple suffixes appear in conjunction, and share the same pivot. E.g., “Protein kinase C alpha, beta, and gamma”. In this case, we create multiple new names where each name is formed by the pivot and one suffix in the conjunction. E.g., from “Protein kinase C alpha, beta, and gamma”, we create names “Protein kinase C alpha”, “Protein kinase C beta”, and “Protein kinase C gamma”. This rule will also cover cases like “ERK 1/2”, where “ERK 1” and “ERK 2” will be generated. The last rule is similar with the second one, except it covers the case where suffixes appear in range. e.g., from “ERK 1-8”, we create “ERK 1”, “ERK 2”, ..., “ERK 8”.

3.2.3 Using Context to Determine Gene Mentions

Dictionary look-up alone is not sufficient to detect gene mentions. For example, consider the following sentence from PMID 6857705: “injection of the purified toxins are 91 (SN1) and 71 (SN2) micrograms/kg mouse”. The name candidate “SN1” matches a pivot and suffix in the dictionary, but it is not a gene mention.

In general, a candidate can have a gene sense or a non-gene sense. In this section, we discuss how we decide if each candidate is a gene mention or has a non-gene sense. First, we apply some heuristic rules to decide whether a candidate gene mention is an actual gene mention or not. Next, we apply a support vector machine (SVM) model for predicting gene or non-gene sense. It is customary for sense disambiguation to be handled by considering context in the form of words that appear nearby, and hence the words appearing nearby are used as the features for SVM learning. One of the distinguishing aspects of our sense disambiguation approach is that we use only immediate local context (within two words to the left or right) as features. We believe that such an approach has potential to achieve high precision but can also lead to low recall. To alleviate this problem, we have developed a technique to propagate gene context to other instances of candidate mentions. This technique is assisted by the idea of pivots we have introduced before.

3.2.3.1 Rule-based Disambiguation

The first rule is applicable for a candidate mention that matches a dictionary full name. Since long names are assumed to not correspond to a non-gene concept, these full name mentions are directly considered as gene mentions, and therefore are not subjected to the processes described below.

Second, candidate mentions can sometimes be inferred to be family names or complex names, and hence excluded on that basis. Rules for filtering out these types of mentions are shown in Table 3.3. In Chapter 5, we will introduce a system for

detecting protein complex mentions. However, that system was not available during the development of pGenN. Thus, here we simply use a few rules to filter out complex names. In the future, we plan to integrate pGenN with the protein complex mention recognition system for mention detection.

Table 3.3: Rules for filtering out family and complex names

If NAME appears at the end of a noun phrase, and the noun phrase starts with "a" or "another", then NAME will be considered as family name and filtered out.
If NAME is followed by words "family" or by an f-Term in plural form, then NAME will be considered as family name and filtered out.
If NAME appears at in the end of a noun phrase, and the NAME is preceded by "subunits of", then NAME will be considered as complex name and filtered out.
If NAME is followed by word "complex", then NAME will be considered as complex name and filtered out.

Third, some language structures are used by the author to give additional descriptive information, hence they provide clear clues to disambiguate names as gene or non-gene. For example, in the sentence…flowering by repressing the transcription of FT, a flowering-integrator gene that encodes…', the appositive of name 'FT', 'a flowering-integrator gene …', provides a strong clue that 'FT' is indeed a gene. The pGenN method detects two language structures, acronym and appositive. Acronyms are detected using an in-house build acronym detector, which is based on Stanford acronym detection algorithm [109], and appositives are detected using the text mining system iSimp [110]. Rules that are based on these two language structures, as well as two other rules (Dictionary matched name in relation and Synonym) used for disambiguation purposes are shown in Table 3.4.

Table 3.4: Rules for disambiguation of name as gene or non-gene

Acronym rule: If an acronym pair is detected, and the full name matches with the gene dictionary or ends with an F-term, then the short name will be assigned a gene sense.
Appositive rule: If NAME has an appositive, and the appositive ends with an F-term, then NAME will be assigned a gene sense.
Dictionary matched name in relation rule: If two or more names are matched with the dictionary and they appear together in a conjunction with other candidate mentions, then all the names will be assigned a gene sense.
Synonym rule: If NAME1 and NAME2 are synonyms (matched with the same dictionary entry), and they appear in the same article, then both NAME1 and NAME2 will be assigned a gene sense.

3.2.3.2 SVM-based Disambiguation

We use the presence of nearby words to determine if the candidate mention is an actual gene mention or represents some other type of entity. We restrict the use of context to words appearing immediately to the left or right of a candidate. Such use of immediate context is motivated by an attempt to achieve high precision, since words adjacent to the candidate are likely to be related to candidate, whereas words that are further away may pertain to some other entities and hence not be necessarily related to the candidate.

Support vector machines (SVMs) [38] are supervised learning models and are commonly used for classification. We employed SVM-light [111], an implementation

of the SVMs, using default parameter settings and a linear kernel to learn the disambiguation model.

The six features used for learning correspond to: a single word appearing to the immediate left or immediate right (LI-SW and RI-SW), a single word appearing within two words to the left or right (LW2-SW and RW2-SW), and two words appearing to the immediate left or immediate right (LI-TW and RI-TW). For example, given the sentence "Mutants lacking jasmonate synthesis or response had decreased MYB21 expression and …", the feature attributes associated with the candidate "MYB21" are 'LI-SW: decreased', 'LI-TW: had decreased', 'LW2-SW: decreased', 'LW2-SW: had', 'RI-SW: expression', 'RI-TW: expression and', 'RW2-SW: expression', and 'RW2-SW: and'. Using an automated method discussed in section 3.2.3.4, we developed a large gene mention corpus which covers plant related literature and annotated the named entities as gene or non-gene. We collected all words and bigrams that appeared next to the annotated named entities to form a feature set. To enhance effective learning, we removed the common English words in the feature set using a stop word list, and selected only the features with a frequency greater than 30. Finally, we got a feature set with 1739 features.

3.2.3.3 Propagating Contexts

Despite developing a large training corpus for training the SVM-based disambiguation model, it still may suffer from low recall. Many occurrences of gene names might not have words immediately to the left or right for us to assign a gene

sense with confidence. Consider the occurrence of “rab5” (or “rab7”) in Figure 3.3. Clearly words nearby are not sufficient to clearly assign a gene sense (or a non-gene sense). However, if there is another occurrence of “rab5” (or “rab7) in the same abstract, which had a clear-cut evidence for the disambiguating model to assign a gene sense, then we assume that both the occurrences will have the gene sense. This assumption, that multiples occurrences of the same candidate name within the same abstract have the same sense, has a long history in natural language processing and can be called as the single sense per name per document rule. But, in this abstract (Figure 3.3) there is no other occurrence of “rab5” (or “rab7”) to apply this rule. However, the notion of pivot allows us to generalize the rule to single sense per pivot per document rule. By this rule, the occurrence of rab2 (which shares pivot with rab5 and rab7), which can be clearly disambiguated as having a gene sense by the SVM model on the basis of its context to the left, will allow occurrences of rab5 and rab7 also to have a gene sense. Note that the single sense per pivot per document rule subsumes the single sense per name per document rule for propagating context.

PMID - 2115402

TI - Localization of low molecular weight GTP binding proteins to exocytic and endocytic compartments.

AB - A set of 11 clones encoding putative GTP binding proteins highly homologous to the yeast YPT1/SEC4 gene products have been isolated from an MDCK cell cDNA library.

We localized three of the corresponding proteins in mammalian cells by using affinity-purified antibodies in immunofluorescence and immunoelectron microscopy studies.

One, the MDCK homolog of rab2, is associated with a structure having the characteristics of an intermediate compartment between the endoplasmic reticulum and the Golgi apparatus.

The second, rab5, is located at the cytoplasmic surface of the plasma membrane and on early endosomes, while the third, rab7, is found on late endosomes.

These findings provide evidence that members of the YPT1/SEC4 subfamily of GTP binding proteins are localized to specific exocytic and endocytic subcompartments in mammalian cells.

Figure 3.3 Example for ‘uni-pivot gene sense assumption’

In spite of our efforts to increase the application of context based disambiguation by such propagation, there could still be instances of candidates which do not have context features to assist the disambiguation. Based on our experience of analyzing a significant number of Medline abstracts, these candidates will be assigned a non-gene sense.

3.2.3.4 Developing annotated corpus for GM task

Recall that we need an annotated gene mention corpus for machine learning to determine if the candidate mention is an actual gene mention or represents some other type of entity. The currently available corpora either tend to be organism specific, domain specific, or not large enough to cover a wide range of immediate gene contexts. e.g., the GENIA corpus [52] is focused on a subset of human hematology, the PENNBIOIE [53] corpus is on oncology, the BioCreative 1 GM corpus [28] contains only 15,000 sentences, the BioCreative 2 GM corpus [29] contains only 20,000 sentences (15,000 of which were used previously in BioCreative 1). A machine learning-based disambiguation model trained on those corpora using only immediate contexts as features is quite likely to suffer from low recall. A similar observation has been made by Wermter et al. [112]. Since our goal is focused on plant species, a gene mention training corpus that includes text from the plant literature will be the best choice for our system.

To develop a new large gene mention corpus that covers substantial plant literature would be difficult if it relies solely on expert annotation. Hence, we developed an automatic method to create annotated gene mention corpora. Since our method only requires raw text as input, we can efficiently create very large annotated data sets. In addition, our method can be applied to other settings beyond our particular use here. However, we need to take care to ensure that there is little noise in the data.

We retrieved all the Medline abstracts containing one or more gene short names that appear on a list of ~50,000 gene short names from our plant dictionary. Next, we used an algorithm modified from the Stanford acronym detector [109] to detect the full name-short name pairs in these abstracts and picked the ones which have short names appeared in the dictionary. Note that the appearance of these names in the abstracts does not mean that they refer to genes necessarily. Our task was to identify which amongst those pairs represented genes and which did not. For this purpose, we considered the extracted full name and assigned it a gene sense only if the full name appeared in the dictionary with the short name or it ended with an f-term. On the other hand, if the full name does not end with an f-term nor has any word in common with corresponding full names in the dictionary, then a non-gene sense was assigned. Otherwise, it was left unannotated. As an example, the pair, “ataxia telangiectasia mutated (ATM)”, was left unannotated. The full name does not meet the requirement for gene sense assignment. However, since there is a partial match with a full name (which includes ataxia and telangiectasia) corresponding to “ATM” in the dictionary, it was not assigned non-gene sense either.

These simple rules allowed us to identify instances of names from a gene dictionary that had gene or non-gene sense with high degree of confidence. We manually analyzed a sample of 40 full name-short name pairs where the short name appears in in our plant dictionary and found the rule marked 36 as having gene sense and remaining four were left unannotated. All 36 names marked with gene sense were indeed gene mentions. It turns out that even the four unannotated instances were gene mentions.

Once the mentions were assigned a gene or non-gene sense, all occurrences of the same name within the abstract were also tagged. Finally, all the positive full names and short names were replaced with the string “NAMEP”, and all the negative full names and short names were replaced with “NAMEN”. By applying this method, we obtained a large annotated corpus and used the first 200,000 abstracts with 157,336 gene positive instances and 120,308 gene negative instances as our gene mention corpus.

Although this method might not identify all the gene mentions in text, our key hypothesis is that we can still automatically obtain a sufficiently large training corpus that covers a wide variety of contexts by applying this rule to a very large number of abstracts. Also, since all name mentions are not annotated with gene or non-gene sense, it cannot be used for evaluation of gene mention detection. Additionally, this corpus can only be used for learning the context of gene and non-gene mentions, but

not words appearing within the names since the mentions are replaced by strings "NAMEP" and "NAMEN".

3.2.4 Species Detection and Assignment

A gene name matched in text can be associated in the dictionary with several UniProt records corresponding to different species. So, the next task we considered was the assignment of a species to a matched instance based on the text around it. This involved using a recognizer of species mentions in text and connecting each gene mention with the species mentions that were detected in the document.

SR4GN [57] is a well-known recognizer of species names in text that assigns species to gene mentions and has been adopted by other gene normalization tools such as GenNorm [11]. SR4GN uses a species name dictionary for recognizing species and uses a few rules to assign species to gene mentions. However, we found we could not use it for our purpose. First, while it uses prefixes found in gene/protein names to assign species, the prefixes are limited to those of a handful of species that did not include the range of plant species of interest to us. Second, SR4GN employs additional heuristics such as the presence of words in the document like "cohort" or "ferment" to assign species when none are detected in the document. Again, these heuristics do not appear to extend to plant species. Finally, SR4GN always tries to assign some specific species to every gene mention, whereas our analysis of plant literature suggested that there were several cases where the gene mentioned did not correspond to a specific species but rather had a more generic usage (i.e., species independent). Thus, we

concluded that we needed to develop a method to detect and assign species to gene mentions that extends the rules developed for SR4GN.

The detection of species is handled by two components. The first component performs a dictionary matching, using a species dictionary built from NCBI Taxonomy. Because authors often abbreviate the species name by using the initial letter for the genus (e.g. A for Arabidopsis) followed by a period and then the species name (thaliana). Accordingly, the dictionary was extended by adding this type of names. e.g., new name 'A. thaliana' was generated from 'Arabidopsis thaliana'. The second component identifies prefixes in gene and protein names that conform to plant species prefix conventions (same conventions as shown in Table 3.2), as these could indicate the presence of a species. e.g., 'At' for 'Arabidopsis thaliana', 'Zm' for 'Zea mays', 'Os' for 'Oryza sativa'.

The gene mentions are associated with the species based on the following rules. Note that the rules are ordered based on our confidence in their precision. This ordering is used to determine which rule should be used in case more than one of them applies. If a species is assigned for a particular mention based on some rule numbered x, then no other rule lower in the ordering (i.e., numbered greater than x) can override that assignment.

1. Prefix. If a gene mention has a species prefix, we assign the species based on the prefix. e.g., 'AtAurora1' would be assigned the species 'Arabidopsis thaliana'. This rule is similar to Rule R1a in SR4GN.

2. Same noun phrase. If a gene mention and species are in the same noun phrase, we assign that species to the gene mention. e.g., ‘Arabidopsis TOC1/PRR1 gene’ would be assigned the species ‘Arabidopsis thaliana’. There is no direct analog in SR4GN, but this rule is inspired by Rule R1b. This rule also considers the case where the species appears in a prepositional phrase that is attached to the noun phrase containing the name (as in ‘SEX4 from Arabidopsis’). This is an improvement over SR4GN Rule R1b in cases where multiple species and multiple genes are mentioned in the same sentence. For example, in PMID 23879260, the sentence 'The predicted protein for CpPG1 has 416 amino acids, with a high homology to other pollen PGs, such as P22 from Oenothera organensis (76%) and PGA3 from Arabidopsis thaliana (73%)'. Using Rule R1b, SR4GN would assign PGA3 with species Oenothera organensis whereas our rule will correctly associate PGA3 with Arabidopsis thaliana.

3. Species-free. If a gene mention is in the ith sentence, and the first species mentioned in the article is in jth sentence and i< j, then we assume that the gene mention doesn’t belong to any particular species. SR4GN does not have a rule corresponding to this; SR4GN will always assign the gene mention with a species as long as any species is found in the article.

4. Single species. If only one species is mentioned in the abstract, then all the gene mentions in this abstract will be connected with this species. This is similar to SR4GN Rule R1c.

5. Species consistency. If a gene mention with name ‘NAME1’ has been assigned a species in one of the previous sentences, then this occurrence of ‘NAME1’ will be assigned the same species used for the closest occurrence. This rule captures the intuition that authors typically do not switch species without some explicit notification. There is no corresponding rule in SR4GN.

6. Species in the same sentence. A gene mention is assigned to a species that occurs in the same sentence. If there are multiple species in the same sentence, pick the one to the left. This is the same as SR4GN Rule R1b.

7. Species in the previous sentence. A gene mention is assigned to a species that occurs in the previous sentence. If there are multiple species in the previous sentence, then this rule is not applied. There is no corresponding rule in SR4GN.

8. Major species. Species in the title and in the MeSH terms are considered to be the major species of the article. A gene mention is assigned to the major species. If there is more than one major species, then this rule is not applied. This is similar to the idea of focus species in SR4GN, except that we have introduced several other rules, which have no corresponding rules in SR4GN, prior to the application of this rule.

The rules described above were reported in the pGenN paper [13]. We have added one additional rule based on the error analysis in that paper, to decide when a species is not associated with a gene mention, even though they occur close to each other:

9. If term “homolog of”, “ortholog of”, “homology to”, or “homologous to” appear between a gene mention and a species, then the species is not associated with the gene. For example, in sentence “WSL4 is predicted to encode a KCS, a homolog of Arabidopsis CER6.” from PMID 27913740, “Arabidopsis” will not be used for gene mention “WSL4”.

3.2.5 Intra-species Normalization

Once we have detected gene mentions and we have assigned a species to each gene mention, we then use the dictionary to obtain a list of candidate identifiers (UniProt ACs) for that name and species pair. To complete the gene normalization task, we need to choose one of the candidate identifiers. We use the context of the gene mention and information about the identifiers obtained from gene/protein resources to make the choice.

Similar to previous cases, we have a number of rules that are applied in order, where the order of the rules is based on our belief in their accuracy.

The first type of context we considered is in the form of the acronym, appositive, or relative clause attached to the gene mention, or words appearing in the same noun phrase containing the gene mention. This type of context is typically used to introduce descriptions relevant to the entity. E.g., in sentence “SEN1 is a senescence-associated gene in the Arabidopsis which is strongly induced by …” from

PMID 15692183, “senescence-associated gene” is used to describe the gene mention “SEN1”. Hence, we believe words in this type of context are the strongest clues for disambiguation. Words in this type of context are compared to the words associated with each candidate identifier in the dictionary, and the identifier with the most word matches is chosen as the normalization result. Back to the previous example, using the context words “senescence-associated gene”, we can normalize the gene mention “SEN1” to UniProt AC A8MRI9, which has words “Senescence-associated protein” in the full name. However, the context discussed above might not always exist for all gene mentions. If we cannot disambiguate (i.e., identify a single identifier) based on this type of context, we will consider words in the same sentence as context.

To ensure that all the occurrences of the same name were assigned with the same ID, and to address the potential low recall due to only use immediate context, we treated all occurrences of the same gene as a single instance where we combined the context for all the occurrences of the same name.

We generalized this context sharing strategy to names with the same pivot: all the names with the same pivot would share the same context, because we believe the same abbreviation in one article will have the same expansion. We have never observed any case where one abbreviation had multiple expansions in one abstract. However, to make our rule more robust, if we detected different expansions for the same abbreviation, then this context sharing strategy would not be used.

3.3 Evaluation

We have conducted two evaluations: the first is the evaluation reported in the pGenN paper, and the next is an evaluation of the new version of pGenN, which incorporates some minor modifications based on the error analysis of the first evaluation.

3.3.1 Evaluation Setup

Since we are not aware of any existing corpus annotated for plant gene normalization, we developed both evaluation corpora in-house.

The abstracts used in the first evaluation corpus were identified by searching PubMed using the query: ("Proteins"[MeSH] OR "Genes"[MeSH]) AND "Viridiplantae"[MeSH]. One hundred and four abstracts were selected from the retrieved abstracts, with a selection process that attempted to ensure coverage of a range of different gene names, different species, and different publication years. The annotation was completed by a senior bio-curator who did not participate in the development of the system. Altogether, 195 instances of UniProt AC-PMID pairs were annotated from the 104 abstracts.

To build the second evaluation corpus, same query: ("Proteins"[MeSH] OR "Genes"[MeSH]) AND "Viridiplantae"[MeSH] were used to search in PubMed for a set of plant related abstracts. 100 abstracts were selected from the retrieved abstracts, with a selection process that attempted to ensure coverage of 8 common plant species: Arabidopsis thaliana, soybean, tobacco, tomato, potato, spinach, wheat and maize. The annotation was completed by 5 senior biocurators who did not participate in the development of the system. Altogether 212 instances of UniProt AC-PMID pairs were

annotated from these 100 abstracts. Both evaluation corpora are publicly available at http://research.bioinformatics.udel.edu/iprolink/corpora.php.

As most of the existing gene normalization tools were designed for non-plant species and hence are not appropriate for comparison. Some of these tools are also not publicly accessible. Thus, in the first evaluation, we were able to compare with GenNorm [11] only. In the second evaluation, we compare our results with GNormPlus [12], an updated version of GenNorm.

In both evaluation, the system performances were computed using the standard measures of precision, recall and F-measure. Gene/protein mentions linked to accession numbers of non-plant genes were not considered. pGenN may return multiple UniProt ACs for one gene mention, due to: (1). the UniProt entries are almost identical except for the subspecies designation. e.g., Oryza sativa subsp. indica (Rice), or Oryza sativa subsp. japonica (Rice). (2). the UniProt entries are redundant, corresponding to reviewed and unreviewed entries for the same entity. Errors that originated from either of these two reasons were ignored. GenNorm and GNormPlus normalize genes to EntrezGene identifiers. To compare the performance with GenNorm and GNormPlus, we used the ID mapping tool provided by UniProt to convert these identifiers to UniProt ACs.

3.3.2 Evaluation Results

Table 3.5 shows the precision, recall and F-measure for both pGenN and GenNorm on the first evaluation corpus We analyzed the false positive and false negatives of pGenN to learn how different components of pGenN contributed to the errors.

Table 3.5: Performance of pGenN & GenNorm on the first evaluation corpus

System	**Precision**	**Recall**	**F-measure**
pGenN	90.9%	87.2%	88.9%
GenNorm	57.6%	39.0%	46.5%

The within-species normalization component worked surprisingly well, considering we use limited features (based only on names in the dictionary) for disambiguation. This component contributed to error in only one instance, resulting in both a FP and a FN. Many of the errors were due to incorrect assignment of species. This situation is consistent with an observation in Wei et al. [11] that accuracy of species assignment is critical for overall performance on the gene normalization task. When the species is incorrectly assigned, clearly a wrong accession number will be assigned. This not only results in false positives but false negatives as well. An example of an incorrect assignment of species comes from PMID 11197326: “OsMADS14 and -15 are highly homologous to the maize MADS box gene ZAP1 which is an orthologue of the floral homeotic gene APETALA1 (AP1).” Based on the same sentence rule, the closest species mention “maize” is assigned to “AP1” incorrectly. Other species assignment errors were also similar and involved mentions of homologs. The second source of errors is due to the dictionary based gene mention detection. For example, in PMID 16455357, the mention “Ljcen1” did not match with the correct dictionary entry because we failed to detect the species prefix “Lj”, which is short for the species “Lotus japonicus”. This was due to the fact that the gene name cen1 did not start with an uppercase letter as expected based on other plant species naming conventions. Some of the errors were due to failure to capture all variations of a gene name in the dictionary. In PMID 17114582, the text includes a mention of "MtHAP2-1". However, the name in UniProt is "HAP2.1" (UniProt AC A4ZVU9), and we had not accounted for this variation.

The precision, recall and F-measure for both pGenN and GNormPlus on the second corpus can be found in Table 3.6. Results show that pGenN achieves higher precision and recall. The species assignment component showed significant improvements compared to our previous pGenN version. An error analysis revealed that the majority of the errors were due to gene mention detection issues (19 out of the 26 FNs and 10 out of the 15 FPs), rather than normalization itself. One source of confusion was in the disambiguation between gene names and gene family names. For example, in PMID 26508775, "TaPR1" is recognized as a gene mention, whereas it is a gene family name mention. Finally, only 2 FPs and 2 FNs were due to mistakes by the intra-species normalization component.

Table 3.6: Performance of pGenN & GNormPlus on the second evaluation corpus

System	**Precision**	**Recall**	**F-measure**
pGenN	92%	88%	90%
GNormPlus	86%	47%	61%

3.4 Use Case Study: Normalization of Genes Related to Phosphorylation in the Brassinosteroid Signaling Pathway

Text mining tools that extract gene/protein-based information from text can be run on a large scale and the information gathered can be stored formats amenable to searching or incorporation into curation pipelines. However, it is even more useful if the gene/proteins can be normalized to unique database (e.g., UniProt) identifiers. As a use case, we applied pGenN to plant-related phosphorylation information obtained through large-scale text-mining using RLIMS-P [2]. RLIMS-P, which extracts information about kinase, substrate, and site from text, has been run on the entire set of Medline abstracts.

For determining the effectiveness of plant gene normalization in conjunction with this tool, we selected a set of 87 abstracts using the query “brassinosteroid signaling” in which RLIMS-P extracted at least one kinase or substrate. Brassinosteroids are a class of plant hormones that regulate gene expression via a signaling cascade that involves multiple phosphorylation events [113]. The quality of gene normalization by pGenN was compared with a manual annotation of kinase and substrate occurrences in these abstracts. We found pGenN achieved 97.9% precision and 93.5% recall. Perhaps the biased nature of the abstracts and the limited number of proteins may explain the better results than those obtained for the evaluation data sets (GenNorm also achieved better performance on these 87 abstracts, with 93.5% precision and 66.0% recall).

We have incorporated pGenN to normalize kinase and substrate mentions in all plant-related abstracts from which RLIMS-P has extracted a mention of phosphorylation events with a kinase and/or phosphorylated substrate. The results of the entire plant-based text-mined phosphorylation results are accessible via iPTMNet [3].

3.5 Medline Abstracts Processing and Interactive Web Interface

To verify the scalability of pGenN and to develop a large body of pGenN results that we could make accessible to the community, we processed all plant-related abstracts that we identified in Medline using the broad query, ‘plant[MeSH] AND hasabstract[Text]’. The 527,481 abstracts which were returned by PubMed for this query (accessed on May 2017) were processed by pGenN and the results were stored

in a local database which we call pGenN_DB. We intend to update pGenN every three months in sync with PubMed.

Table 3.7 shows the statistics of the processing of 527,481 plant-related Medline abstracts that were obtained by the PubMed query "plants[MeSH] AND hasabstract[Text]". 117,443 PMID-UniProt AC pairs were obtained from 75,125 abstracts.

Table 3.7: Statistics of pGenN large-scale processing of plant Medline abstracts

# of abstracts processed	527,481
# of abstracts which are pGenN positive	75,125
# of unique UniProt ACs obtained	28,445
# of PMID-UniProt AC pairs obtained	117,443

A web interface (proteininformationresource.org/pgenn) has been developed to enable community access to the plant gene normalization results in pGenN_DB.

3.6 Extending pGenN to Full Length Article

So far, we have described pGenN on normalizing gene mentions that appear in abstracts. In this section, we will extend pGenN to full length articles. Based on our informal study, we found normalizing genes in the results section is different from normalizing genes in the abstract and other sections. We also found that many techniques that have been introduced in abstract level gene normalization are still directly applicable to full length article. For these reasons, in extending pGenN to full length articles, we will focus on developing methods specific for the results section. We note that the findings of the experiments found in the research articles should be

expected to be described in this section. Finally, we noticed that the major modifications are required to the species assignment process only. Thus, our extensions are concerned with results section and species assignment.

3.6.1 Observations

In order to investigate how pGenN, more specifically, the species assignment process, can be extended to work on the results section of full length articles, we studied the annotations in the 32 full length articles from BioCreative III training set, and drew the following observations:

(1) The changes need to take into account that different sections of the article have different roles. For example, the setups of the experimental study are usually described in the methods section, while the results of the experimental studies are usually described in results section. Thus, the relevant species information can often be found in the methods section and mention of the species need not be found in the results section at all. On the other hand, the species mentioned in the background section may or may not be relevant for normalization of genes mentioned in the results section.

(2) Many articles describe one or more experimental studies that are focused on genes from a single species, even when genes from multiple species are mentioned in the article. (Out of the 32 articles we have studied, only 4 articles (PMC 2048754, 2443158, 2579434, and 2631505) conducted their research on genes from more than a single species). We observed that the genes used in the experiments and their species are identifiable from the methods section and that such species (if there is a single one) can be used for species assignment in the results section. However, if a particular gene's mentions correspond to different species, we noticed that authors usually

provide that information in immediate context. For example, in PMC 2396500, the experimental studies are focused on genes from Arabidopsis thaliana. When genes from other species are used for comparison purpose, mentions such as “human DDB2” and “DDB1 in mammalian” were used.

(3) For the detection of species from the methods section, we observed that such species are often mentioned in the titles of the methods subsections, or in the beginning of the methods section or subsections, where authors introduce how they conduct their research. Thus, the position (e.g., title of method subsections) and the identification of textual patterns involving species in the first 1-2 sentences of the method subsections can be used to identify the species.

3.6.2 Methodology

Based on these three observations, we developed the following species assignment rules for genes that are mentioned in the results section.

We first detect the species of the genes used in the experiments. These species are detected from the Methods sections and will be called the experimental species. We identify the experimental species if they appear in: (1) the titles of the methods subsections, (2) the first two sentences of the methods subsections and adjacent to phrases such as “sample”, “derived from”, “carried out”, “harvesting” while extracting the species names. For example, in PMC 2423616, sentence which describes species from which the experimental genes are got is “MEFs were prepared by harvesting embryonic stage 14.5 mice”. As discussed in our creation of species dictionary, we also detect species name from presence of cell line names. If multiple species are detected in this manner from all Methods subsections, we see if specific genes can be

associated with them. Techniques used in conjunction with specific genes can also be associated with species but currently we have not implemented this aspect.

We will now consider how species will be assigned to gene mentions in the results sections. The most straightforward case is when only one experimental species is detected. Like the case of major species in our species assignment process for abstracts, this experimental species will be used as the default choice for genes in the result section. It can be overridden by immediate species context, i.e., in cases where the gene mention includes (1) species prefix, (2) species in the same noun phrase or (3) has an attached prepositional phrase with the species name.

If multiple experimental species or no experimental species is detected, then we hypothesize that the species for a gene mention will be explicitly stated in the results section. Thus, this situation becomes similar to an abstract which is read before the methods. Therefore, in this case, each subsection in the result section will be treated like a abstract and the rules will be applied the same way they were designed for the abstracts (as described in section 3.2.4).

3.6.3 Accuracy of the approach

The BioCreative III gene normalization corpus consists of a test set that includes 50 articles which are annotated manually. The annotations are in the form of Gene ID and PMCID pairs. Thus, if there are more than one Gene IDs corresponding to one unique gene name, there is no information that indicates the species for each of the gene mentions. For this reason, we do not have species information for the gene mentions that specifically appear in the results section. Hence, we run our system on 20 articles which are randomly selected from the BioCreative III gene normalization gold standards test set and manually analyze the system output for species assignment

errors in the results section (errors caused by other process, e.g., gene mention recognition, are not included). Table 3.8 shows the number of TPs, FPs and FNs.

Table 3.8 Accuracy of the species assignment process

# of TPs	**# of FPs**	**# of FNs**
106	13	9

We can see the species assignment process contributes to very few errors. These errors mainly correspond to cases where no experiment species is detected and no species information from immediate context can be used. In these cases, some low confidence rules, e.g., species in the same sentence, will be applied. This causes both FPs and FNs.

3.7 Conclusion

We have described a gene normalization system, pGenN, initially designed to normalize plant genes that appears in the abstracts. When developing pGenN, we introduced a new concept called pivot. We believe this concept can be used for other named entity recognition and normalization tasks as well. We also developed a method to automatically generate a large gene mention corpus. This method can be generalized for other types of entities and other specific sub-domains, as well as for distant supervision for development of NER tools. The evaluation shows that pGenN achieves an F-value of 88.9% and 90% on two in-house annotated plant gene normalization corpora, significantly outperforming existing state of the art gene normalization system. Based on the case study we conducted, we believe that pGenN can be integrated into text mining pipelines. In fact, pGenN has already been integrated into iPTMNet, a resource for protein post-translational modification that draws, in part, on information gathered by text mining tools. Additionally, pGenN has been used to process a comprehensive set of over 527,481 plant related Medline

abstracts. The pre-processed results have been stored in our local database, pGenN_DB, and. can be searched, sorted and downloaded via a web interface, found at http://biotm.cis.udel.edu/gn/. We also extended pGenN to full length article. The focus of this extension was on the assigning species to gene mentions found in the results sections of the articles. In this work, we treated different sections of full length article differently. Same idea can be explored for other text mining tasks that are applied on full length articles. Our evaluation shows that this component yields good performance.

In the future, we would like to investigate the usage of pivot based dictionary matching to enhance two aspects of curation of the Protein Ontology (PRO) [16]: (1) Detecting gene family names to enhance coverage of protein family classes and (2) Normalizing family protein names to PRO IDs when terms already exist in the ontology.

Chapter 4

EXTENDING COMPUTATIONALLY MAPPED BIBLIOGRAPHY FOR UNIPROT KNOWLEDGEBASE

UniProt Knowledgebase (UniProtKB) is a publicly available database with access to vast amount of protein sequence and functional information. To widen the scope of the publications associated with a protein entry, UniProt has introduced the computationally mapped additional bibliography section in each protein entry to include relevant literature collected from external sources. This effort has focused on adding literature mainly to entries from model organisms, while the entries for other species remain under annotated. To alleviate this situation, especially for plant protein entries, we have developed a text mining system, eGenPub, which selects articles for automatic inclusion of additional bibliography for given UniProt protein entries.

4.1 Introduction

UniProt is a comprehensive, high-quality and freely accessible resource of protein sequence and functional information [1]. The Knowledgebase (UniProtKB) is the central database containing information for over 85 million protein sequences (as of Release 2017_05). UniProtKB consists of two sections, one known as

UniProtKB/Swiss-Prot that is annotated by experts, and the other, UniProtKB/TrEMBL that is automatically annotated. Expert curation in UniProtKB/Swiss-Prot includes manual verification of each protein sequence as well as a critical review of experimental data from the literature and predicted data from a range of sequence analysis tools [114]. A recent article describes in detail the basis for selection of relevant articles for expert curation [115]. Rather than curating all articles about a given protein, UniProt chooses a representative set that maximizes the information content. Thus, an article related to the entry with potential useful information content may not have been included in the entry because: its information overlaps (is redundant) with existing annotations, its main theme is out of scope for UniProt curation, or it is a new publication that has not yet been reviewed by curators. Moreover, UniProt focuses the curation effort on the most widely studied species, therefore some organisms for which experimental data may be available are not actively annotated.

To complement the curated literature set in UniProtKB/Swiss-Prot with additional publications and to add relevant literature to UniProtKB/TrEMBL entries that have not yet been curated, UniProt compiles additional bibliography from external sources. This additional bibliography consists of literature mapped to UniProt entries from other curated databases (such as, Wormbase [116], Rat Genome Database [117], Intact [118], TAIR [119], GeneRIF [120] and IC4R [121]) added in a collaborative manner. However, this effort provides literature mainly to entries from model organisms, while UniProt entries for other species remain under annotated.

To tackle this issue, the use of text mining tools to systematically associate literature to protein entries, with focus on species not yet covered by the curated resources, can be explored. The detection of protein names in an article by itself is not sufficient for associating the article with the protein entry in UniProt bibliography. For UniProt inclusion, the article is expected to describe at least one experiment conducted on the given protein and in the given species. For example, many articles describe properties of a protein/gene and mention its homologs, e.g., “The function of PsBRC1, the pea (Pisum sativum) homolog of the maize (Zea mays) TEOSINTE BRANCHED1 and the Arabidopsis (Arabidopsis thaliana) BRANCHED1 (AtBRC1) genes, was investigated.” from PMID 22045922. The above article provides functional characterization of the pea BRC1 protein only, then the article should not be linked to the other homologs mentioned for the purpose of UniProt additional bibliography. Similarly, some gene mentions are listed as background information, e.g., as in the case of soybean LOX1 in “It has been known that lipoxygenase (LOX) isozymes exhibit differences in product formation, but most product information to date is for LOX 1 among soybean (Glycine max) LOX isozymes. In this study, LOXs 2 and 3 were purified and used to generate hydroperoxide (HPOD) products in an in vitro system using linoleic acid as a substrate in the presence of either air or O2.” from PMID 15998134. This article does not appear to contain any experimental study on LOX1 and accordingly, it should not be linked to LOX1. Other times, proteins/genes are part of a methodology, like a marker or a reporter gene, e.g., “In this study, the promoter of PtrWRKY89 (ProPtrWRKY89) was isolated and used to drive GUS reporter gene.” from PMID 27019084. This article should not be linked to GUS entry.

A key requirement for such an association is that the article must describe some experimental data about the protein. It is more important to include a correct article than to miss one. The examples above demonstrate the need to couple the normalization of gene/protein mentions with the concept of "aboutness" to ensure that articles are linked to the relevant UniProt entries. In this chapter, we describe a method that utilizes a trained support vector machine (SVM) model to predict whether an article, based on its abstract, is appropriate for linking to some UniProt entry as additional bibliography. We use pGenN, a normalization tool for plant genes and proteins described in Chapter 3, for the detection of gene/protein mentions and association to UniProt entries. By utilizing these two tools, we have developed a system, eGenPub, that adds articles to the computational mapped bibliography section of UniProt entries. It is currently limited to UniProt entries for 8 common plant species: Arabidopsis thaliana (Arabidopsis), Glycine max (soybean), Nicotiana tabacum (tobacco), Solanum lycopersicum (tomato), Solanum tuberosum (potato), Spinacia oleracea (spinach), Triticum aestivum (wheat), and Zea mays (maize).

4.2 Methodology

eGenPub employs a 2-stage process (green boxes in Figure 4.1). Given a UniProt accession (AC) as input, it first uses pGenN output to identify PubMed abstracts that have mentions linked to that UniProt entry. Then, each UniProt AC-PMID pair is converted into a set of features, which are input to the SVM. If a UniProt AC-PMID pair is labeled as relevant by the SVM then it means that eGenPub is predicting that article given by the PMID is about the protein given by the UniProt AC

and therefore, eGenPub predicts that the article can be included in the protein entries bibliography section. We have described pGenN in detail in Chapter 3. In this chapter, we will only introduce the SVM model that suggests when a PMID is an appropriate bibliography entry to add for an UniProt entry.

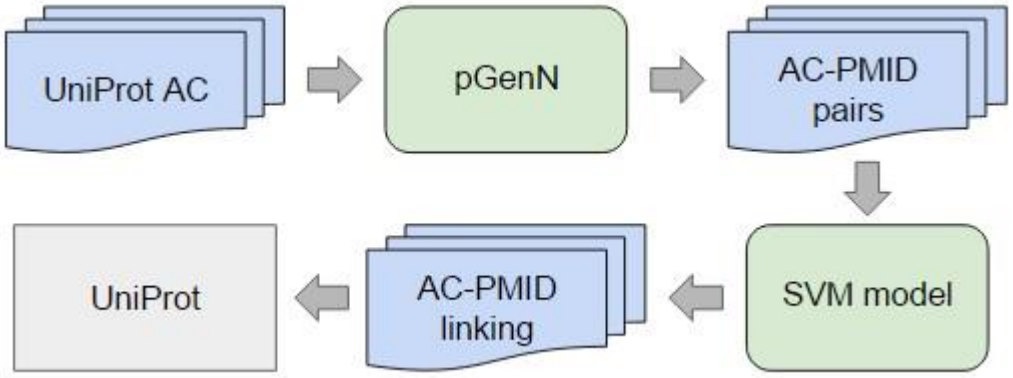

Figure 4.1 eGenPub system architecture

4.2.1 Using SVM for detecting AC-PMID linking

Suppose a gene (normalized to a UniProt AC) occurs in a PubMed abstract (indexed by a PMID). Based on the occurrences of this gene in the abstract, we assign features that are used in the learning of the SVM model. There are a few types of features considered, with additional details described below.

The first type of features is concerned with the frequency of occurrence of the gene in the abstract. Our assumption is that if the gene is mentioned several times in the abstract, it is highly likely that the abstract is “about” the gene. In contrast, if the

gene appears only once in the abstract then it is possible that it was just mentioned in passing and the gene is not central to the study reported in the abstract. We have used two features that consider the number of mentions of the gene: feature **FGE3** counts whether the gene appears at least three times in the abstract and feature **FEQ1** is true/false based on whether the gene is mentioned once/multiple times.

The second set of features is concerned with the location of the mention of the gene in the abstract. Our hypothesis is that if a gene is central to the study reported in an article, the gene is likely to be mentioned in certain prominent positions of the abstract, e.g., the title, the first sentence of the abstract and the last/concluding sentence of the abstract. Thus, for occurrences of the gene in these positions, we have corresponding features: **LocTI** records whether the gene is mentioned in the title and likewise **LocFS** and **LocLS** keep track of whether the gene is mentioned in the first sentence and last sentence, respectively.

Next, we consider features that are based on whether the gene in question is the object of an investigation described in the article. Table 4.1 includes a list of lexemes (a lexeme roughly corresponds to a set of forms taken by that single word -- e.g., the lexeme "detect" includes the words "detect", "detected", "detection", …) that indicate an investigation. While clearly this is not an exhaustive list, it is the current list of words we use to determine if a sentence mentions an investigation. The features that we consider indicate whether a gene appears close to these mentions of investigation. We use the distance (number of tokens) between the "investigation" words and the gene to measure the likelihood that the gene is an object being investigated. Based on

this notion, we introduce two features **InvCl** (whether the gene is within 5 words and hence close to the "investigation" word) and **InvSS** (whether the gene co-occurs with an "investigation" word in the sentence but beyond 5 words of it).

Table 4.1: Lexemes used for "investigation" words

Lexemes			
analysis	characterize	clone	demonstrate
detect	determine	develop	express
investigate	isolate	observe	purify
result	sequence	show	test

Our assumption is that abstracts that have the above features are likely to be relevant. Conversely, we also considered some features that may indicate when the gene in question is unlikely to be the object of study in the article. The first feature focuses on the species information. Suppose we have normalized a gene mention and we use our model to decide if this gene-PMID pair should be considered as positive. Since this gene mention has been normalized, we have already associated a species, say s, to this gene. If this species is not mentioned in the title, but rather some other species is mentioned in the title then we assume it is unlikely the gene-PMID pair is positive. Thus, we consider a feature **SOthT** to indicate that another species name appears in the title that is different from the species of the gene. We also consider a complementary feature **SInT** which is set to true if the species name of the gene in question appears in the title.

Similarly, we considered other features concerning gene names rather than species names. Specifically, we consider a feature **GOthT** that records whether

another gene is mentioned in the title and a feature **GOthGE3** that is set to true if another gene appears 3 or more times in the abstract. These two features could be taken to indicate whether some other gene is object of the reported research. However, we do not preclude the possibility that multiple genes can be studied in the article. We have noticed that when multiple genes are studied in one article they are invariably connected in some way, such as being members of the same gene family. A feature, **GFamM**, is introduced to indicate whether any member of the same gene family as the given gene is also mentioned in the abstract. To decide whether multiple genes are from the same gene family, we apply the notion of pivot, which is introduced in Chapter 3. E.g., gene “CDK1” and gene “CDK2” are treated as belonging to the same gene family since they share the same pivot “CDK” but have different suffixes, i.e., "1" and "2". Thus, if multiple genes share the same pivot but have different suffixes, they are treated as from the same gene family.

Table 4.2 summarizes the type of features used in the SVM models.

Table 4.2: List of feature types considered for SVM models

SVM Feature	**Description**
FGE3	gene is mentioned at least 3 times
FEQ1	gene is mentioned once or multiple times
LocTI	gene is mentioned in title
LocFS	gene is mentioned in first sentence
LocLS	gene is mentioned in last sentence
InvCI	gene co-occurs with an "investigation" word in sentence and within 5 words
InvSS	gene co-occurs with an "investigation" word in sentence but beyond 5 words
SOthT	another species appears in title

SInT	species of the gene appear in title
GOth	another gene is mentioned in title
GOthGE3	another gene is mentioned at least 3 times
GFamM	another gene that belongs to same family is mentioned

We considered three different feature combinations (Table 4.3) as different ways of trade-off between precision and recall. We hypothesize that the feature combinations 1 (Model 1) and 3 (Model 3) (Table 4.3) might result in high precision and high recall, respectively. Feature combination 2 (Model 2) represents our guess as to what might be a good trade-off between the two metrics. We explored these three combinations by training three different models using SVM-light [111], an implementation of the SVMs, using default parameter settings and a polynomial kernel to learn the models.

Table 4.3: Feature combinations applied on the SVM model

Models	**Features**
Model 1	FGE3, LocTI, SOthT
Model 2	FGE3, LocTI, SOthT, LocFS, LocLS, InvCI
Model 3	All 12 features

4.2.2 Evaluation Method

We developed a corpus in-house to evaluate the ability of our SVM model to determine the aboutness, i.e., whether the AC-PMID pair should be linked. 450 AC-PMID pairs were selected and marked as either positive (i.e., the pair should be linked) or not. Altogether, 245 AC-PMID pairs were annotated as positive (for

aboutness) and the remaining 205 pairs were annotated as negative. In our evaluation, we used 10-fold cross validation with standard measures of precision, recall and F-measure. The corpus is publicly available at http://research.bioinformatics.udel.edu/iprolink/corpora.php. There is no alternate system to compare with, since as far as we are aware of, our system is the only one to detect the aboutness of article and gene.

4.2.3 Pipeline for adding additional bibliography to UniProt entries

The aim of eGenPub is to automatically suggest additional bibliography for the UniProt. Every three months, new plant related abstracts are retrieved from PubMed using the query: ("Current date"[Date - Publication] : "Old date"[Date - Publication]) AND plants[MeSH]. We filter out review articles. eGenPub processes these abstracts and suggests the additional bibliography (in the form of UniProt AC-PMID pairs) to the UniProt consortium. Since the process of adding additional bibliography to UniProt entries has now been automated (with some spot checking), we only want to include those PMIDs that we can be most confident about. Since Model 1 obtains the highest precision among the three models (see Results and Discussion section), we use it in this automated process.

4.2.4 Semi-automatic categorization of publications in general annotation topics

To determine the value of the bibliography associated with the entries via eGenPub, we conducted a study where we assigned UniProt entry annotation topics [122] to the suggested publications semi-automatically. For this study, we took a set of

UniProt AC-PMID pairs suggested by eGenPub and process these PMIDs using RLIMS-P [2], a tool for extraction of kinase-substrate phosphorylation events. We then checked if any kinase or phosphorylated proteins detected by RLIMS-P mapped to the linked accession. The abstract was tagged with topic [PTM/Processing] label if the UniProt entry was linked to the substrate, with topic [Function] if it was linked to the kinase, or with both if it was an autophosphorylation event. In addition, all abstracts were tagged by an expert curator for other topics.

4.3 Results and Discussion

4.3.1 Evaluation results of the SVM model

We conducted an evaluation of the ability of the SVM model to correctly predict when a UniProt AC-PMID pair should be linked together. Our evaluation was based on 10-fold cross-validation on a set of 450 pairs. Table 4.4 shows the average precision, recall and F-measure of 10-fold cross validation using Models 1, 2 and 3.

Table 4.4: Results of 10-fold cross validation using feature combination 1, 2 and 3

Models	**Precision**	**Recall**	**F-measure**
Model 1	95.3%	60.9%	74.3%
Model 2	88.5%	67.8%	76.8%
Model 3	83.2%	77.5%	80.3%

As expected, Model 1 achieved high precision and the lowest recall. The low recall can be mostly attributed to the fact none of the features hold for some pairs. For example, in PMID 7846163, “GRF2” (UniProt AC Q01526) is mentioned only once and none of the features of Model 1 are true. However, the abstract contains a sentence

that indicates the gene is subject of the experimental investigation: “Two maize (Zea mays) genes, designated GRF1 and GRF2, have been isolated and characterized”.

The recall increases with the inclusion of more features (a 7% increase for Model 2, and a 17% increase for Model 3). The previous example, involving the mention of “GRF2” (UniProt AC Q01526) in PMID 7846163, becomes a true positive for model 2 and 3 as the feature InvCI captures the information “GRF2” is an object of an investigation described in that article. However, there is a decrease in the precision for Models 2 and 3, dropping by nearly 7 percentage points and 12 percentage points respectively, compared to Model 1.

4.3.2 Statistics of full-scale PubMed processing

As previously mentioned, for the purpose of assigning relevant articles to UniProtKB entries, we give more weight to precision than recall. As a result, in our first implementation of the pipeline, we selected the SVM Model 1 to run the full-scale processing applied to the 8 common plant species.

Table 4.5 shows the number of accession (ACs)-abstracts (PMIDs) pairs suggested by eGenPub for the full-scale processing of PubMed. As expected, a subset of these articles is already cited in reviewed entries (second column). However, this constitutes less than half the pairs suggested by eGenPub. More importantly, eGenPub adds additional bibliography to curated entries in Swiss-Prot (which may be source of updates, similar or complementary information, third column), as well as a significant number of uncurated entries in TrEMBL section.

Table 4.5: Statistics of large-scale processing using SVM Model 1

Species	Number of suggested AC-PMID pairs		Number of suggested AC-PMID pairs mapping to		Number of suggested AC-PMID pairs not in UniProt mapping to	
	Suggested	Already in UniProt	Swiss-Prot	TrEMBL	Swiss-Prot	TrEMBL
Arabidopsis	6662	3017	6322	340	3326	319
Maize	588	186	290	298	205	197
Soybean	149	45	45	104	26	78
Tobacco	361	104	142	219	114	143
Tomato	56	15	51	5	38	3
Wheat	369	130	129	240	86	153
Spinach	455	147	136	319	87	221
Potato	385	129	100	285	61	195
Total	9025	3773	7215	1810	3943	1309

The overlapping set of publications in UniProt with those suggested by pGenN is an indication of the relevant bibliography added by eGenPub. To further show the value of the additional papers suggested, we looked at a random set of unique AC-PMID pairs (193 pairs), and mapped them to the general annotation topics in relation to the associated entries. UniProt publications are categorized into the general annotation topics of the entry, namely, Function, Expression, Subcellular location, PTM/Processing, Structure, Sequence, Pathology & Biotech, Family and domains, and Interaction. From these, 150 mapped to a single topic, 35 to two topics, and 8 to more than two topics. Table 4.6 shows the distribution of annotation topics for the additional bibliography suggestions. The results show that the PMIDs added by eGenPub contains valuable information content related to the entry.

Table 4.6: Distribution of UniProt AC-PMID pairs in annotation topics

Topic	Number of AC-PMID pairs
Function	70
Expression	56
PTM/Processing	43
Pathology & Biotech	26
Subcellular location	17
Interaction	16
Sequence	11
Structure	5
Family and domain	2

The bibliography provided by eGenPub is publicly available in the Publication section of the UniProt entry, under “computationally mapped” section. As an example of the information content added, consider the unreviewed entry B5A4B4, corresponding to gene NAC1 in maize (http://www.uniprot.org/uniprot/B5A4B4, Figure 4.2). As of release 2017_05, there is no expert annotation on this entry (it is unreviewed), and the automatic annotation information is limited (Figure 4.2, 1). The publication section (Figure 4.2, 2) lists the source of publications/submissions available. In this case, there are eight submissions listed with no PMIDs in the UniProt entries, and one article in the computationally mapped section. The article added by eGenPub (shown as source:pGenN, Figure 4.2, 3) provides phosphorylation and functional information for Nac1.

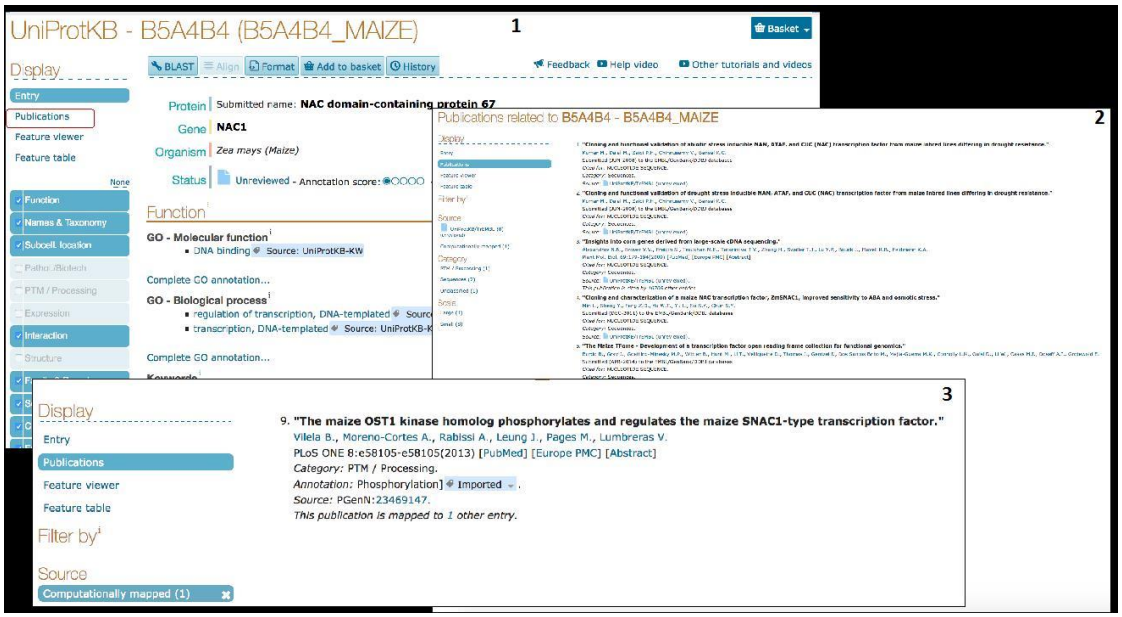

Figure 4.2 Access to UniProt computationally mapped bibliography

4.4 Conclusion

We have presented here a system, eGenPub, to automatically predict whether an article, based on its abstract, is appropriate for additional bibliography for some UniProt entry. The system employs a 2-stage process: (1) using pGenN to detect plant gene/protein mentions in a given abstract and normalize them to UniProt entries, and (2) utilizing a trained SVM model, to predict whether the abstract should be linked to the normalized UniProt entries. We conducted evaluations on 3 trained SVM models which use different feature combinations. Results shows that Model 1 achieved a precision of 95.3%, and a recall of 60.9%, while Models 2 and 3 achieved lower precision but higher recall. A pipeline has been set up to run a full-scale PubMed processing for the selected 8 common plant species using Model 1 to suggest additional bibliography with high confidence. Altogether, 9,025 UniProt AC-PMID pairs have been identified, among which 5,252 (3.943 in UniProtKB/Swiss-Prot

entries and 1,309 in UniProtKB/TrEMBL entries) were not in the existing UniProt publication section. The additional bibliography suggested by eGenPub is integrated in the UniProt production of computationally mapped literature, and can be accessed via the UniProtKB protein entries view of publications.

In the future, we plan to use eGenPub to add additional bibliography for all plant species in UniProt and will provide regular updates in sync with PubMed updates. We will investigate how to improve the recall of eGenPub without significantly affect its precision. Working closely with UniProt, we have demonstrated a robust text mining method for automatically adding bibliography to protein entries in selected plant species, and have shown the added information that these articles bring to the entry. Once integrated into the entries, these additional bibliographies may be used by curators to prioritize and identify entries in need of curation. Because it was not feasible to build a large enough corpus to train the SVM models, in contrast with traditional approach, we manually selected the features in this work. In the future, several additional features can be considered and when necessary automatic feature selection methods can be applied. Other possible future work directions could be: (1) generalizing the detection of aboutness for articles and other named entity, (2) exploring the idea of aboutness in the ranking process of information retrieval.

Chapter 5

TEXT MINING OF PROTEIN COMPLEX RELATED INFORMATION

5.1 Introduction

A protein complex is often defined as a stable set of interacting proteins and where the complex has been shown to exist as an isolated, functional unit in vivo [123]. The interacting proteins, which form the protein complex, are called the components of the complex. Proteins often function as components of larger complexes to perform a specific function, and formation of these complexes may be regulated [14]. Some molecules exist only in certain types of complex (e.g. collagen type I, EBI-2325312) [15]. The need for recognizing protein complex and protein complex components in text stems from their importance in bio-medicine.

Protein Ontology [16], Complex Portal at IntAct [15] and CORUM [17] are some well-known resources which containing information about protein complexes. Entries for individual complex in these resources commonly include the protein complex name and its synonyms, the species and the component proteins of the complex. These types of information are usually obtained from individual experiments published in scientific articles, rather than from the high-throughput experiments data [17].

Currently, the coverage of protein complex in these resources is limited. For example, there are only 374 organism specific protein complex entries in Protein Ontology (04/2017 release), 1905 organism specific protein complex entries in Complex Portal (07/2017 release) and 3408 organism specific protein complex entries in CORUM (03/2017 release). Additionally, they appear to be mainly for a few species. For example, in the case of the Complex Portal, 570 entries are from Homo sapiens (NCBI Taxonomy ID: 9606), 531 from Mus musculus (10090), 424 from Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (559292), with the remaining 380 from 15 other different species. While in the case of CORUM, 2357 entries are from Homo sapiens, 561 from Mus musculus, 373 from Rattus norvegicus, and the remaining 117 from 7 other different species.

In this chapter, we describe our work that is intended to assist in the improvement of the coverage of protein complex in these resources. Although a lot of work has been conducted for recognizing bio-named entities such as genes and diseases, as far as we are aware, there is no system publicly available for protein complex mention recognition. In this work, we will develop a system that detects protein complex mentions and its components. This work consists of three parts: (I) protein complex recognition, (II) protein complex-component relation extraction, and (III) protein complex component-component relation extraction. For example, consider the sentence “Human Fip1 is a subunit of CPSF that binds to U-rich RNA elements and stimulates poly(A) polymerase.” from PMID 14749727, we will first detect the protein complex mention “CPSF” (Task I). Since the annotation of a protein complex contains its components information, detecting the relation between protein

complex “CPSF” and protein “Fip1” will be helpful for complex annotation (Task II. Of course, this information will also help with the annotation of the protein. Also, complex formation can be found without the name of the protein complex. In these cases, we will extract the protein complex component and component relation (Task III). e.g., for sentence “hFip1, CPSF160 and PAP form a ternary complex in vitro” from PMID 14749727, we will extract the component-component relation among proteins “hFip1”, “CPSF160” and “PAP”. Extraction of this relation will assist the annotation for both the protein complex and the protein.

We will introduce our work on Task I protein complex mention recognition in section 5.2. In section 5.3, we will discuss described our work on Tasks II and III. Finally, conclusion will be given in section 5.4.

5.2 Protein Complex Mention Recognition

5.2.1 Text preprocessing

The input for our protein complex mention recognition system can be abstracts or full length articles. The same sentence splitting and tokenization process as described in Chapter 3 is applied. Note that common tokenization method breaks one sentence into either a contiguous block of letters and/or digits or a single punctuation mark. We treat the adjacent words which are c-terms as one token. For example, in the tokenized sentence “Signal transduction by the alpha 6 beta 4 integrin” from PMID 7556090, “alpha 6 beta 4” is treated as one token.

5.2.2 Using CRF

Conditional random field (CRF) [35] is a machine learning technique which is commonly used in the sequence labelling task. As introduced in the related work chapter (Chapter 2), there are many named entity recognition systems, e.g., ABNER [41], BANNER [33], which apply CRF and obtain the best performance in many bio-named entity recognition tasks. We employ CRFsuite [124], an implementation of the CRF, to label the tokens in the text.

It is common to treat the named entity recognition task as a sequence labelling task using the **BIO** model, as introduced in Chapter 2, where each token in the text is labelled as **B**eginning of a named entity, **I**nside a named entity, or **O**utside a named entity. In this work, we also treat the protein complex mention recognition task in this way. For example, in sentence "Endosomal sorting complex required for transport-I is one of three defined protein complexes ...", to recognize the protein complex mention "endosomal sorting complex required for transport-I", we will need to assign label "**B**" to the token with word "Endosomal", assign label "**I**" to tokens with words "sorting", "complex", "required", "for", "transport-I", and assign the rest of the tokens with label "**O**".

5.2.3 Machine Learning Features

Similar to other NER system, such as BANNER, we use both internal features and external features. For the internal features, in addition to the commonly used part

of speech (POS) and lemma features, we also consider the nature of protein complex names, i.e., some protein complex are mentioned via their components, e.g., “Mis16-Mis18” in PMID 24774534. But noted not all the mentions in this format are protein complex mentions, e.g., “Xnr1-Xnr6” in PMID 11934150 mentions a set of genes: “Xnr1”, “Xnr2”, … “Xnr6”. Some NER systems, e.g., BANNER, use features such as whether the current token is a number, or Roman numerals, or the name of the Greek letters. These features are mainly used to decide what can be considered a mention. Since this issue has already been handled in our tokenization method (described in Chapter 3), these features are not considered.

The external features are in the form of the context. Normally, contextual words describing some functions are very helpful to recognize the type of bio-named entity. E.g., in sentence “EMF1 encodes a putative transcriptional regulator, while EMF2 encodes a Polycomb group (PcG) protein.” from PMID 19783648, we can know “EMF1” is a gene based on its context “encodes”. However, since protein complexes and proteins can have the same type of function, their contexts can also be similar. E.g., in sentence “Sgk1 phosphorylates Nedd4-2.” from PMID 18197893, the gene mention “Sgk1” has the context “phosphorylates”. However, complexes such as CK2 can also have this context, as in sentence “We showed that CK2 phosphorylates PU.1” from PMID 16439360.

Therefore, due to the similarity in the descriptions of actions conducted by proteins and protein complexes, we do not learn from context for disambiguation. Instead, we only consider f-terms as used in pGenN. However, of course, we will have

to find an exhaustive set of f-terms for protein complexes. For this purpose, we use complex portal to automatically mine the protein complex f-terms. Entries in Complex Portal contain terms which describe the complex-assembly information, e.g., “heterodimer. These terms are collected and used as protein complex f-terms. Table 5.1 shows the regular expression, which is generated from the collected protein complex f-term list, we used for identifying protein complex f-terms.

Table 5.1. Regular expression for identifying protein complex f-terms

/(complex|dimer|trimer|tramer|hexamer|nonamer|tamer|decamer|octomer|oligomer)$/

Dictionary matching is helpful to recognize the bio-named entities, as was the case with pGenN. However, currently the coverage in the protein complex resources is limited. The value of dictionary lookup feature may be underestimated by a machine-learning method, if the protein complex dictionary is only created based on those resources. Thus, we develop a method to automatically mine protein complex names from Medline abstracts. Details of this mining process is described in section 5.2.3.

Eventually, we decided to use the following set of features. For a current token, represented as token i below, the following features are used:

1. POS feature. The part of speech tags of tokens i-2, i-1, i, i+1, and i+2.
2. Lemma feature. The lemmas for the words of tokens i-2, i-1, i, i+1, and i+2.
3. c-term feature. Whether the word/words of token i contains a c-term.

4. f-term feature1. The protein complex f-term flag (whether it is a f-term) for word of tokens i-2, i-1, i+1, and i+2.

5. f-term feature2. Whether the headword of token i is a protein complex f-term. The headwords here include: (a) the head of the NP which contains token i, (b) the head of the appositive of token i, (c) the head of the relative clause of token i.

6. Multiple gene names feature. Whether the word/words of token i contain multiples gene names. Gene dictionary lookup is applied on the word/words of token i to see whether it contains multiple gene names. The gene dictionary used here is based on the one we used in Chapter 3, with gene names from human, mouse, rat, and yeast added.

7. Dictionary lookup feature. Is token i: (a) the beginning of a dictionary match, (b) inside of a dictionary match, or (c) outside of a dictionary match.

8. Feature combinations. Combinations of feature 3 and 4, as well as feature 3 and 5.

5.2.4 Dictionary Creation

We want to extract protein complex names automatically from Medline abstracts. Recall that we have created a list of protein complex f-terms (described in section 5.2.2). The mining of protein complex names is based on this f-term list, with the following steps:

(1) Search in all Medline abstracts for sentences that contain protein complex f-terms.

(2) Detect the “is_a” relation in the extracted sentences. The detection of the “is_a” relation is handled by one tool is our lab. For each “is_a” relation pair, it outputs two arguments (two NPs).

(3) When one argument ends with a f-term or contains an adjectival form of a f-term (e.g., heterdimeric), then we will extract the other argument as protein complex name if it is a c-term (and hence likely to be a named entity).

Using this method, 3,445 unique protein complex names are extracted. Among them, 1,318 are not in Protein Ontology, Complex Portal, or CORUM.

5.2.5 Post-processing

To improve the performance of the protein complex mention recognition system, we apply several postprocessing rules to the CRF results: (1) If one mention is recognized as a protein complex, then all the mentions in the same abstract/article which have same name will also be tagged as protein complex. E.g., “Arp2/3” in PMID 23354023 sentence 7 is recognized as a protein complex by the CRF model, then all occurrence of “Arp2/3” in this PMID is tagged as protein complex. (2) Abbreviation pairs are extracted using Stanford acronym detection algorithm [109]. Both the full name and the short name will be tagged as a protein complex if either one of them is recognized. (3) Recognized mention which is number or starts with number is removed. E.g., “850-kDa complexes” in PMID 4092036.

We also considered additional postprocessing step to improve the recall. In Section 5.3.2 we discuss the detection of protein complex-component relation using triggers and syntactic dependencies. Some of the triggers and syntactic dependencies can be used as text evidence for recognizing protein complex mentions. For example, if we can detect the structure “X is a subunit of Y”, and we know “Y” is a named entity (by checking whether it contains c-term), then we view “Y” as a protein complex name.

5.2.6 Evaluation

5.2.6.1 Evaluation Setup

Since we are not aware of any annotated corpus available for protein complex mention recognition, we develop our evaluation corpus in-house. The Gene Ontology (GO) is a resource which provides structured ontologies that describe gene products in terms of their associated biological processes, cellular components and molecular functions. Entries in GO contain names of the GO term, and some of these names are associated with PMID. 200 PMIDs are extracted randomly from entries under Protein complex (GO:0043234) sub-hierarchy, and abstracts are retrieved from PubMed using the 200 PMIDs.

As stated in the introduction section, Complex Portal at IntAct and CORUM are two famous resources for protein complex, Entries in these two resources are

associated with PMIDs. Another 100 PMIDs are extracted randomly from these two resources (50 from each).

Protein complex mentions are manually annotated in these 300 abstracts. The annotation is completed by two senior biocurators. Altogether, 745 protein complex mentions (329 unique PMID-protein complex name pairs) are annotated from the first 200 abstracts, and 471 protein complex mentions (191 unique PMID-protein complex name pairs) are annotated from the second 100 abstracts.

We use the first set of 200 abstracts for development, and the second set of 100 abstracts for evaluation. We apply 10-fold cross validation to evaluate the performance of our system. Like the evaluation for other NER tasks, the system performance is computed using the standard measures of precision, recall and F-measure. Since our system has a propagation step (If one mention is recognized as a protein complex, then all the mentions in the same abstract/article which have same name will also be tagged as protein complex), the system performance is computed based on the PMID-protein complex name pairs, instead of mentions. As far as we are aware of, our system is the first system for protein complex recognition. Thus, there is no alternate system to compare with.

5.2.6.2 Evaluation Results

Table 5.2 shows the average precision, recall and F-measure of 10-fold cross validation. We compared the performance of the system using only the CRF model, and the CRF model plus postprocessing steps.

Table 5.2: Results of 10-fold cross validation

	Precision	**Recall**	**F-measure**
CRF	73%	79%	76%
CRF + postprocessing	80%	82%	81%

We analyzed the FNs and the FPs of the CRF model, and found that majority of the FNs are due to the fact that they do not have valid features to be used by the model. For example, in sentence “Ego3 assumes a homodimeric structure similar to …” from PMID 23123112. The context “homodimeric structure” indicates “Ego3” is a protein complex. However, this is not captured by the current used machine learning features. Other FNs are because the system fails to detect the actually name of the protein complex. For example, in sentence “… interaction with an Rps28p/3'UTR mRNP complex.” from PMID 15225542. The actual name of the complex is “Rps28p/3'UTR”. However, the system detects “mRNP” as the name. Of course, this type of errors will also lead to FPs. As expected, the postprocessing steps improve both the precision and the recall. Using of acronym detection and clues based syntactic dependencies contribute more TPs. While the step to filter out invalid names reduce the number of FPs.

5.3 Complex-associated Relation Extraction

5.3.1 Relation Extraction based on Triggers and syntactic dependencies

We use trigger based approach for the relation extraction tasks in this section. A trigger will be a word or multi-word expression that indicates a relation. For example, in sentence “CSC-1 is a subunit of the Aurora B kinase complex”, the relation between the protein complex and its subunit is indicated by word “subunit”. Similarly, the word “dimerizes” can be taken as a trigger for the relation between two protein complex components as indicated in the sentence “Mad1 dimerizes with Max”.

Parsing the text where the trigger and the entities (in this section, the entities considered are protein complex and protein) are mentioned can identify any syntactic dependencies between them. In our relation extraction approach, after detecting the triggers, we use rules based on the dependency edges between the trigger and the entities to see whether the entities fit into the arguments of the trigger to confirm the relation. For the extraction of the component-component relation from the second example sentence, consider Figure 5.1 that shows the syntactic dependencies between the trigger and the two entities that are related.

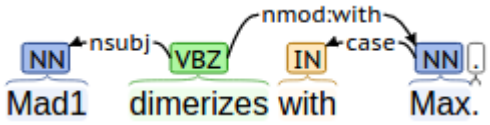

Figure 5.1: Syntactic dependencies for example sentence 2

To confirm the relation between “Mad1” and “Max”, our rule will enforce the following constraints.: (1) the trigger in a verb form appears in the sentence; (2) One of the entities, of type protein (“Mad1”), is the nominal subunit of trigger “dimerizes”; and (3) “Max” is the modifier of trigger “dimerizes”, where the modification relation is given by nmod: with.

We use an existing framework in our lab to develop rules for confirming relations. Given one sentence, the framework uses the BLLIP parser [125, 126] to obtain the parse trees and then apples Stanford Conversion tool to get the Standard Dependency Graph. This framework provides a template to write rules that are in the form of a set of conditions and actions. Back to the second example sentence, after specifying the triggers and the syntactic dependency conditions, an “argComponent” edge is added from the trigger to each protein entity, as shown in Figure 5.2. Details of the template can be found in the Appendix A.

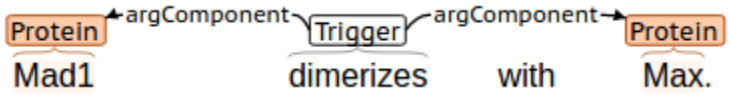

Figure 5.2: Output for example sentence 2

This framework also detects other extra-syntactic information such as is-a and member-collection relations and use them to propagate the original relations to form new relations. For example, Figure 5.3 shows the syntactic dependencies constructed based on the BLLIP parser output for sentence “The yeast eIF3 complex contains five core components: Rpg1, Nip1, Prt1, Tif34, and Tif35.”. Based on the syntactic dependencies, we can see the subject and object attached with “contains” are nodes

"complex" and "components". We just need to write rule to detect the relation between these two nodes. The framework will handle the member-collection structure and add new edges, as shown in Figure 5.4.

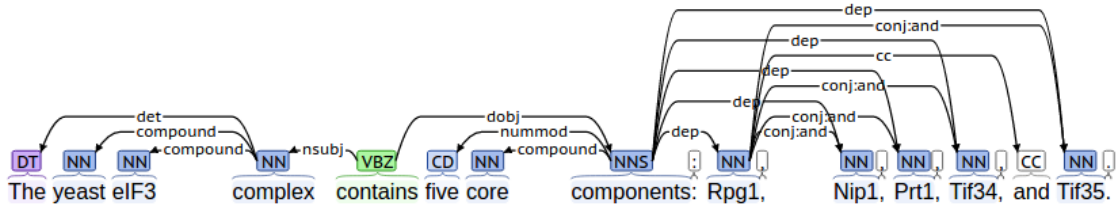

Figure 5.3: Syntactic dependencies for example sentence 3

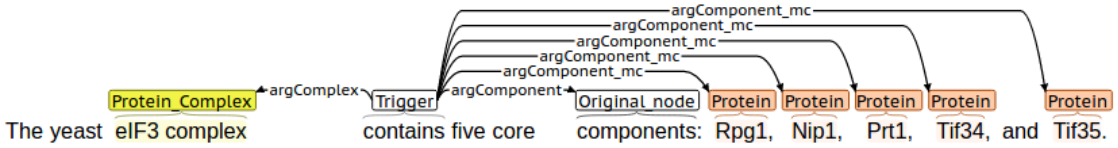

Figure 5.4: Output for example sentence 3

5.3.2 Protein Complex and Component Relation Extraction

Based on our study, we found the protein complex and component relation is often indicated by the "part whole" semantic structure, where a protein mention is found in the "part" argument position and a protein complex is found in the "whole" argument position. Consider sentence "CSC-1 is a subunit of the Aurora B kinase complex.". In this case, the "part whole" structure is indicated by trigger "subunit". However, the "part whole" structure can be indicated by more general English words. E.g., in sentence "MRE complex contains Sp1 or related proteins", the "part whole" structure is indicated by word "contains".

We use three types of rules to extract the part of relation between a protein complex and its component from individual sentences. The first type of rules covers cases like “protein is a subunit of protein complex”, where the triggers are word such as “subunit”, and “component”. In this type of rules, the agent should be the protein, and the argument should be the protein complex. The second type of rules covers cases like “protein complex contains protein”, where the triggers are words such as “contains” and “consists”. In this type of rules, the agent should be the protein complex, and the argument should be the protein. The last type of rules covers cases like “protein is detected in protein complex”, where the triggers can be “detected-in” and “observed-in”. In this type of rules, the agent should be the protein, and the argument should be the protein complex.

By specifying the triggers and the conditions, we add an “argComponent” edge from the trigger to the protein entity, and an “argComplex” edge from the trigger to the protein complex entity if the conditions are met. Figure 5.5 to 5.7 show examples of the output using these three types of rules.

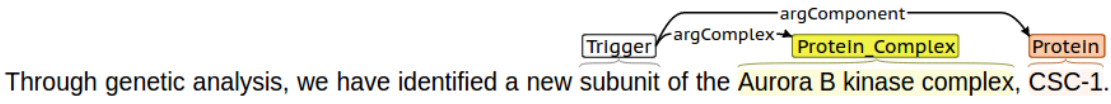

Figure 5.5: Example output using type 1 rules

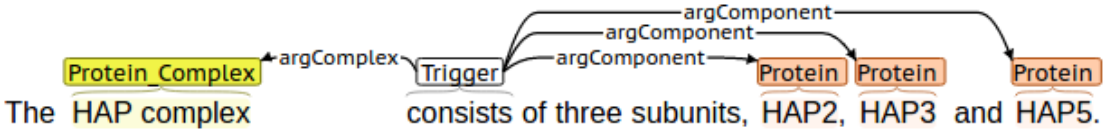

Figure 5.6: Example output using type 2 rules

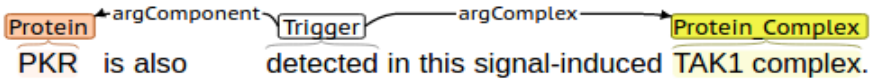

Figure 5.7: Example output using type 3 rules

The range of triggers plus syntactic dependency conditions for the three types of rules can be found in the Appendix B.

5.3.3 Protein Complex Component and Component Relation Extraction

We use two types of rules to extract the protein complex component and component relation from individual sentences: (1) “protein and protein form a complex”, (2) rules based on protein protein interaction (PPI). For the latter, we first use a set of triggers and rules described in [83] to detect PPI relations. Then we use a heuristic that such binding forms a complex (i.e., the interacting partners are in a component-component relation) if the text also indicates that the PPI: (a) is stable, (b) performs some function, or (c) has more than two proteins involved. By specifying the triggers and the conditions, we add “argComponent” edges from the trigger to the protein entities if these conditions are met.

The first type of rules covers the most basic and straightforward cases. Triggers used here are “form” with any protein complex f-term such as “complex” and “trimer”. Noted that the word “form” and protein complex f-term do not need to be adjacent, as shown in Figure 5.8 (in this example, trigger f-term needs to be the object of trigger “form”).

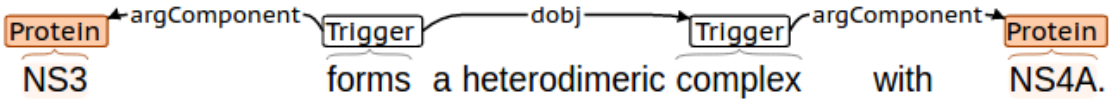

Figure 5.8: Example output1 using type 1 rules

The first type also covers cases like “protein complexes with protein”, as shown in Figure 5.9.

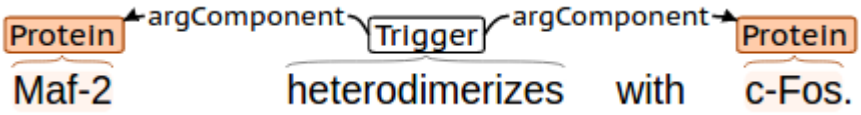

Figure 5.9: Example output2 using type 1 rules

For the first type, we also consider the nominal form of the verb and use words such as formation (of complex) and dimerization as triggers.

The second type is based on the detection of the PPI relation. Triggers used for detecting the PPI relation are words such as “interact”, “bind”, and “associate”. We consider three cases where the PPI indicates protein complex component and component relation. Since a protein complex is often defined as a stable set of interacting proteins, the first case we consider is when the PPI is stable (PPI + stable), as shown in Figure 5.10.

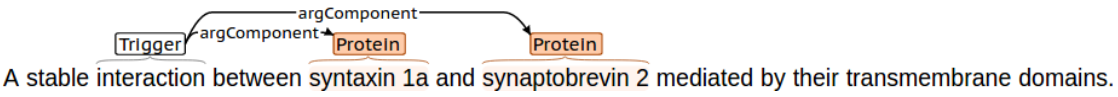

Figure 5.10: Example output for PPI + stable

The second case we consider is when protein interacts with other protein to perform some function, or the interaction itself performs some function (PPI + function). Our hypothesis is that if the interaction results in some function being performed, it is stable enough or lasts long enough to be called a complex. For example, in Figure 5.11, based on the trigger "binds" and the syntactic dependencies between the trigger and the two proteins "CD47" and "TSP-1", we can confirm there is a PPI relation. Since there is a conjunction edge from the PPI trigger "binds" to another verb "inhibits", we can infer that "CD47" binds to "TSP-1" to perform some function, i.e., "inhibits angiogenesis".

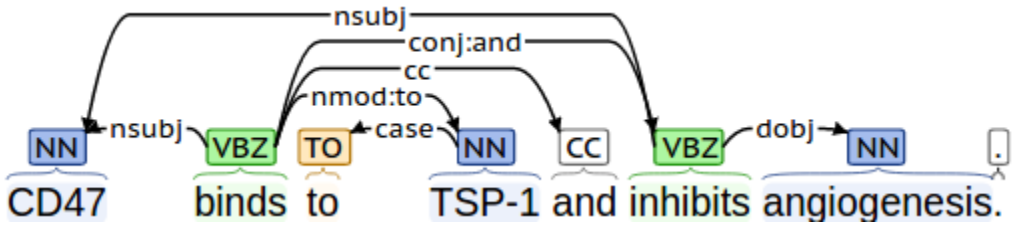

Figure 5.11: Syntactic dependencies for example of PPI + function

Finally, if more than two proteins are found to be involved in one interaction, we treat these proteins as components of one same protein complex (PPI + multiple components). Consider this sentence "Moreover, Bmh1p and Bmh2p associate with Ste20p in vivo.". Since there are more than two proteins involved in the interaction, we can infer that those proteins are components of one same protein complex, as shown in Figure 5.12.

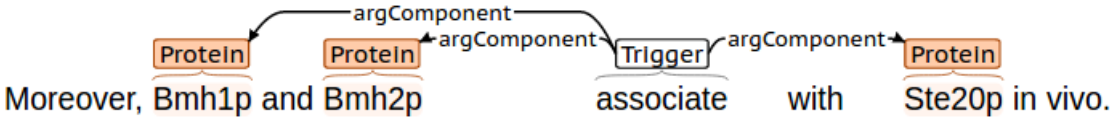

Figure 5.12: Example output for PPI + multiple components

The list of rules (triggers plus syntactic dependency conditions) for detecting the protein complex component and component relation can be found in the Appendix C.

5.3.4 Evaluation Setup

We use the same 300 abstracts described in section 5.2.5 as dataset for development and evaluation purpose. The annotation of the complex-component and component-component relations was completed by two senior biocurators. Altogether, 51 protein complex and component relation pairs, and 76 protein complex component and component relation pairs are annotated from the first 200 abstracts. 77 protein complex and component relation pairs, and 44 protein complex component and component relation pairs are annotated from the second 100 abstracts.

We used the annotations in the first set of 200 abstracts for development, and the second set of 100 abstracts for evaluation. In both relation extraction subtasks, system performances are computed using the standard measures of precision, recall and F-measure.

5.3.5 Results

Table 5.3 shows the precision, recall and F-measure for our complex and component relation extraction approach. The performance of component and component relation extraction is shown in Table 5.4.

Table 5.3: Performance of Protein Complex and Component relation extraction

Precision	**Recall**	**F-measure**
93%	79%	85%

Table 5.4: Performance of Component and Component relation extraction

Precision	**Recall**	**F-measure**
86%	77%	81%

Since we use rules based on the syntactic dependencies, our approaches achieve very high precision. Most of the errors are caused by the parsing errors. For example, Figure 5.13 shows part of the dependencies for sentence “In addition, we have discovered two novel subunits of DASH, Hsk2 and Hsk3 (helper of Ask1), which are microproteins of fewer than 75 amino acids, as dosage suppressors of ask1 mutants.” From PMID 15632076. We can see the edges from trigger “subunits” to the two proteins “Hsk2” and “Hsk3” are both “nmod:of”. While the correct dependency edges should be “appos” (appositive). The remaining errors correspond to cases where there is no clear-cut syntactic dependency between the entities. Thus, our syntactic dependency driven approaches are unlikely to have captured these cases.

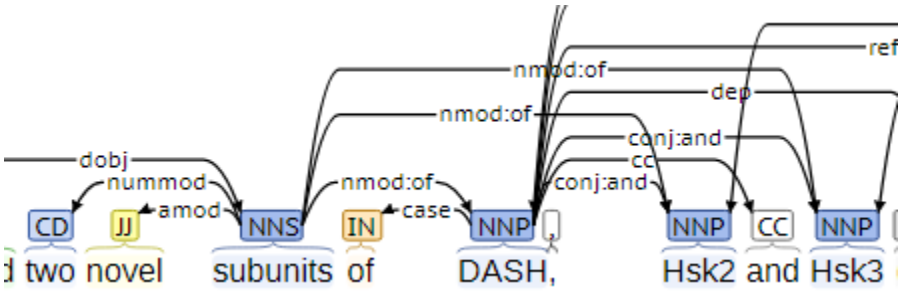

Figure 5.13 Part of syntactic dependencies for example of parsing error

5.4 Conclusion

In this chapter, we have presented our work on three tasks of mining protein complex related information: (I) protein complex mention recognition, (II) protein complex and component relation extraction, and (III) protein complex component and component relation extraction.

Evaluation shows our system for protein complex mention recognition achieves good results. As far as we are aware, there is no system publicly available for protein complex recognition. Our system will be the first one to serve the community. When developing the system, we proposed a method to create a dictionary of protein complex names by extracting protein complex names from the literature using high confidence. Same idea can be used in other situations when a dictionary of names needs to be created. For the two complex related relation extraction tasks, we have shown that our approaches achieves very high precision (with 93% and 86% precision respectively). We believe they can provide high confidence text evidence for the

protein complex resources such as Complex Portal, and assist in the improvement of the coverage in these resources.

Many of the features in the training of the CRF model were manually designed. Again, this is due to the fact that the corpus we were able to obtain for training is not large enough. With a large enough corpus, more data driven approach to feature engineering can be adopted. In the future, we also plan to integrate our gene normalization system with the protein complex recognition system, we will investigate whether this can improve the precision for both systems. We also plan to keep adding syntactic patterns for extracting the two relations to improve the recall.

Chapter 6

GENE ANNOTATION USING GO TERMS FROM CELLULAR COMPONENT DOMAIN

In this chapter, I will describe our work on employing text mining for assisting Cellular Component Gene Ontology (GO) annotation. I will first introduce the task, then the methodology. Finally, I will present the evaluation results.

6.1 Introduction

The Gene Ontology (GO) [18] is a resource that supplies information about gene product function using ontologies to represent biological knowledge. These ontologies cover three domains: (i) Cellular Component (CC), the parts of a cell or its extracellular environment; (ii) Molecular Function (MF), the elemental activities of a gene product at the molecular level, such as binding or catalysis; and (iii) Biological Process (BP), operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units (cells, tissues, organs, and organisms).

GO annotation is a process which assigns gene functional information using GO terms to relevant genes in the literature. It is a common task among the Model Organism Database (MOD) groups. Manual GO annotation relies on human curators assigning gene functional information using GO terms by reading the biomedical literature. Currently manual annotations are made by experienced biocurators from annotation projects including TAIR (http://www.arabidopsis.org/), Saccharomyces Genome Database (http://www.yeastgenome.org/), Mouse Genome Informatics (http://www.informatics.jax.org/), WormBase (http://www.wormbase.org/), PomBase (http://www.pombase.org/), FlyBase (http://flybase.org/), and ZFIN (http://zfin.org/). This process is very time-consuming and labor-intensive. As a result, many MODs can only afford to curate a fraction of relevant articles. For example, the curation team of TAIR has been able to curate less than 30% of newly published articles that contain information about Arabidopsis genes [19].

As mentioned in Chapter 2, many systems have been developed for automatic GO annotation. Despite the fact that there are three types of GO terms (CC, MF, and BP) and the kind of textual evidences for each type are different, current approaches for automatic GO annotation tend to use a single approach for annotations of all types of GO terms. In contrast, our approach is to treat annotation with different types of GO terms as individual subtasks.

In this chapter, we will present our work on annotating genes (as stated in the previous chapters, we will not distinguish gene from protein) using GO terms from the Cellular Component domain. Based on our study, we found that GO terms from this

domain can be divided into two sub-hierarchies: (1) subcellular location terms: i.e., terms in GO under CC category, which are in the sub-hierarchy rooted by 19 GO terms (shown in Table 6.1, all of which have "cellular component" GO:0005575 as parent node). (2) protein complex terms: i.e., terms in GO under CC category, which are in the sub-hierarchy rooted by protein complex (GO:0043234). These two sub-hierarchies cover almost all of the 3757 CC terms in GO. We will treat the task of gene annotation using GO terms from these two sub-hierarchies as two relation extraction tasks: (1) extract cases where a protein is found to be in a subcellular location, and (2) extract cases where a protein is a subunit of a protein complex. For the latter, we will simply apply the methods for the complex associated relation extraction, which are discussed in the chapter 5.

Table 6.1: 19 root GO Terms for Subcellular Location

Extracellular region (GO:0005576)
Cell (GO:0005623nucleoid)
Nucleoid (GO:0009295)
Membrane (GO:0016020)
Virion (GO:0019012)
Cell junction (GO:0030054)
Extracellular matrix (GO:0031012)
Membrane-enclosed lumen (GO:0031974)
Viral occlusion body (GO:0039679)
Organelle (GO:0043226)
Extracellular matrix component (GO:0044420)

Extracellular region part (GO:0044421)

Organelle part (GO:0044422)

Virion part (GO:0044423)

Membrane part (GO:0044425)

Synapse part (GO:0044456)

Cell part (GO:0044464)

Synapse (GO:0045202)

Symplast (GO:0055044)

6.2 Methodology

6.2.1 Gene Annotation concerned with Subcellular Location

Recall this annotation for a protein requires the identification of text that relates a protein with a subcellular location. As in the previous chapter, we are going to use a trigger based approach and employ the same framework involving syntactic dependencies to detect the relations.

The first relation we are interested in is the found-in relation, where triggers such as “found” and “detected” are used. A key aspect of the rules for these triggers is the presence of a locative preposition such as “in”, “at” and “on”. Figure 6.1 shows the syntactic dependencies of the sentence “SIRT2 is found primarily in the cytoplasm.”. Given the word “found” as trigger, we can see “SIRT2” is the nominal subject of the trigger, and “cytoplasm” is the modifier of the trigger (there is an “nmod:in” edge

from “found” to “cytoplasm”). Thus, we can confirm the found-in relation between “SIRT2” and “cytoplasm”.

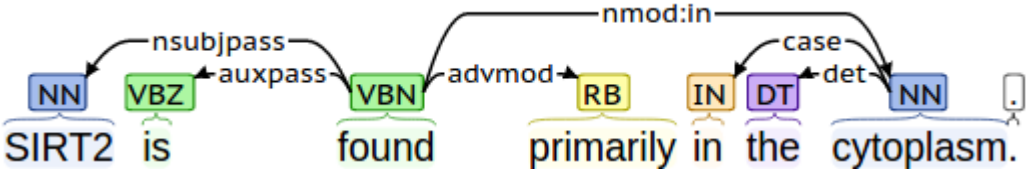

Figure 6.1: Syntactic dependencies for example sentence 1

Note the triggers do not need to appear only in the verbal forms but can also appear in their nominal forms. For example, the trigger “detected” is used in its nominal form “detection” as in the phrase: “the detection of ZmHK1 in endoplasmic reticulum”. The use of the trigger in adjectival form is illustrated in Figure 6.2 Because of the use of the copular structure, the syntactic argument structure is similar to the verb case where the nsubj and nmod:in (or other locative preposition) identifies the protein and the location.

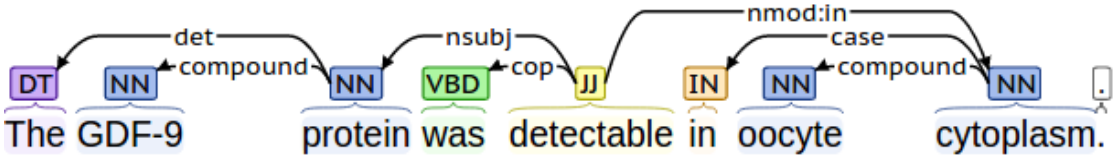

Figure 6.2: Syntactic dependencies for example sentence 3

In addition to the trigger words that correspond to a broadly-defined found-in relation, we also consider “movement” verbs. For these cases, we need to ensure that the moved entity is a protein and the new/old location is indicated by a locative prepositional phrase. An example of this kind can be found in Figure 6.3, Unlike

previous cases, since the verbs used as triggers indicate movement of their objects, these prepositions are likely to be “to” and “from”, rather than “in”, “on”, “at”.

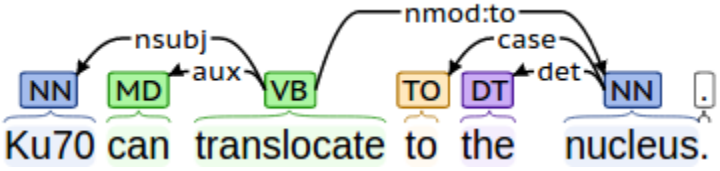

Figure 6.3: Syntactic dependencies for example sentence 4

The next type of relation we consider does not include a trigger that indicate the found-in or movement relation explicitly. Instead, this type of relation covers cases where the proteins are arguments of an event which occurred in some subcellular location and hence the protein-subcellular location relation is implied. For example, from sentence “Fbxo45 interacts with Par-4 in the cytoplasm.”, we can infer that “Fbxo45” and “Par-4” must both be in the cytoplasm since the interaction took place in this region. Figure 6.4 shows the syntactic dependencies of this sentence. From the graph, we can identify the proteins by considering the verb’s syntactic arguments such as “nsubj”, “nmod:with” and “dobj”. The nmod:in edge from the same verb to the location phrase indicates where the event took place.

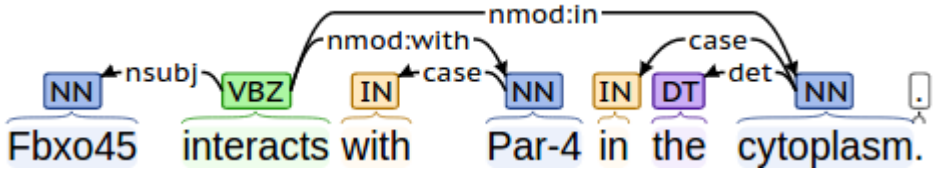

Figure 6.4: Syntactic dependencies for example sentence 5

Note that there are a variety of verb structures that can be used including passives. Also, the event need not be indicated by words in verbal forms but can also appear in their nominal forms. For example, in Figure 6.5, we can know "SGLT1" must be in "plasma membrane" since it is expressed there.

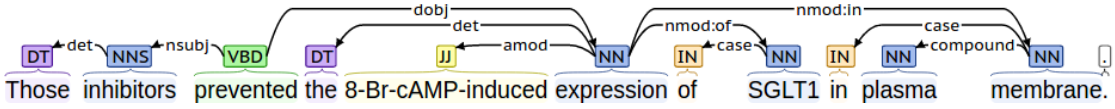

Figure 6.5: Syntactic dependencies for example sentence 6

Finally, if we were only interested in protein-subcellular location relation, then we can consider triggers that are adjectives that are modifiers specifying subcellular locations (e.g., nuclear and cytoplasmic). In such cases, we will look for the protein which these adjectives modify, e.g., in the phrase "the nuclear transcription factors AP-1 or NFIL-2A". However, our interest here is not just in detection textual mentions of a protein's subcellular location. Rather, we are interested in annotation of a protein. Thus, the text must indicate not only the relation but that it is a finding. Therefore, we will not consider cases such as "cytoplasmic CD3" in this work, because we believe it refers to existing knowledge.

The list of rules (triggers plus syntactic dependency conditions) to detect protein-subcellular location relation can be found in the Appendix D.

In addition to the above rules, we also attempted to capture descriptions of experimental results on protein localization, through fluorescent microscopy. Consider the following text: "The localization of CeCDC-14 was analyzed in wild-type C.

elegans embryos, using the affinity-purified anti-CeCDC-14 antibodies for immunofluorescence microscopy. Later in mitosis, during telophase, this staining compacted to a single dot that was positioned between the two daughter cells, highly reminiscent of the midbody.". The word "staining" above indicates protein localization. While we haven't developed concrete rules for these cases, we look for location terms in the same sentence. In this case, we thus obtain the location of "midbody". Next, we attempt to find which protein's staining is detected or observed. For this purpose, we try to see if there is a unique protein (other than tags such as GFP) mentioned in the same or previous sentence. In this case, there is only one protein mentioned, i.e., "CeCDC-14". We use this (as a low confident clue) to infer that protein "CeCDC-14" is in location "midbody".

Finally, as for the detection of subcellular location mentions, we use a dictionary based method, where the subcellular location dictionary is created using terms from GO subcellular location sub-hierarchies (described in the introduction section). In this way, after we detect the relation between a protein and a subcellular location, the protein can be directly associated with a GO term.

6.2.2 Gene Annotation concerned with Protein Complex

In order to annotate a protein with a protein complex GO term, we will first detect the text where a protein is mentioned as a component of a protein complex. This can be done through detecting either of the two types of relations discussed in Chapter 5: (1) Protein Complex and Component Relation, and (2) Protein Complex

Component and Component Relation. Then we will infer the proper GO terms under the protein complex sub-hierarchy based on the text to annotate the protein.

For the first task, we only confirm whether a protein is mentioned as a component of a protein complex. So, to achieve this goal, for the Protein Complex and Component Relation extraction task, we do not need to limit ourselves to when the complex is named. Some general phrases that denote a protein complex such as "a complex" can also be used as valid context. For example, in Figure 6.6, if we know "JAB1" is a subunit of a protein complex, this is enough for us to select this sentence as GO annotation evidence sentence. Thus, in this chapter, we will also detect phrases that denote a complex by checking whether the phrase is ending with protein complex f-terms.

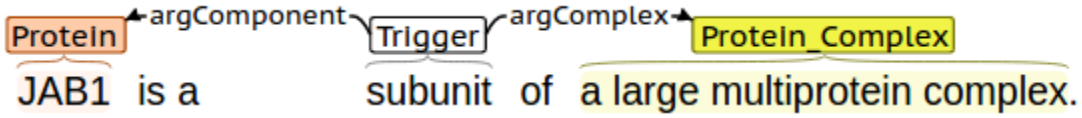

Figure 6.6: Relation extraction output for example sentence 7

Similarly, for the Protein Complex Component and Component Relation extraction task, we do not need to require all the components involved to be to explictly named proteins. Thus, for example, the following sentence (Figure 6.7) suffices to assert that CAR-1 belongs to a complex.

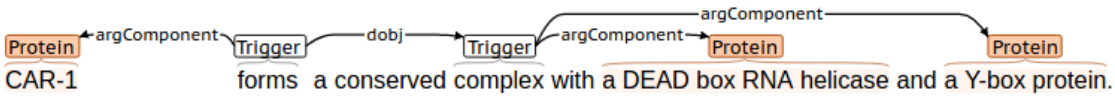

Figure 6.7: Relation extraction output for example sentence 8

After the relations are extracted, we will infer the proper GO terms under the protein complex sub-hierarchy based on the extracted information to annotate the protein. Two types of information are used: (1) protein complex names, and (2) complex components.

Since entries from PRO, Complex portal and CORUM can all be linked to GO entries, if the protein complex we recognized can be matched with the dictionary created from these resources, we can directly associate it with the corresponding GO terms. Note that we do not need to consider the species information during the dictionary lookup process, since GO terms are not species specific.

If we cannot figure out more information for the protein complex mentioned in the article, we will just annotate the proteins involved in the extracted relations with GO term “protein complex” (GO:0043234).

6.3 Evaluation

6.3.1 Evaluation Setup

The BC4GO corpus is a publicly available corpus for the Gene Ontology Annotation task. It consists of a set of articles and associates GO annotations for these articles. This corpus contains 200 full-text articles, 100 of them were designated for training, 50 for development, and the remaining 50 were used for testing. Annotations

in this corpus include the PMID, Gene ID and GOID triplets (a list of relevant GO terms for genes in a paper).

From the BC4GO test set, we extract the PMID, Gene ID and GOID triplets with annotated GO terms covered by the range of our methodologies, and use these triplets for evaluation. Altogether. 97 of the 102 PMID, Gene ID, and GOID triplets with annotation from the Cellular Component domain are extracted and used for evaluation. According to the annotation guideline, we only try to predict GO terms based on the text from the abstract, results section, discussion section, and conclusion section.

6.3.2 Evaluation Results

Table 6.2 shows the precision, recall and F-measure using our approach to predicting GO terms from Cellular Component Domain for given genes.

Table 6.2 Performances of predicting GO terms for given genes

Precision	**Recall**	**F-measure**
91%	58%	71%

Since many of the annotations used text evidence from multiple sentences, while our approach mainly captures the ones from one single sentence, majority of the FNs are due to fact that we didn't anaphoric expressions such as "both proteins" and "it". For example, in sentence "We monitored the subcellular localization of

ZmOST1-GFP and ZmSNAC1-GFP constructs in Nicotiana benthamiana and found that both proteins are localized in the nucleus and the cytoplasm of tobacco epidermal cells (Figure 3).” from PMID 23469147 (PMC3585266). If we can anaphoric “both proteins” to “ZmOST1” and “ZmSNAC1”, then we can capture the relations for the two proteins to location “nucleus” and “cytoplasm”, based on the syntactic dependencies. The remaining FNs correspond to cases where there is no clear-cut syntactic dependencies between the protein and the location. Thus, the syntactic dependency driven approach is unlikely to have captured these cases. We found sentential co-occurrence with careful restrictions might help to improve the recall without hurting the precision. This can be a potential future investigation topic.

6.4 Conclusion

In this chapter, we have developed a system to automatically annotating genes using GO Terms from Cellular Component Domain. We have described a novel approach of treating this process as two relation extraction tasks: (1) extract cases where a protein is in a subcellular location, and (2) extract cases where a protein is a subunit. Evaluation results shows that our approach achieves very high precision in this GO annotation process. Thereby we believe our system can be used as an useful tool for the bio-annotators from Model Organism Database groups to accelerate the process of GO annotation.

In the future, we plan to investigate how to improve the recall of our system. One possible direction is to apply sentential co-occurrence heuristic with careful restrictions.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this section, we conclude by summarizing the contributions of this dissertation. Specifically, our contributions for each task introduced in this dissertation are as follows:

1. Gene normalization. We have developed a system, pGenN, which fulfills a need for automatic detection of plant gene names in the literature and their normalization to UniProt. In developing pGenN, we have introduced a new concept of pivot that has been used in all phases of the normalization and helps improve the performance. We believe this concept can be also used for other named entity recognition and normalization tasks. We also have introduced a method to create a large gene mention corpus to learn the plant gene context. This method can be generalized not only for entities of other types and from other specific sub-domains, but also for distant supervision for development of NER tools. Evaluation on two in-house annotated corpora shows pGenN achieves state of the art performance on abstract level plant gene normalization (with 88.9% and 90.0% F-measures on the two corpora respectively). pGenN has been used to process a comprehensive set of over 527,481 plant related Medline abstracts. The results, which are updated monthly in sync with PubMed, have been stored in our local database, pGenN_DB, and can be

searched, sorted and downloaded via a web interface, found at http://biotm.cis.udel.edu/gn. pGenN was also extended to process full length articles, with a focus on the species assignment process for genes in the results section. The idea used in the task, i.e., treating different full length article sections differently, especially the results section, can be explored for other text mining tasks applied on full length articles.

2. Extending computationally mapped bibliography for UniProt. We have presented a system, eGenPub, which utilizes a trained SVM model to automatically predict whether an article, based on its abstract, is appropriate for additional bibliography for some UniProt entry. We have evaluated the SVM model for predicting whether the article can be linked to the corresponding UniProt entry. Evaluation results show this model obtains 95.3% precision on our in-house annotated corpus. A full-scale PubMed processing has been conducted using eGenPub for 8 common plant species. The literature collected by our system has already been integrated in the UniProt production of computationally mapped literature, and can be accessed via the UniProtKB protein entry publication view. So far, 9,025 articles have been suggested as relevant bibliography for 4,752 UniProt entries, among which 5,252 articles are additional papers not in the existing UniProt publication section. Additionally, we can use our approach to detect the aboutness of an article for a specific entity (not limited to genes) and we believe this can benefit many curation tasks as well as many information retrieval tasks where ranking of articles is needed.

3. Text mining of protein complex related information. We have presented our work on three subtasks that are intended to assist in improving the coverage in protein complex resources: (I) protein complex mention recognition, (II) protein complex and component relation extraction, and (III) protein complex component and component relation extraction. For the recognition of protein complex mentions, we have delivered the first system to serve the community. Evaluation shows our system achieves good results (with a F-measure of 81% on our in-house annotated corpus). In additional to the good performance, we believe the idea of automatically creating a name dictionary by extracting names from the literature using high confidence rules, can be used in other tasks where a dictionary of names needs to be created. For the two complex related relation extraction tasks, we have shown that our approaches achieves very high precision (with 93% and 86% precision respectively). Thus, we believe our work on the three subtasks can provide high- confidence text evidence for the protein complex resources such as Complex Portal, and can be used to assist in the improvement of the coverage in these resources.

4. Gene annotation using GO terms from Cellular Component domain. We have described a novel approach of treating annotation with GO terms from this domain as two subtasks, where we cast the task as relation extraction between gene and other entities: (1) extract cases where a protein is found to be in a subcellular location, and (2) extract cases where a protein is a subunit of a protein complex. We tested our approach on the BC4GO test set, and evaluation results show that our approach 91% precision for predicting GO terms for the given genes. Thereby we

believe our system can be used as a~~n~~ useful tool for the bio-annotators to accelerate the process of GO annotation.

It can be noted that there is not much prior work to be compared with. In fact, as far as we are aware, we were the first to tackle many of the tasks included in this dissertation. This is the reason why there were no other systems that could be compared with.

7.2 Future Work

In addition to the methods/systems that we have presented in this dissertation, our study opens up several opportunities for future work. One main direction is to integrate the gene normalization system with the protein complex recognition system. Since the contexts of protein complex and gene can be similar (as discussed in Chapter 5), this integration has the potential to improve the accuracy of both systems. Another direction is to extend the work and build a full system for gene mention detection and normalization in full length articles, especially the results sections.

Another main direction is to continue our work on gene annotation for the remaining aspects of GO. We have introduced our approach of treating annotation with GO terms from the Cellular Component domain as two subtasks, where we cast each subtask as relation extraction between gene and other entities. This idea can also be applied to other types of GO terms. For example, based on our preliminary study, we believe our approach can be especially useful for two major sub-hierarchies of the

Molecular Function domain: protein binding terms and catalytic activity terms. Annotation with terms from these two sub-hierarchies can be benefit from: (1) extracting cases where a protein binds to other proteins, and (2) extracting cases where a protein is involved in catalytic activity. We believe approaches following these processes can yield high precision, thus can assist in the process of GO annotation.

REFERENCES

[1] The UniProt Consortium. (2017). UniProt: the universal protein knowledgebase. Nucleic Acids Research, 45(D1), D158–D169.

[2] Torii, M., Arighi, C. N., Li, G., Wang, Q., Wu, C. H., & Vijay-Shanker, K. (2015). RLIMS-P 2.0: A Generalizable Rule-Based Information Extraction System for Literature Mining of Protein Phosphorylation Information. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 12(1), 17–29

[3] Ross, K. E., Huang, H., Ren, J., Arighi, C. N., Li, G., Tudor, C. O., … Wu, C. H. (2017). iPTMnet: Integrative Bioinformatics for Studying PTM Networks. Methods in Molecular Biology , 1558, 333–353.

[4] Liu, H., Torii, M., Hu, Z. Z., & Wu, C. (2007). Gene mention and gene normalization based on machine learning and online resources. In Proc of the Second BioCreative Challenge Workshop (pp. 135–140). CNIO.

[5] Fluck, J., Mevissen, H. T., Dach, H., Oster, M., & Hofmann-Apitius, M. (2007). ProMiner: recognition of human gene and protein names using regularly updated dictionaries. In Proceedings of the second BioCreAtIvE challenge evaluation workshop (pp. 149–151). Centro Nacional de Investigaciones Oncologicas, CNIO.

[6] Hakenberg, J., Plake, C., Leaman, R., Schroeder, M., & Gonzalez, G. (2008). Inter-species normalization of gene mentions with GNAT. Bioinformatics , 24(16), i126–132.

[7] Wermter, J., Tomanek, K., & Hahn, U. (2009). High-performance gene name normalization with GeNo. Bioinformatics , 25(6), 815–821.

[8] Bhattacharya, S., Sehgal, A. K., & Srinivasan, P. (2010). Cross-species gene normalization at the University of Iowa. In Proceedings of the BioCreative III workshop (pp. 55–59).

[9] Dai, H.-J., Lai, P.-T., & Tsai, R. T.-H. (2010). Multistage gene normalization and SVM-based ranking for protein interactor extraction in full-text articles. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 7(3), 412–420.

[10] Huang, M., Liu, J., & Zhu, X. (2011). GeneTUKit: a software for document-level gene normalization. Bioinformatics , 27(7), 1032–1033.

[11] Wei, C.-H., & Kao, H.-Y. (2011). Cross-species gene normalization by species inference. BMC Bioinformatics, 12 Suppl 8, S5.

[12] Wei, C.-H., Kao, H.-Y., & Lu, Z. (2015). GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains. BioMed Research International, 2015, 918710.

[13] Ding, R., Arighi, C. N., Lee, J.-Y., Wu, C. H., & Vijay-Shanker, K. (2015). pGenN, a gene normalization tool for plant genes and proteins in scientific literature. PloS One, 10(8), e0135305.

[14] Gingras, A.-C., Aebersold, R., & Raught, B. (2005). Advances in protein complex analysis using mass spectrometry. The Journal of Physiology, 563(Pt 1), 11–21.

[15] Meldal, B. H. M., Forner-Martinez, O., Costanzo, M. C., Dana, J., Demeter, J., Dumousseau, M., … Orchard, S. (2015). The complex portal--an encyclopaedia of macromolecular complexes. Nucleic Acids Research, 43(Database issue), D479–84.

[16] Natale, D. A., Arighi, C. N., Blake, J. A., Bult, C. J., Christie, K. R., Cowart, J., … Wu, C. H. (2014). Protein Ontology: a controlled structured network of protein entities. Nucleic Acids Research, 42(Database issue), D415–21.

[17] Ruepp, A., Waegele, B., Lechner, M., Brauner, B., Dunger-Kaltenbach, I., Fobo, G., … Mewes, H.-W. (2010). CORUM: the comprehensive resource of mammalian protein complexes--2009. Nucleic Acids Research, 38(Database issue), D497–501.

[18] Harris, M. A., Clark, J., Ireland, A., Lomax, J., Ashburner, M., Foulger, R., … Gene Ontology Consortium. (2004). The Gene Ontology (GO) database and informatics resource. Nucleic Acids Research, 32(Database issue), D258–61.

[19] Mao, Y., Van Auken, K., Li, D., Arighi, C. N., McQuilton, P., Hayman, G. T., … Lu, Z. (2014). Overview of the gene ontology task at BioCreative IV. Database: The Journal of Biological Databases and Curation, 2014. https://doi.org/10.1093/database/bau086

[20] Zhu, D., Li, D., Carterette, B., & Liu, H. (2014). Integrating information retrieval with distant supervision for gene ontology annotation. Database: The Journal of Biological Databases and Curation, 2014. https://doi.org/10.1093/database/bau087

[21] Li, Y., & Yu, H. (2014). A robust data-driven approach for gene ontology annotation. Database: The Journal of Biological Databases and Curation, 2014, bau113.

[22] Gaudan, S., Jimeno Yepes, A., Lee, V., & Rebholz-Schuhmann, D. (2008). Combining evidence, specificity, and proximity towards the normalization of Gene Ontology terms in text. EURASIP Journal on Bioinformatics & Systems Biology, 342746.

[23] Couto, F. M., Silva, M. J., Lee, V., Dimmer, E., Camon, E., Apweiler, R., … Rebholz-Schuhmann, D. (2006). GOAnnotator: linking protein GO annotations to evidence text. Journal of Biomedical Discovery and Collaboration, 1, 19.

[24] Emadzadeh, E., Nikfarjam, A., Ginn, R. E., & Gonzalez, G. (2014). Unsupervised gene function extraction using semantic vectors. Database: The Journal of Biological Databases and Curation, 2014. https://doi.org/10.1093/database/bau084

[25] Tuan, L. A., Kim, J.-J., & Ng, S.-K. (2013). Gene ontology concept recognition using cross-products and statistical methods. In BioCreative Challenge Evaluation Workshop vol. (p. 174).

[26] Chen, J.-M., Chang, Y.-C., Wu, J. C.-Y., Lai, P.-T., & Dai, H.-J. (2013). Gene ontology evidence sentence retrieval using combinatorial applications of semantic class and rule patterns. In Proceedings of the Fourth BioCreative Challenge Evaluation Workshop, Bethesda, MD, USA (Vol. 1, pp. 169–173).

[27] Van Auken, K., Schaeffer, M. L., McQuilton, P., Laulederkind, S. J. F., Li, D., Wang, S.-J., … Lu, Z. (2014). BC4GO: a full-text corpus for the BioCreative IV GO task. Database: The Journal of Biological Databases and Curation, 2014. https://doi.org/10.1093/database/bau074

[28] Yeh, A., Morgan, A., Colosimo, M., & Hirschman, L. (2005). BioCreAtIvE task 1A: gene mention finding evaluation. BMC Bioinformatics, 6 Suppl 1, S2.

[29] Smith, L., Tanabe, L. K., Ando, R. J. N., Kuo, C.-J., Chung, I.-F., Hsu, C.-N., … Wilbur, W. J. (2008). Overview of BioCreative II gene mention recognition. Genome Biology, 9 Suppl 2, S2.

[30] Hanisch, D., Fundel, K., Mevissen, H.-T., Zimmer, R., & Fluck, J. (2004). ProMiner: Organism-specific protein name detection using approximate string matching. BioCreative: Critical Assessment for Information Extraction in Biology.

[31] Fukuda, K., Tamura, A., Tsunoda, T., & Takagi, T. (1998). Toward information extraction: identifying protein names from biological papers. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 707–718.

[32] Narayanaswamy, M., Ravikumar, K. E., & Vijay-Shanker, K. (2003). A biological named entity recognizer. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 427–438.

[33] Leaman, R., & Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 652–663.

[34] Liu, H., Torii, M., Hu, Z. Z., & Wu, C. (2007). Gene mention and gene normalization based on machine learning and online resources. In Proc of the Second BioCreative Challenge Workshop (pp. 135–140). CNIO.

[35] John Lafferty, C. M. U., Andrew McCallum, W. L., Fernando C.N. Pereira, U. of P., & Authors. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Retrieved from http://repository.upenn.edu/cis_papers/159/

[36] Liu, H., Hu, Z.-Z., Zhang, J., & Wu, C. (2006). BioThesaurus: a web-based thesaurus of protein and gene names. Bioinformatics , 22(1), 103–105.

[37] Huang, H.-S., Lin, Y.-S., Lin, K.-T., Kuo, C.-J., Chang, Y.-M., Yang, B.-H., … Hsu, C.-N. (2007). High-recall gene mention recognition by unification of multiple backward parsing models. In Proceedings of the second BioCreative challenge evaluation workshop (Vol. 23, pp. 109–111). Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain.

[38] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory (pp. 144–152). New York, NY, USA: ACM.

[39] Chen, Y., Liu, F., & Manderick, B. (2007). Improving the performance of gene mention recognition system using reformed lexicon-based support vector machine. Margin, 500, 2.

[40] Zhou, G., Shen, D., Zhang, J., Su, J., & Tan, S. (2005). Recognition of protein/gene names from text using an ensemble of classifiers. BMC Bioinformatics, 6 Suppl 1, S7.

[41] Settles, B. (2005). ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. Bioinformatics, 21(14), 3191–3192.

[42] Finkel, J., Dingare, S., Manning, C. D., Nissim, M., Alex, B., & Grover, C. (2005). Exploring the boundaries: gene and protein identification in biomedical text. BMC Bioinformatics, 6 Suppl 1, S5.

[43] McDonald, R., & Pereira, F. (2005). Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics, 6 Suppl 1, S6.

[44] Kinoshita, S., Ogren, P., Cohen, K. B., & Hunter, L. (2004). Entity identification in the molecular biology domain with a stochastic POS tagger: the BioCreative task. In BioCreAtIvE Workshop, Granada, Spain.

[45] Ando, R. K. (2007). BioCreative II gene mention tagging system at IBM Watson. In Proceedings of the Second BioCreative Challenge Evaluation Workshop (Vol. 23, pp. 101–103). Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain.

[46] Kuo, C.-J., Chang, Y.-M., Huang, H.-S., Lin, K.-T., Yang, B.-H., Lin, Y.-S., … Chung, I.-F. (2007). Rich feature set, unification of bidirectional parsing and dictionary filtering for high F-score gene mention tagging. In Proceedings of the second BioCreative challenge evaluation workshop (Vol. 23, pp. 105–107). Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain.

[47] Klinger, R., Friedrich, C. M., Fluck, J., & Hofmann-Apitius, M. (2007). Named entity recognition with combinations of conditional random fields. In Proceedings of the second biocreative challenge evaluation workshop.

[48] Torii, M., Hu, Z., Wu, C. H., & Liu, H. (2009). BioTagger-GM: a gene/protein name recognition system. Journal of the American Medical Informatics Association: JAMIA, 16(2), 247–255.

[49] Struble, C. A., Povinelli, R. J., Johnson, M. T., Berchanskiy, D., Tao, J., & Trawicki, M. (2007). Combined conditional random fields and n-gram language models for gene mention recognition. In Proceedings of the Second BioCreative Challenge Evaluation Workshop; 23 to 25 April 2007; Madrid, Spain (pp. 81–83).

[50] Baumgartner, W., Lu, Z., Johnson, H. L., Caporaso, J. G., Paquette, J., Lindemann, A., … Others. (2007). An integrated approach to concept recognition in biomedical text. In Proceedings of the Second BioCreative Challenge Evaluation Workshop (Vol. 23, pp. 257–271). CNIO.

[51] Tsai, R. T.-H., Sung, C.-L., Dai, H.-J., Hung, H.-C., Sung, T.-Y., & Hsu, W.-L. (2006). NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. BMC Bioinformatics, 7 Suppl 5, S11.

[52] Kim, J.-D., Ohta, T., Tateisi, Y., & Tsujii, J. (2003). GENIA corpus—a semantically annotated corpus for bio-textmining. Bioinformatics, 19(suppl_1), i180–i182.

[53] Kulick, S., Bies, A., Liberman, M., Mandel, M., McDonald, R., Palmer, M., … White, P. (2004). Integrated annotation for biomedical information extraction. In Proc. of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL) (pp. 61–68).

[54] Hirschman, L., Colosimo, M., Morgan, A., & Yeh, A. (2005). Overview of BioCreAtIvE task 1B: normalized gene lists. BMC Bioinformatics, 6 Suppl 1, S11.

[55] Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., … Hirschman, L. (2008). Overview of BioCreative II gene normalization. Genome Biology, 9 Suppl 2, S3.

[56] Lu, Z., Kao, H.-Y., Wei, C.-H., Huang, M., Liu, J., Kuo, C.-J., … Wilbur, W. J. (2011). The gene normalization task in BioCreative III. BMC Bioinformatics, 12 Suppl 8, S2.

[57] Wei, C.-H., Kao, H.-Y., & Lu, Z. (2012). SR4GN: a species recognition software tool for gene normalization. PloS One, 7(6), e38460.

[58] Wei, C.-H., Kao, H.-Y., & Lu, Z. (2013). PubTator: a web-based text mining tool for assisting biocuration. Nucleic Acids Research, 41(Web Server issue), W518–22.

[59] Van Landeghem, S., Björne, J., Wei, C.-H., Hakala, K., Pyysalo, S., Ananiadou, S., … Ginter, F. (2013). Large-scale event extraction from literature with multi-level gene normalization. PloS One, 8(4), e55814.

[60] Leaman, R., Wei, C.-H., & Lu, Z. (2015). tmChem: a high performance approach for chemical named entity recognition and normalization. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S3.

[61] Lu, Y., Ji, D., Yao, X., Wei, X., & Liang, X. (2015). CHEMDNER system with mixed conditional random fields and multi-scale word clustering. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S4.

[62] Lowe, D. M., & Sayle, R. A. (2015). LeadMine: a grammar and dictionary driven approach to entity recognition. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S5.

[63] Mahbub Chowdhury, M. F., & Lavelli, A. (2010). Disease Mention Recognition with Specific Features. In Proceedings of the 2010 Workshop on Biomedical Natural Language Processing (pp. 83–90). Stroudsburg, PA, USA: Association for Computational Linguistics.

[64] Kaewphan, S., Hakala, K., & Ginter, F. (2014). UTU: Disease Mention Recognition and Normalization with CRFs and Vector Space Representations. In SemEval@ COLING (pp. 807–811).

[65] Batista-Navarro, R., Rak, R., & Ananiadou, S. (2015). Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S6.

[66] Munkhdalai, T., Li, M., Batsuren, K., Park, H. A., Choi, N. H., & Ryu, K. H. (2015). Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S9.

[67] Akhondi, S. A., Hettne, K. M., van der Horst, E., van Mulligen, E. M., & Kors, J. A. (2015). Recognition of chemical entities: combining dictionary-based and grammar-based approaches. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S10.

[68] Khabsa, M., & Giles, C. L. (2015). Chemical entity extraction using CRF and an ensemble of extractors. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S12.

[69] Xu, S., An, X., Zhu, L., Zhang, Y., & Zhang, H. (2015). A CRF-based system for recognizing chemical entity mentions (CEMs) in biomedical literature. Journal of Cheminformatics, 7(Suppl 1 Text mining for chemistry and the CHEMDNER track), S11.

[70] Leaman, R., & Lu, Z. (2014). Automated disease normalization with low rank approximations. In Proceedings of BioNLP (pp. 24–28).

[71] Dogan, R. I., & Lu, Z. (2012). An Inference Method for Disease Name Normalization. In AAAI Fall Symposium: Information Retrieval and Knowledge Discovery in Biomedical Text. Retrieved from http://www.aaai.org/ocs/index.php/FSS/FSS12/paper/viewFile/5604/5843

[72] Kang, N., Singh, B., Afzal, Z., van Mulligen, E. M., & Kors, J. A. (2013). Using rule-based natural language processing to improve disease normalization in biomedical text. Journal of the American Medical Informatics Association: JAMIA, 20(5), 876–881.

[73] Lee, H.-C., Hsu, Y.-Y., & Kao, H.-Y. (2015). An enhanced CRF-based system for disease name entity recognition and normalization on BioCreative V DNER Task. In Proceedings of the Fifth BioCreative Challenge Evaluation Workshop (pp. 226–233).

[74] Deleger, L., Grouin, C., & Bossy, R. (2015). Hybrid approaches for the DNER task at BioCreative V: the INRA/LIMSI system. In Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, 154-166.

[75] Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., & Tsujii, J. 'ichi. (2009). Overview of BioNLP'09 Shared Task on Event Extraction. In Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task (pp. 1–9). Stroudsburg, PA, USA: Association for Computational Linguistics.

[76] Kim, J.-D., Nguyen, N., Wang, Y., Tsujii, J. 'ichi, Takagi, T., & Yonezawa, A. (2012). The Genia Event and Protein Coreference tasks of the BioNLP Shared Task 2011. BMC Bioinformatics, 13 Suppl 11, S1.

[77] Kim, J.-D., Wang, Y., & Yasunori, Y. (2013). The genia event extraction shared task, 2013 edition-overview. In Proceedings of the BioNLP Shared Task 2013 Workshop (pp. 8–15). Association for Computational Linguistics.

[78] Krallinger, M., Leitner, F., Rodriguez-Penagos, C., & Valencia, A. (2008). Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology, 9 Suppl 2, S4.

[79] Krallinger, M., Vazquez, M., Leitner, F., Salgado, D., Chatr-Aryamontri, A., Winter, A., … Valencia, A. (2011). The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC Bioinformatics, 12 Suppl 8, S3.

[80] Bui, Q.-C., Campos, D., van Mulligen, E., & Kors, J. (2013). A fast rule-based approach for biomedical event extraction. In Proceedings of the BioNLP Shared Task 2013 Workshop (pp. 104–108). Association for Computational Linguistics.

[81] Nebhi, K. (2013). A Rule-based Relation Extraction System Using DBpedia and Syntactic Parsing. In Proceedings of the 2013th International Conference on NLP & DBpedia - Volume 1064 (pp. 74–79). Aachen, Germany, Germany: CEUR-WS.org.

[82] Li, G., Ross, K. E., Arighi, C. N., Peng, Y., Wu, C. H., & Vijay-Shanker, K. (2015). miRTex: A Text Mining System for miRNA-Gene Relation Extraction. PLoS Computational Biology, 11(9), e1004391.

[83] Peng, Y., Arighi, C., Wu, C. H., & Vijay-Shanker, K. (2016). BioC-compatible full-text passage detection for protein-protein interactions using extended dependency graph. Database: The Journal of Biological Databases and Curation, 2016. https://doi.org/10.1093/database/baw072

[84] Kim, J.-J., & Rebholz-Schuhmann, D. (2011). Improving the extraction of complex regulatory events from scientific text by using ontology-based inference. Journal of Biomedical Semantics, 2 Suppl 5, S3.

[85] Kilicoglu, H., & Bergler, S. (2011). Adapting a General Semantic Interpretation Approach to Biological Event Extraction. In Proceedings of the BioNLP Shared Task 2011 Workshop (pp. 173–182). Stroudsburg, PA, USA: Association for Computational Linguistics.

[86] Cohen, K. B., Verspoor, K., Johnson, H. L., Roeder, C., Ogren, P. V., Baumgartner, W. A., Jr, … Hunter, L. (2011). HIGH-PRECISION BIOLOGICAL EVENT EXTRACTION: EFFECTS OF SYSTEM AND OF DATA. Computational Intelligence. An International Journal, 27(4), 681–701.

[87] Hakenberg, J., Leaman, R., Vo, N. H., Jonnalagadda, S., Sullivan, R., Miller, C., … Gonzalez, G. (2010). Efficient extraction of protein-protein interactions from full-text articles. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 7(3), 481–494.

[88] Narayanaswamy, M., Ravikumar, K. E., & Vijay-Shanker, K. (2005). Beyond the clause: extraction of phosphorylation information from medline abstracts. Bioinformatics , 21 Suppl 1, i319–27.

[89] Van Landeghem, S., Ginter, F., Van de Peer, Y., & Salakoski, T. (2011). EVEX: A Pubmed-scale Resource for Homology-based Generalization of Text Mining Predictions. In Proceedings of BioNLP 2011 Workshop (pp. 28–37). Stroudsburg, PA, USA: Association for Computational Linguistics.

[90] Liu, X., Bordes, A., & Grandvalet, Y. (2013). Biomedical event extraction by multi-class classification of pairs of text entities. In BioNLP Shared Task 2013 Workshop (pp. 45–49).

[91] Björne, J., & Salakoski, T. (2013). TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task. In Proceedings of the BioNLP Shared Task 2013 Workshop (pp. 16–25). Association for Computational Linguistics.

[92] Miwa, M., Sætre, R., Miyao, Y., & Tsujii, J. (2009). Protein–protein interaction extraction by leveraging multiple kernels and parsers. International Journal of Medical Informatics, 78(12), e39–e46.

[93] Björne, J., Ginter, F., Pyysalo, S., Tsujii, J. 'ichi, & Salakoski, T. (2010). Complex event extraction at PubMed scale. Bioinformatics , 26(12), i382–90.

[94] Bui, Q.-C., Katrenko, S., & Sloot, P. M. A. (2011). A hybrid approach to extract protein-protein interactions. Bioinformatics , 27(2), 259–265.

[95] Miwa, M., Sætre, R., Miyao, Y., & Tsujii, J. 'ichi. (2009). A Rich Feature Vector for Protein-protein Interaction Extraction from Multiple Corpora. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1 (pp. 121–130). Stroudsburg, PA, USA: Association for Computational Linguistics.

[96] Riedel, S., McClosky, D., Surdeanu, M., McCallum, A., & Manning, C. D. (2011). Model Combination for Event Extraction in BioNLP 2011. In Proceedings of the BioNLP Shared Task 2011 Workshop (pp. 51–55). Stroudsburg, PA, USA: Association for Computational Linguistics.

[97] Vlachos, A., & Craven, M. (2012). Biomedical event extraction from abstracts and full papers using search-based structured prediction. BMC Bioinformatics, 13 Suppl 11, S5.

[98] Baumgartner, W. A., Jr, Cohen, K. B., Fox, L. M., Acquaah-Mensah, G., & Hunter, L. (2007). Manual curation is not sufficient for annotation of genomic databases. Bioinformatics, 23(13), i41–8.

[99] Badawi, O., Brennan, T., Celi, L. A., Feng, M., Ghassemi, M., Ippolito, A., … MIT Critical Data Conference 2014 Organizing Committee. (2014). Making big data useful for health care: a summary of the inaugural mit critical data conference. JMIR Medical Informatics, 2(2), e22.

[100] Krallinger, M., Valencia, A., & Hirschman, L. (2008). Linking genes to literature: text mining, information extraction, and retrieval applications for biology. Genome Biology, 9 Suppl 2, S8.

[101] Arabidopsis nomenclature [http://www.arabidopsis.org/portals/nomenclature/guidelines.jsp]

[102] VandenBosch, K. A., & Frugoli, J. (2001). Guidelines for genetic nomenclature and community governance for the model legume Medicago truncatula. Molecular Plant-Microbe Interactions: MPMI, 14(12), 1364–1367.

[103] Ostergaard, L., & King, G. J. (2008). Standardized gene nomenclature for the Brassica genus. Plant Methods, 4, 10.

[104] Grimplet, J., Adam-Blondon, A.-F., Bert, P.-F., Bitz, O., Cantu, D., Davies, C., … Cramer, G. R. (2014). The grapevine gene nomenclature system. BMC Genomics, 15, 1077.

[105] McCouch, S. R., & CGSNL (Committee on Gene Symbolization, Nomenclature and Linkage, Rice Genetics Cooperative). (2008). Gene Nomenclature System for Rice. Rice , 1(1), 72–84.

[106] The PLANTS Web Site: Understanding Its Basic Functionality [http://plants.usda.gov/plants_tutorial.pdf]

[107] Verspoor, K., Roeder, C., Johnson, H. L., Cohen, K. B., Baumgartner, W. A., Jr, & Hunter, L. E. (2010). Exploring species-based strategies for gene normalization. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM, 7(3), 462–471.

[108] Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2013). GenBank. Nucleic Acids Research, 41(Database issue), D36–42.

[109] Schwartz, A. S., & Hearst, M. A. (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 451–462.

[110] Peng, Y., Tudor, C. O., Torii, M., Wu, C. H., & Vijay-Shanker, K. (2012). iSimp: A sentence simplification system for biomedicail text. In 2012 IEEE International Conference on Bioinformatics and Biomedicine (pp. 1–6).

[111] Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 133–142). New York, NY, USA: ACM.

[112] Wu, H. C., Luk, R. W. P., Wong, K. F., & Kwok, K. L. (2008). Interpreting TF-IDF Term Weights As Making Relevance Decisions. ACM Transactions on Information and System Security, 26(3), 13:1–13:37.

[113] Gruszka, D. (2013). The brassinosteroid signaling pathway-new key players and interconnections with other signaling networks crucial for plant development and stress tolerance. International Journal of Molecular Sciences, 14(5), 8740–8774.

[114] Poux, S., Magrane, M., Arighi, C. N., Bridge, A., O'Donovan, C., Laiho, K., & UniProt Consortium. (2014). Expert curation in UniProtKB: a case study on dealing with conflicting and erroneous data. Database: The Journal of Biological Databases and Curation, 2014, bau016.

[115] Poux, S., Arighi, C. N., Magrane, M., Bateman, A., Wei, C.-H., Lu, Z., … Roechert, B. (2016, December 14). On expert curation and sustainability: UniProtKB/Swiss-Prot as a case study. bioRxiv. https://doi.org/10.1101/094011

[116] Howe, K. L., Bolt, B. J., Cain, S., Chan, J., Chen, W. J., Davis, P., … Sternberg, P. W. (2016). WormBase 2016: expanding to enable helminth genomic research. Nucleic Acids Research, 44(D1), D774–80.

[117] Shimoyama, M., De Pons, J., Hayman, G. T., Laulederkind, S. J. F., Liu, W., Nigam, R., … Jacob, H. (2015). The Rat Genome Database 2015: genomic, phenotypic and environmental variations and disease. Nucleic Acids Research, 43(Database issue), D743–50.

[118] Kerrien, S., Aranda, B., Breuza, L., Bridge, A., Broackes-Carter, F., Chen, C., … Hermjakob, H. (2012). The IntAct molecular interaction database in 2012. Nucleic Acids Research, 40(Database issue), D841–6.

[119] Lamesch, P., Berardini, T. Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., … Huala, E. (2012). The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. Nucleic Acids Research, 40(Database issue), D1202–10.

[120] Jimeno-Yepes, A. J., Sticco, J. C., Mork, J. G., & Aronson, A. R. (2013). GeneRIF indexing: sentence selection based on machine learning. BMC Bioinformatics, 14, 171.

[121] IC4R Project Consortium, Hao, L., Zhang, H., Zhang, Z., Hu, S., & Xue, Y. (2016). Information Commons for Rice (IC4R). Nucleic Acids Research, 44(D1), D1172–80.

[122] Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bansal, P., Bridge, A. J., … Xenarios, I. (2016). UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View. Methods in Molecular Biology , 1374, 23–54.

[123] http://geneontology.org/page/protein-complexes

[124] Okazaki, N. (2007). CRFsuite: a fast implementation of Conditional Random Fields. 2015-03-24]. Http://www. Chokkan. Org/software/crfsuite.

[125] Charniak, E. (2000). A Maximum-entropy-inspired Parser. In Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (pp. 132–139). Stroudsburg, PA, USA: Association for Computational Linguistics.

[126] Mcclosky, D. (2010). Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. Brown University, Providence, RI, USA. Retrieved from http://dl.acm.org/citation.cfm?id=2020153

Appendix A

TEMPLATE TO WRITE RULES BASED ON SYNTACTIC DEPENDENCIES

In this section, we describe the template for writing rules for relation extraction based on syntactic dependencies. The template consists 3 parts: (1) RuleID: name/symbol for the rule, (2) Cond_#: the conditions that need to be satisfied, and (3) Action_#: the dependency edge to be added. I will use an example to illustrate how this template works. Consider the rule in Table A.1, which captures the protein complex component and component relation for "Mad1" and "Max" in sentence "Mad1 dimerizes with Max". The syntactic dependencies are shown in Figure A.1.

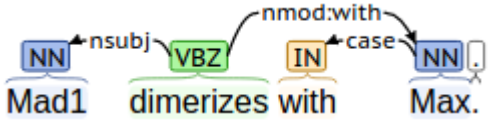

Figure A.1: Syntactic dependencies for example sentence

Table A.1 Example rule

RuleID: pp_dimerizeWith
Cond_0 : {lemma:/(.*dimerize|complex)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:with/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

For the rule shown in Table A.1, there are 3 conditions needs to be satisfied: (1) the lemma of one node "N0" (in this case "dimerizes") needs to be matched with

the regular expression /.*dimerize|complex/, (2) there needs to be a “nsubj” edge points from node “N0” to another node “N1” (in this case “Mad1”), and (3) there needs to be a “nmod:with” edge points from node “N1” to another node “N2” (in this case “Max”). If these three conditions are all satisfied, then we add an “argComponent” edge from “N0” to “N1”, and an “argComponent” edge from “N0” to “N2”, as shown in Figure A.2.

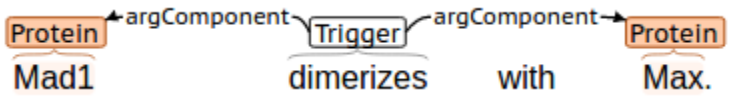

Figure A.2: Output for example sentence

Appendix B

RULES FOR DETECTING PROTEIN COMPLEX AND COMPONENT RELATION

In this section, we list the three types of rules (described in Chapter 5) we developed for detecting Protein Complex and Component Relation. The rules are in the same form with the template described in Appendix A.

Type 1 rules:

RuleID : pc_IsASubunitOf
Cond_0 : {lemma:/(subunit|member|component|constituent|part)/}=N0
Cond_1 : {}=N0 >/(nmod:of)/ {}=N1
Action_1 : N0 >> argProtein >> N0
Action_2 : N0 >> argComplex >> N1

RuleID : pc_IsAComplexSubunit
Cond_0 : {lemma:/(subunit|member|component|constituent|part)/}=N0
Cond_1 : {}=N0 >/(compound)/ {}=N1
Action_1 : N0 >> argProtein >> N0
Action_2 : N0 >> argComplex >> N1

RuleID : pc_compoundTail
Cond_0 : {lemma:/(subunit|member|component|constituent|part)/}=N0
Cond_1 : {}=N0 >/(compound)/ {}=N1
Cond_2 : {}=N0 >/(nmod:of)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pc_compoundHead
Cond_0 : {lemma:/(subunit|member|component|constituent|part)/}=N0
Cond_1 : {}=N0 >/(dep|compound)/ {}=N1
Cond_2 : {}=N0 >/(compound)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

Type 2 rules:

RuleID : pc_contains
Cond_0 : {lemma:/(contain|include)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >dobj {}=N2
Action_1 : N0 >> argComplex >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pc_consistsOf
Cond_0 : {lemma:/(consist)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:(of)/ {}=N2
Action_1 : N0 >> argComplex >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pc_containing
Cond_0 : {word:/(containing)/}=N0
Cond_1 : {}=N2 >acl {}=N0
Cond_2 : {}=N0 >dobj {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pc_composedOf
Cond_0 : {lemma:/(compose|form|consist)/}=N0
Cond_1 : {}=N1 >acl {}=N0
Cond_2 : {}=N0 >/nmod:(of|by)/ {}=N2
Action_1 : N0 >> argComplex >> N1

Action_2 : N0 >> argProtein >> N2

RuleID : pc_isComposedOf
Cond_0 : {lemma:/(compose|form|consist)/}=N0
Cond_1 : {}=N0 >nsubjpass {}=N1
Cond_2 : {}=N0 >/nmod:(of|by)/ {}=N2
Action_1 : N0 >> argComplex >> N1
Action_2 : N0 >> argProtein >> N2

Type 3 rules:

RuleID : pc_detectedIn
Cond_0 : {lemma:/(found|detect|observe|discover|note)/}=N0
Cond_1 : {}=N0 >nsubjpass {}=N1
Cond_2 : {}=N0 >/nmod:(in|on|at|throughout|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pc_presentIn
Cond_0 : {lemma:/(present)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:(in)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pc_engagedIn
Cond_0 : {lemma:/(engage)/}=N0
Cond_1 : {}=N0 >nsubjpass {}=N1
Cond_2 : {}=N0 >/nmod:(in|throughout)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pc_associationInComplex
Cond_0 : {lemma:/(.*association|binding)/}=N0

Cond_1 :
{lemma:/(complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/
}=N3
Cond_2 : {}=N0 >/nmod:of/ {}=N1
Cond_3 : {}=N0 >/nmod:(with|to)/ {}=N2
Cond_4 : {}=N0 >/nmod:in/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2
Action_3 : N0 >> argComplex >> N3

RuleID : pc_associationInComplex
Cond_0 : {lemma:/(.*association|binding)/}=N0
Cond_1 :
{lemma:/(complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/
}=N2
Cond_2 : {}=N0 >/nmod:of/ {}=N1
Cond_3 : {}=N0 >/nmod:in/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pc_enterComplex
Cond_0 : {lemma:/(enter)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >dobj {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

Appendix C

RULES FOR DETECTING PROTEIN COMPLEX COMPONENT AND COMPONENT RELATION

In this section, we list the two types of rules (described in Chapter 5) we developed for detecting Protein Complex Component and Component Relation. The rules are in the same form with the template described in Appendix A.

Type 1 rules:

RuleID : pp_formComplexWith
Cond_0 : {lemma:/(form)/}=N0
Cond_1 : {lemma:/(complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/}=N2
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >dobj {}=N2
Cond_4 : {}=N2 >/nmod:with/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2
Action_3 : N0 >> argProtein >> N3

RuleID : pp_formComplex
Cond_0 : {lemma:/(form)/}=N0
Cond_1 : {lemma:/(.*complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/}=N2

Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >dobj {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

RuleID : pp_formationOfComplex
Cond_0 : {lemma:/(formation)/}=N0
Cond_1 : {lemma:/(complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/}=N1
Cond_2 : {}=N0 >/nmod:of/ {}=N1
Cond_3 : {}=N0 >/nmod:by/ {}=N2
Action_1 : N0 >> argProtein >> N2
Action_2 : N0 >> argComplex >> N1

RuleID : pp_existInComplexWith
Cond_0 : {lemma:/(exist)/}=N0
Cond_1 : {lemma:/(.*complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/}=N2
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >/nmod:in/ {}=N2
Cond_4 : {}=N2 >/nmod:with/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2
Action_3 : N0 >> argProtein >> N3

RuleID : pp_dimerizeWith
Cond_0 : {lemma:/(.*dimerize|complex)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:with/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_stableInteractionOf
Cond_0 : {lemma:/(interaction|association)/}=N0
Cond_1 : {lemma:/(stable)/}=N1

Cond_2 : {}=N0 >/amod/ {}=N1
Cond_3 : {}=N0 >/nmod:of/ {}=N2
Action_1 : N0 >> argProtein >> N2

RuleID : pp_stableInteractionOfWith
Cond_0 : {lemma:/(interaction|association)/}=N0
Cond_1 : {lemma:/(stable)/}=N1
Cond_2 : {}=N0 >/amod/ {}=N1
Cond_3 : {}=N0 >/nmod:of/ {}=N2
Cond_3 : {}=N0 >/nmod:with/ {}=N3
Action_1 : N0 >> argProtein >> N2
Action_1 : N0 >> argProtein >> N3

RuleID : pp_bindStably
Cond_0 : {lemma:/(bind|associate)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:(with|to)/ {}=N2
Cond_3 : {word:/(stably)/}=N3
Cond_4 : {}=N0 >advmod {}=N3
Action_1 : N0 >> argProtei >> N1
Action_2 : N0 >> argProtei >> N2

RuleID : pp_bind_with_as_complex
Cond_0 : {lemma:/(bind|associate)/}=N0
Cond_1 :
{lemma:/(.*complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octome
r)/}=N3
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >/nmod:(with|to)/ {}=N2
Cond_4 : {}=N0 >/nmod:(as)/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2
Action_2 : N0 >> argComplex >> N3

RuleID : pp_bind_with_as_complex2
Cond_0 : {lemma:/(bind)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1

Cond_2 : {}=N0 >dobj {}=N2
Cond_3 : {lemma:/(.*complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/}=N3
Cond_4 : {}=N0 >/nmod:(as)/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2
Action_2 : N0 >> argComplex >> N3

RuleID : pp_bind_as_complex
Cond_0 : {lemma:/(bind)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {lemma:/(.*complex|.*dimer|.*trimer|.*tramer|.*tamer|.*hexamer|.*nonamer|.*octomer)/}=N2
Cond_3 : {}=N0 >/nmod:(as)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argComplex >> N2

Type 2 rules:

RuleID : pp_associationOfWithFunction
Cond_0 : {lemma:/(interaction|association|binding)/}=N0
Cond_1 : {pos:/(VBZ|VBD)/}=N2
Cond_2 : {}=N0 >/nmod:of/ {}=N1
Cond_4 : {}=N2 >/nsubj/ {}=N0
Action_1 : N0 >> argProtein >> N1

RuleID : pp_interactionBetweenWithFunction
Cond_0 : {lemma:/(interaction|association|binding)/}=N0
Cond_1 : {pos:/(VBZ|VBD)/}=N2
Cond_2 : {}=N0 >/nmod:between/ {}=N1
Cond_4 : {}=N2 >/nsubj/ {}=N0
Action_1 : N0 >> argProtein >> N1

RuleID : pp_associationOfWithFunction

Cond_0 : {lemma:/(association|binding)/}=N0
Cond_1 : {pos:/(VBZ|VBD)/}=N3
Cond_2 : {}=N0 >/nmod:of/ {}=N1
Cond_3 : {}=N0 >/nmod:(with|to)/ {}=N2
Cond_4 : {}=N3 >/nsubj/ {}=N0
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_interactionOfWithFunction
Cond_0 : {lemma:/(interaction)/}=N0
Cond_1 : {pos:/(VBZ|VBD)/}=N3
Cond_2 : {}=N0 >/nmod:of/ {}=N1
Cond_3 : {}=N0 >/nmod:(with)/ {}=N2
Cond_4 : {}=N3 >/nsubj/ {}=N0
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_bindFunction
Cond_0 : {lemma:/(interact|associate|bind)/}=N0
Cond_1 : {pos:/(VB)/}=N3
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >dobj {}=N2
Cond_4 : {}=N0 >advcl {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_bindFunction2
Cond_0 : {lemma:/(associate|bind)/}=N0
Cond_1 : {pos:/(VBN)/}=N3
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >/nmod:(with|to)/ {}=N2
Cond_4 : {}=N3 >advcl {}=N0
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_bindFunction2
Cond_0 : {lemma:/(interact|associate|bind)/}=N0

Cond_1 : {pos:/(VBZ|VBD)/}=N3
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >dobj {}=N2
Cond_4 : {}=N3 >nsubj {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_associateWithFunction
Cond_0 : {lemma:/(interact|associate|bind)/}=N0
Cond_1 : {pos:/(VB)/}=N3
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >/nmod:(with)/ {}=N2
Cond_4 : {}=N3 >nsubj {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

RuleID : pp_isAssociatedWithFunction
Cond_0 : {word:/(associated|bound)/}=N0
Cond_1 : {pos:/(VB)/}=N3
Cond_2 : {}=N0 >nsubjpass {}=N1
Cond_3 : {}=N0 >/nmod:(with|to)/ {}=N2
Cond_4 : {}=N3 >nsubj {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

(For the following type 2 rules, the number of components involved needs to be larger than 2)

RuleID : ppi_associateWith
Cond_0 : {lemma:/(interact|associate|bind|coIP|colocalize)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:(with|to)/ {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_isAssociatedWith
Cond_0 : {word:/(associated|bound)/}=N0

Cond_1 : {}=N0 >nsubjpass {}=N1
Cond_2 : {}=N0 >/nmod:(with|to)/ {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_associated
Cond_0 : {word:/(associated|bound)/}=N0
Cond_1 : {}=N1 >acl {}=N0
Cond_2 : {}=N0 >/nmod:(with|to)/ {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_associate
Cond_0 : {lemma:/(interact|associate|bind|coIP)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >nsubj {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_associationBetween
Cond_0 : {lemma:/(interaction|association|dissociation|binding)/}=N0
Cond_1 : {}=N0 >/nmod:(between|of)/ {}=N1
Cond_2 : {}=N0 >/nmod:(between|of)/ {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_associationOfWith
Cond_0 : {lemma:/(interaction|association|binding)/}=N0
Cond_1 : {}=N0 >/nmod:of/ {}=N1
Cond_2 : {}=N0 >/nmod:(with|to)/ {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_bind
Cond_0 : {lemma:/(interact|associate|bind|coIP)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >dobj {}=N2

Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

RuleID : ppi_combineWith
Cond_0 : {lemma:/(combine|couple|copurify|co-purify)/}=N0
Cond_1 : {}=N0 >nsubj {}=N1
Cond_2 : {}=N0 >/nmod:with/ {}=N2
Action_1 : N0 >> argPPI >> N1
Action_2 : N0 >> argPPI >> N2

Appendix D

RULES FOR DETECTING PROTEIN AND SUBCELLULAR LOCATION RELATION

In this section, we list all the rules we developed for detecting Protein and subcellular location Relation. The rules are in the same form with the template described in Appendix A.

RuleID : pl_foundIn
Cond_1 : {word:/(found|noted|detected|observed|discovered)/}=N0
Cond_2 : {}=N0 >/(nsubjpass)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_localizedTo
Cond_1 : {word:/(localised|localized)/}=N0
Cond_2 : {}=N0 >/(nsubjpass)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|to)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_transportedTo
Cond_1 : {word:/(transported|translocated|displaced|attached)/}=N0
Cond_2 : {}=N0 >/(nsubjpass)/ {}=N1
Cond_3 : {}=N0 >/nmod:(to|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_removedFrom
Cond_1 : {word:/(removed)/}=N0
Cond_2 : {}=N0 >/(nsubjpass)/ {}=N1
Cond_3 : {}=N0 >/nmod:(from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_ExpressedIn
Cond_1 : {word:/(translated|expressed|activated)/}=N0
Cond_2 : {}=N0 >/(nsubjpass)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_stableIn
Cond_1 :
{word:/(stable|visible|detectable|enriched|located|localized|transported)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_expressionOfProteinIn
Cond_1 : {lemma:/(localization|colocalization)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|to)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_expressionOfProteinIn
Cond_1 : {lemma:/(translation|expression|activation)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_distributionOfProteinIn

Cond_1 : {lemma:/(distribution)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_relocalizationOfProteinTo
Cond_0 : {lemma:/(relocalization|translocation|transport)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1
Cond_3 : {}=N0 >/nmod:(to|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_ProteinRelocalizationTo
Cond_1 : {lemma:/(relocalization|translocation|transport)/}=N0
Cond_2 : {}=N0 >compound {}=N1
Cond_3 : {}=N0 >/nmod:(to|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_relocalizeTo
Cond_1 : {lemma:/(relocalize|translocate|transport)/}=N0
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >/nmod:(to|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_relocalizationTo
Cond_1 : {lemma:/(relocalization|translocation|transport)/}=N0
Cond_2 : {}=N0 >compound {}=N1
Cond_3 : {}=N0 >amod {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_JJrelocalizationof
Cond_1 : {lemma:/(relocalization|translocation|transport)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1

Cond_3 : {}=N0 >amod {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_relocalizationTo
Cond_1 : {lemma:/(relocalize|transport)/}=N0
Cond_2 : {}=N0 >dobj {}=N1
Cond_3 : {}=N0 >/nmod:(to|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_enrichmentIn
Cond_1 : {lemma:/(enrichment|decrease|increase)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_increaseIn
Cond_1 : {lemma:/(decrease|increase|accumulate)/}=N0
Cond_2 : {}=N0 >nsubj {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_localizeIn
Cond_1 : {lemma:/(localize|localise|colocalize|relocalize)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|to)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_appearIn
Cond_1 : {lemma:/(occur|appear)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1

Action_2 : N0 >> argLocation >> N2

RuleID : pl_colocalizeWithIn
Cond_1 : {lemma:/(colocalize)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(with)/ {}=N2
Cond_4 : {}=N0 >/nmod:(in|at|on|throughout|within|to)/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2
Action_3 : N0 >> argLocation >> N3

RuleID : pl_JJlocalizationOf
Cond_1 : {lemma:/(localization)/}=N0
Cond_2 : {}=N0 >/(amod)/ {}=N2
Cond_3 : {}=N0 >/nmod:(of)/ {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_isPresentIn
Cond_1 : {lemma:/(present)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_PresentIn
Cond_1 : {lemma:/(present)/}=N0
Cond_2 : {}=N1 >/(amod)/ {}=N0
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_labeled
Cond_1 : {word:/(labeled)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/(dobj)/ {}=N2
Action_1 : N0 >> argProtein >> N1

Action_2 : N0 >> argLocation >> N2

RuleID : pl_recruitmentOfTo
Cond_1 : {lemma:/(recruitment)/}=N0
Cond_2 : {}=N0 >/nmod:(of)/ {}=N1
Cond_3 : {}=N0 >/nmod:(to)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_recruitmentOf
Cond_1 : {lemma:/(recruitment)/}=N0
Cond_2 : {}=N0 >/(compound)/ {}=N2
Cond_3 : {}=N0 >/nmod:(of)/ {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_recruitTo
Cond_0 : {lemma:/(recruit)/}=N0
Cond_1 : {}=N0 >dobj {}=N1
Cond_2 : {}=N0 >/nmod:(to)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_protein_adj_localization
Cond_1 : {lemma:/(localization|colocalization)/}=N0
Cond_2 : {}=N0 >amod {}=N2
Cond_3 : {}=N0 >compound {}=N1
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_actionIn
Cond_1 : {pos:/(VB|VBD|VBG|VBN)/}=N0
Cond_2 : {}=N0 >/nsubj/ {}=N1
Cond_3 : {}=N0 >/dobj/ {}=N2
Cond_4 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N3
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argProtein >> N2

Action_3 : N0 >> argLocation >> N3

RuleID : pl_location_NN_of
Cond_1 : {pos:/(NN)/}=N0
Cond_1 : {}=N0 >/(amod|compound)/ {}=N1
Cond_2 : {}=N0 >/nmod:(of)/ {}=N2
Action_1 : N0 >> argProtein >> N2
Action_2 : N0 >> argLocation >> N1

RuleID : pl_JJin
Cond_1 : {pos:/(JJ)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|to|from)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_actionIn
Cond_1 : {pos:/(VB|VBD|VBG|VBN)/}=N0
Cond_2 : {pos:/(NN)/}=N1
Cond_3 : {}=N0 >/dobj/ {}=N1
Cond_4 : {}=N1 >/nmod:of/ {}=N2
Cond_5 : {}=N0 >/nmod:(in|at|on|throughout|within)/ {}=N3
Action_1 : N0 >> argProtein >> N2
Action_2 : N0 >> argLocation >> N3

RuleID : pl_VBNTo
Cond_1 : {pos:/(VBN)/}=N0
Cond_2 : {}=N0 >/(nsubjpass)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|to|from|over)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

RuleID : pl_protein_NN_on
Cond_1 : {pos:/(NN)/}=N0
Cond_1 : {}=N0 >amod {}=N1
Cond_2 : {}=N0 >/nmod:(in|at|on|to|throughout|within)/ {}=N2
Action_1 : N0 >> argProtein >> N1

Action_2 : N0 >> argLocation >> N2

RuleID : pl_verbIn
Cond_1 : {pos:/(VBZ|VBP)/}=N0
Cond_2 : {}=N0 >/(nsubj)/ {}=N1
Cond_3 : {}=N0 >/nmod:(in|at|on|throughout|within|to|from|over)/ {}=N2
Action_1 : N0 >> argProtein >> N1
Action_2 : N0 >> argLocation >> N2

Appendix E

REPRINT PERMISSION LETTER