Automated Translation Framework for Biological Event Annotation

by

Difei Tang

Bachelor of Engineering, Shanghai University, 2016

Submitted to the Graduate Faculty of the

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science in Electrical and Computer Engineering

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Difei Tang

It was defended on

April 12, 2023

and approved by

Murat Akcakaya, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, Ph.D., Professor, Department of Electrical and Computer Engineering

Thesis Advisor: Natasa Miskov-Zivanov, Ph.D., Assistant Professor, Department of Electrical and Computer Engineering

Automated Translation Framework for Biological Event Annotation

Difei Tang, M.S.

University of Pittsburgh, 2023

Computational modeling improves our understanding of the components and dynamics of biological systems. However, current biological models are restricted in size and scope due to limitations resulting from manual curation and validation. While natural language processing (NLP) techniques now have the ability to capture detailed semantics by event extraction from large amounts of rapidly processed text, a gap still exists between these NLP event representations and modeling formalisms. If the events contained in the natural language of published biomedical literature, related scientific articles, could be extracted and translated accurately into generic representation of biological knowledge, the impact on computational modeling and analysis of complex biological systems would be significant.

In this thesis, we develop a standardized framework for translating the events found in semantic NLP event annotations. To capture complex event structures, especially nested events in which one event causes another, we first present an intermediate graph representation. Our framework then enables the extraction of causal interactions between biological entities by defining a set of translation rules. We also develop a SBML-compatible network format for creating reaction models from events. With this network, we extend the existing methods to facilitate the conversion from events to SBML reaction model. Here, we briefly introduce our approach and investigate how well event annotations can be translated into other representation formats without incorporating information from any external resources. We demonstrate the effectiveness of our framework by the automatic translation of selected event annotations. By

standardizing the translation of events from NLP extractions, we propose this as a generalizable, scalable method for rapid, large-scale integration of knowledge on biological events extracted from literature.

Table of Contents

List of Tables

List of Figures

Preface

First and foremost, I want to express my sincere gratitude and thanks to my supervisor, Prof. Natasa Miskov-Zivanov for her guidance and unwavering support. I enjoyed working with her and being her student. I would like to extend my thanks to Prof. Murat Akcakaya and Prof. Zhi-Hong Mao for their support and valuable feedbacks on my thesis. I also would like to express my appreciation to my parents. I could not have accomplished my studies without their support. I would like to thank Cici Liu who have supported me emotionally. I would like to express my thanks to all the lab members and my friend who have helped my during this academic journey.

1.0 Introduction

The most efficient way for researchers to summarize and distribute their findings is their own natural language. In the biomedical domain, scientific literature plays an important role in disseminating new knowledge. Information extraction from those publications can help the scientists to find, organize and analyze the biological information in their research field. This includes biological event extraction from scientific texts for describing biological processes, functions, and mechanisms. Furthermore, natural language processing (NLP) techniques are applied to automate the process of event extraction from literature, and the development of such tools facilitates the mining of biological events that can be used for further applications, such as network construction and pathway curation.

In recent years, an event annotation is introduced in the GENIA corpus [1] to advance the development of event extraction systems. This annotation is more complex than relatively simple term annotation which only focuses on identifying biological entities. In event annotation, an event is a basic unit that consists of biological entities, such as genes, proteins and complex, as its participants and event trigger. An event trigger can be either verbal form (e.g., “activate”) or nominalization of a verb (e.g., “phosphorylation”) that specifies this event. The relationships between an event and its participants are indicated by the roles of events. The participants of an event are also called event arguments, and Theme and Cause are two common roles of the event arguments with the given definitions below:

Theme: the entity and event affected by the current event.

Cause: the entity or event that causes the current event to occur.

An event is defined as a nested event when it has other events in its arguments [2]. In contrast, a flat event only has entities in its arguments. We measure the level k of nested events using the definition proposed in [3], where $k = 1$ indicates the event has flat events in its arguments and $k = 2$ means that there is at last one nested event at level 1 in the event's arguments. In Figure 1, the "Positive_regulation" event (E3) is a nested event at level 1 with a "Gene_expression" event (E4) as its Theme participant.

Molecular interactions network can explain complex biological processes, and knowledge of molecular events is contained in scientific literature that could be helpful for network construction. In details, biological events can either describe the state change of a single entity such as a gene or protein, or indicate the participants involved in a binding process. Moreover, researchers often use different modeling approaches to specify the networks into executable models and then analyze their dynamic properties. Traditionally, biological models are often entirely constructed and parameterized by hand from the information in literature, leading to limitations in scope and consistency [4]. However, due to the exponential growth of scientific literature in biomedical domains, it is impractical to organize the extracted information manually for large biological systems. Consequently, BioNLP community has proposed a series of tasks in order to leverage the information from event extraction corpora. For example, the pathway curation (PC) task [5] is designed to evaluate the NLP extraction systems on supporting the curation and construction of pathway models. A pathway model is a graphical representation of a biological system that is composed of interrelated events, which indicates that events can be transformed into reactions of a pathway model. Another type of extraction systems called machine readers is developed in the Big Mechanism Project [6] that aims to extract the casual mechanisms through reading and assemble them into models of complicated systems.

Even though NLP systems perform well in specific BioNLP tasks, there is still a gap between natural language and other representations of biological systems. One of the obstacles is that the application is hampered by different representation formats across NLP systems. Also, models vary widely in their abstraction of biological systems that may require different levels of information. Thus, it is necessary to be able to translate the extracted information automatically from literature into a comprehensive and re-useable format. However, to the author's best knowledge, very few publications can be found to address the issue of bridging the gap from NLP event annotations to other biological representation formats. INDRA (Integrated Network and Dynamical Reasoning Assembler) is an example of a comprehensive system developed to automate model creation from text input [7]. It utilizes domain specific "statements" to represent detailed information about interactions that are extracted from scientific papers by machine reading systems. For biological applications, dozens of unique statement classes are instantiated with attributes such as location, mutation, residue. Then, these statements are integrated with default parameters for model assembly. The limitation of INDRA includes the lack of complexity in its statements, as they only construct the models using simple declarative sentences rather than raw text from literature. As INDRA statements are designed to represent biochemical mechanisms from multiple sources, these statements must be extremely detailed and restrictive in capturing the information of complex events (e.g., nested events with high level). Another study [8] proposes graph analysis methods to measure the discrepancy between human curated pathway maps and event representations, which is also the output of NLP extraction systems. In this study, the authors use a software system [9] to convert NLP event representations to SBML, and then evaluates the generated pathway with curated one. The evaluation is performed on mTOR pathway, and this study highlights the challenges of using NLP systems to automatically construct pathway maps.

In [10], the authors utilize a rule set to translate the events to the exact Biological Expression Language (BEL) syntax. Proposed conversion methods implement a direct mapping algorithm from event representations to BEL statements with restrictions on defined annotation types. A common limitation for these tools is their inability to translate intricate event structures. Even though another work proposed in [11] presents methods for capturing the biological nested events, all mentioned nested events have a level no greater than 1, hence it is insufficient to prove their capability to handle complex event structure. Except for INDRA, other tools are limited by their functionalities due to the fixed input and output formats.

To better illustrate our methods of translation, we use the definitions of continuants and instances introduced in [12], which refer a biological entity to an instance of a continuant. In [1], the authors conclude that there are two significant differences between pathway representations and event annotation. First, pathway representation is entity-centered, while natural language usually organizes information in a predicate-centered manner. This means that in pathway representations, entities are bound to continuants in specific biological context and the state changes of each continuant are explicitly depicted in the pathway. In other words, it is challenging for natural language to distinguish between instances of the same continuants in a biological context. Secondly, pathway organizes a sequence of biological entities in a network, where an entity in the upstream is linked by other entities in the downstream. Also, biological events are intertwined with each other in a pathway, which makes it hard to determine the causation. However, the representation of general causality in event annotation allows both events and entities to interact with their own and other types through causal relationship. These causal relationships can be thought of as links between events and their adjacent nodes. In event

annotation, a set of schemes has been defined to represent the general causality in biological events.

The primary purpose of this study is to develop a generic way to represent the knowledge about biological events, and explore how to translate events with complex structure and integrate them into a network representation. The results of the translation can be transformed into several forms commonly used by the computational and systems biology community. It is worth noting that in this thesis, we do not take other annotations (e.g., causality annotation, meta-knowledge) into account, since our goal is to develop a generic representation format for event extraction. The main corpus annotations we are referring to are GENIA corpus [13] and pathway curation annotation (BioNLP PC 13) [5].

Here, we developed a framework for the translation of event annotations from literature by introducing an intermediate graphical representation of biological events. We also defined a set of rules for generating this graph and for the extraction of causal interactions between biological entities. We designed a reaction network format to create a reaction model and integrated the existing mapping algorithms into our pipeline. Our translation framework supports several event annotation formats as its input and can also output different model formats. Using the algorithms outlined in this thesis, we demonstrate the possibility of translating event annotations into different types of representations with the minimum loss of information. The output of our translation framework can be readily used to assemble large-scale executable models.

In Chapter 2, we describe the background of event annotations and model representation formats as the input and output of our translation. We also introduce the element-based model and Biological system Representation for Evaluation, Curation, Interoperability, Preserving, and Execution (BioRECIPE) format. In Chapter 3, we introduce the methodology details for two

intermediate representations, Event Annotation Graph and Reach Network. We also propose the mapping algorithms from events to these forms. We define a set of rules for assembling element-based models from the reaction network. In Chapter 4, we evaluate our translation framework using a polarity dataset and a mammalian target of rapamycin (mTOR) pathway events corpus.

2.0 Background

2.1 NLP Event Representations

Since we are interested in the translation of biological events and developing our interface based on relevant event representations as the input, we mainly focus on three event representation formats. The first representation that we use here is the *standoff representation* proposed for BioNLP Shared Task 11 [1]. The following is an example of standoff-style annotation:

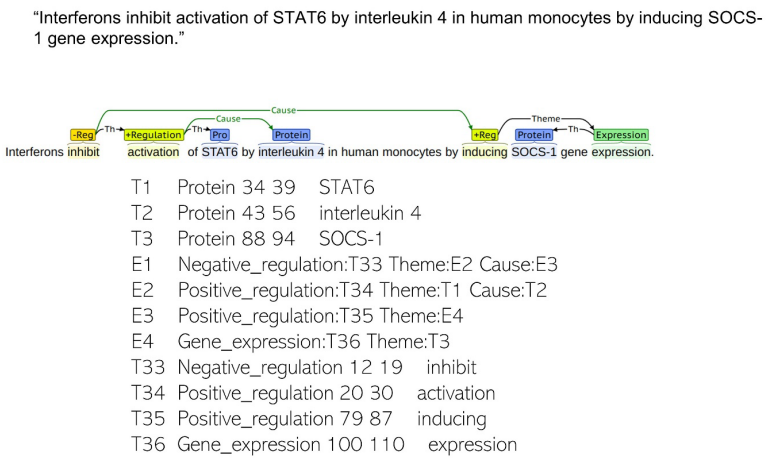


Figure 1: Example sentence with standoff representation

The event and other semantic information are represented in a standoff-style annotation as shown in Figure 1. The annotation starting with "T" denotes the entity, which is followed by its type and character offsets. Annotation starting with "E" stands for an event, which is identified by a trigger word. For example, E1 is identified by type "Negative_regulation" that refers to an event trigger, T2 ("inhibit"). The arguments of E1 can be found by their roles, "Theme" and "Cause". A

key feature of event annotation is the capability to annotate events as participants in other events, creating complex event structures. In this example, the participants of event E1 are events E2 and E3. In Figure 1, this sentence is visualized using a web-based tool named brat [15], and concepts including events and entities are linked by semantic roles, Cause and Theme.

The other two representation formats that we use come from two *machine reading systems*, TRIPS/DRUM [16] and REACH [17] respectively. TRIPS is a general language processing system and DRUM is a customized version of TRIPS used in the biological domain. In TRIPS/DRUM, a logical form (LF) is used for extracting events and relationships, and the extracted content is then transformed into a graph-based representation called extraction knowledge base (EKB). The REACH reading engine, on the other hand, defines a set of rules to extract events from text and the output of a rule is termed as a "mention", which is converted into a series-JSON representation. The TRIPS and REACH representation formats differ slightly from one another, but all of them are capable of identifying biological events, as well as representing the relationships with semantic roles. For instance, Figure 2 illustrates the same extraction output using REACH reading system as the example of standoff annotation we mentioned above. As an alternative, REACH uses *Controller* and *Controlled* to represent that one event or entity is the cause of another. The extraction result using REACH is not consistent with the manual annotation, which indicates that NLP extraction systems will output some number of erroneous annotations.

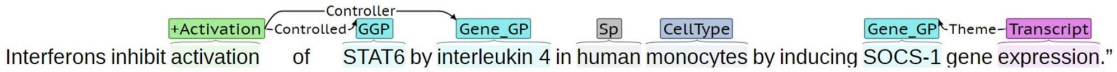


Figure 2: Example sentence with REACH annotations

Event structure in BioNLP Shared Task could be more complex because the event-related corpora in Shared Tasks are usually annotated by specific domain experts and serve as sources for the development of advanced NLP extraction systems. However, the BioNLP Shared Task provides no normalization (linking entity mentions in text to concepts of a reference vocabulary) of the entities, while both REACH and TRIPS/DURM have their own ontology database and named entity recognition (NER) component for adding information from many external ontology and vocabulary resources.

2.2 Model Representation Formats

Biological networks can be represented using different formats. Here, a reaction-based model which is composed of networks of biochemical reactions and their involved species is a format of the output that our framework will produce. In a reaction-based model, the concentration of each species is measured by reaction rates and other kinetic parameters over time. These machine-readable model formats include the Systems Biology Markup Language (SBML) [18] and Biological Pathway Exchange (BioPAX) [19]. SBML is an XML-based data format used for describing mathematical models of biological systems. BioPAX is an RDF/OWL-based standard language for describing the components and interactions of biological pathway data. Besides, Biological Expression Language (BEL) is a language designed for representing qualitative causal relationships between biological entities, including genes, proteins and small molecules, as well as the mechanistic details. The first two languages (SBML, BioPAX) focus on pathway construction which require more information, while BEL enables the systematical integration of

knowledge from publications and is particularly useful to construct knowledge networks that contain qualitative causal relationships between biological entities.

There are several tools that provide visual representations of biological networks modeling, such as CellDesigner [20] and SBMLDiagrams [21]. In CellDesigner, networks can be drawn in a process diagram with a set of defined symbols and graphical notations. The notations include general components such as Species (including Complex), Reaction and Compartment. As shown in Figure 3, components are linked by arrows which represent the state change of molecule, and we will use these notation schemes to display the reactions from event translation.

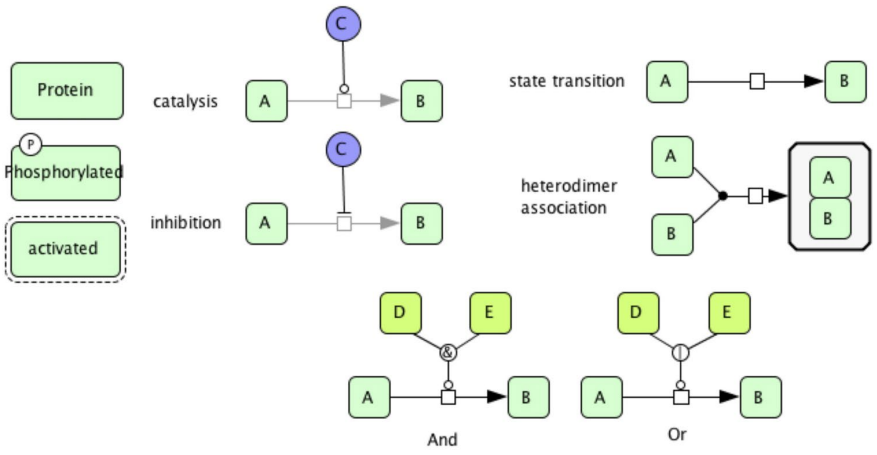


Figure 3: The CellDesigner graphical notation schemes for network representations. These notation schemes are used to illustrate the reaction model and our translation rules in section 3.6.

2.3 Hybrid Element-Based Models

While the machine reading output is usually a list of individually extracted events and causal relationships, complex systems consist of many connected components and different combinations of influences between them. Therefore, hybrid element-based models are well suited for representing and analyzing such complex systems. These models can be defined as a pair $M(S, F)$, where $S = \{s_1, s_2, \dots, s_n\}$ is a set of n model elements, each element representing a component of the complex system, and $F = \{f_1, f_2, \dots, f_n\}$ is a set of n regulatory (update) functions such that each element $s_i \in S$ has a corresponding function $f_i(s_1, s_2, \dots, s_n) \in F$. Using element update functions, one can determine changes of model elements in time, for example, through simulations. Therefore, we also refer to these models as executable models. In general, functions in F can be of different types (e.g., from discrete to continuous, thus, hybrid models), depending on the knowledge available about the complex system and its components. For each element s_i, the set of elements that the function f_i is sensitive to is called an influence set of s_i, and each element in that set is referred to as a regulator of s_i [22], [23]. The underlying structure of element-based models is a directed graph, where nodes are model elements, and directed edges between nodes represent regulatory influences between elements. Besides direction, different types of directed edges are used to indicate the sign of influence (positive or negative). While such graphs, also referred to as influence networks, are beneficial for visualizing complex system networks, usually they do not hold all the information necessary to determine update functions.

Recently, a tabular representation format named Biological system Representation for Evaluation, Curation, Interoperability, Preserving, and Execution (BioRECIPE) [24] was proposed to capture all the relevant information in element-based models of intra- and intercellular signaling

networks [25]. Attributes in that format are specific to the biology domain, accounting for the information element types (e.g., proteins, genes, chemicals, biological processes), cellular locations of reactions (e.g., cell membrane, cytoplasm, nucleus), or cell and tissue types (e.g., immune cells, pancreas, etc.). In this format, each element is assigned with an initial discrete value, and the changes of the model elements are determined by their logical update functions in time. For a regulated element, the logical operators are used to combine all regulators. In Section 3.6, we will discuss this representation format and how we translate the event annotations into that format.

3.0 Methodology

We show in Figure 4 an overview of our translation framework and some key steps in our methodology to map event extraction to executable models. Here, we propose two intermediate representation formats:

- Event annotation graph (EAG) - a semantic graph-based representation format for event annotation.
- Reaction network (RN) - a SBML-compatible network for representing the reactions. This network is compatible with the specifications of SBML package and can be exported to pure SBML files. Mapping algorithms are also performed for translating events to this format. In the last assembly step, we develop a method to assemble element-based models from the generated reaction network.

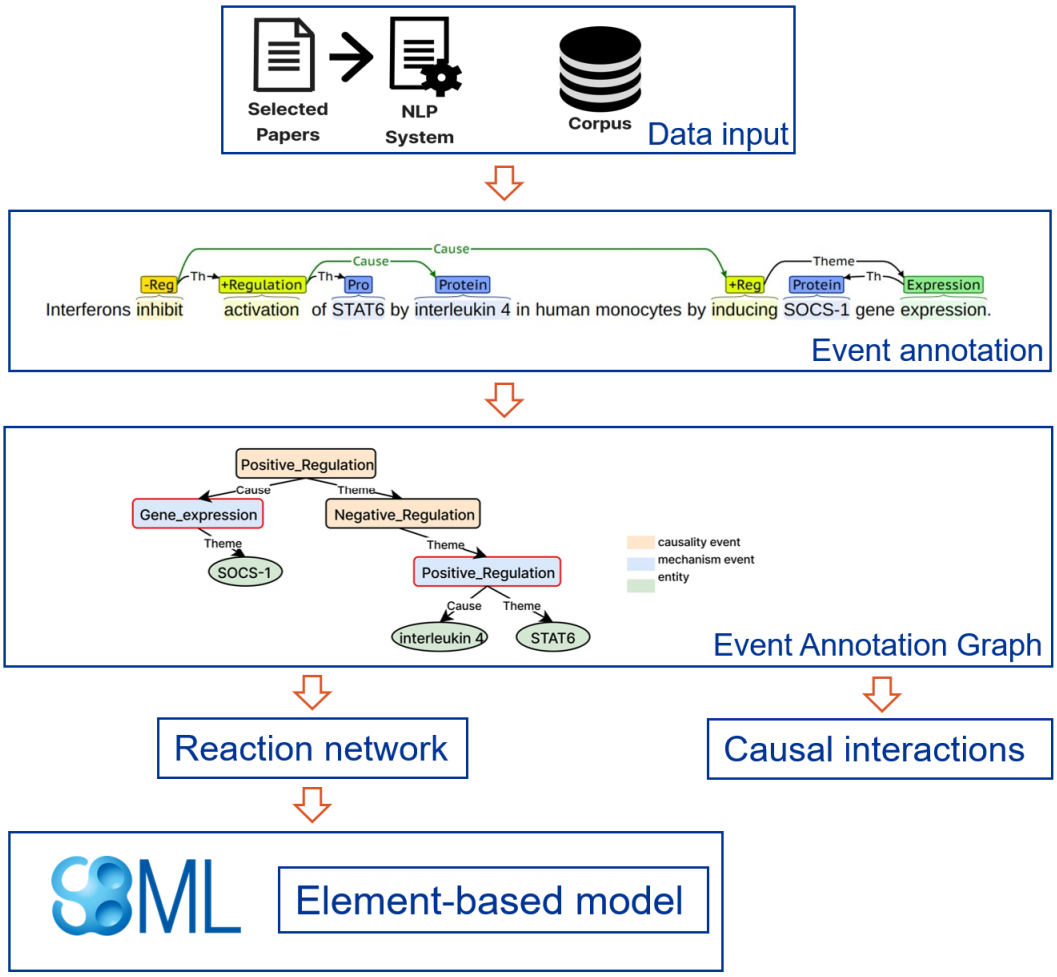


Figure 4: Overview of our translation framework

3.1 Construction of Event Annotation Graph

Each event-based extraction system uses a subset of semantic definitions, often unique to the annotation format. As mentioned above, many NLP extraction systems (e.g., TEEs [26], EventMine [27], DeepEventMine [3]) are developed based on the BioNLP Shared Task annotations, while machine reading systems have their own ontology and event annotation

formats. Therefore, we propose here a graph-based representation format, Event Annotation Graph (EAG) that can capture key terms and semantics among NLP extraction systems. Entity and event nodes, which are used by all extraction systems studied in this work, are incorporated directly into this form. Therefore, the first step in our framework is to translate event annotation into EAG. Given an event annotation, we can represent it using a weighted undirected graph $G = (\mathrm{V}, \mathrm{E}, \mathrm{L_v}, \mathrm{W_e})$. The graph consists of a set of nodes $V = \{v_1, v_2, \ldots, v_n\}$ that represent the events $E = \{e_1, e_2, \ldots, e_m\}$ and entities $N = \{n_1, n_2, \ldots, n_m\}$, while events and their participants are linked by edges. L_v is a set of node labels that are used to represent the attributes of entity and event, such as type and character offset. Each element in the set W_e is the weight of an edge, which represents the semantic role of an event argument, such as Theme and Cause. Thus, an EAG can be constructed using the events and entities in the example sentence (Figure 1), while the other annotation of this sentence (Figure 2) will have several unconnected subgraphs during the construction step.

In order to capture the knowledge of mechanisms in the form of event annotation, we divide the events into different subtypes in EAG. If an entity is a Theme argument of an event, we identify this event as a *mechanism event* (me). One exception is the dissociation event which has an entity as its Product argument. In standoff format, usually non-regulation events can be differentiated as me, and regulation events are known to pose challenges for extraction in involving other events as arguments, thus creating nested event structures. For such complex event structure, in the example sentence, the verb "inhibit", which is the trigger word of a nested event (E1), is taken to express a causal relationship (negative regulation) between the two events (inducing gene expression (E3) and activation of STAT6 (E2)). This type of event that expresses the causal relationships is notated as *causality event* (ce). Moreover, BioNLP Shared Task annotations

include equivalence relations that will be discussed later and event modifications as extraction targets, Negation and Speculation. TRIPS and REACH readers also have their own in-event structure to represent these modifications. Usually, these modifications will mark events as being explicitly negated or stated in a speculative context. In our framework, these events or annotations are regarded as *event modifications* (em).

Assuming accurate event extractions from sentences, the EAG would look like the graph in Figure 5 with given complex biological statements: "Interferons inhibit activation of STAT6 by interleukin 4 in human monocytes by inducing SOCS-1 gene expression." Two events (activation of STAT6 (E2) and SOCS-1 gene expression (E4)) which explicitly involve Theme entities STAT6 and SOCS-1 are mechanism events. Nested events E1 and E2 are causality events in this generated EAG.

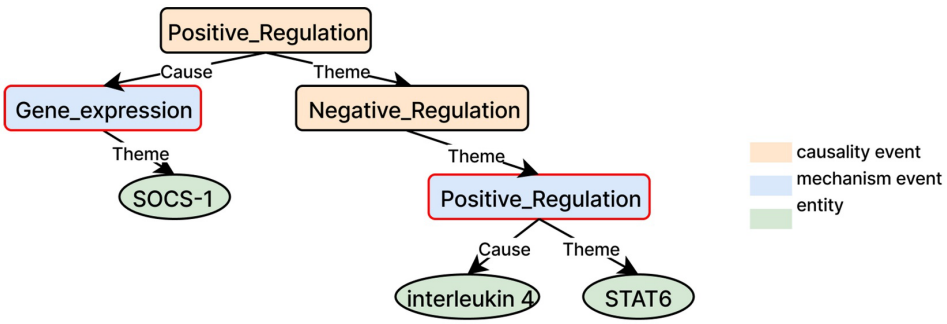


Figure 5: Event annotation graph (EAG) extracted from example sentence.

3.2 Event Translations

As the representation of biological model is entity-centered, entities are defined as primitives and events need to be handled with caution to represent the state or value changes of entities. In this section, we will introduce our methods for event translations.

Table 1: Algorithm for extraction of Causal relationship paths

ALGORITHM 1: EXTRACTION OF CAUSAL RELATIONSHIP PATHS (CRPS)

```
H ← subgraph(EAG)
CRPs: new list for causal relationship paths
entityPairs = permutation(H, 2) # a list of entity pairs
For entityA, entityB in entityPairs do:
    For path in find_all_simple_path(H, entityA, entityB) do:
        If entity is Theme of event do:
            event ← mechanism event
        Else: event ← casuality event
        remove path if numEntity(path) > 2
        remove path if checkDirection(path) is wrong
        remove path using chain subtitution
        append path to CRPs
```

3.2.1 Extraction of Causal Interactions

Causal interactions are informative in the assembly of large biomolecular networks, and essentially event annotation can be regarded as a combination of named entity recognition and relation extraction. To this end, the second step is the extraction of causal mechanisms and to obtain the dynamic relationships between entities, therefore it is not essential that all of the more specialized thematic roles are captured. Based on the definitions of Theme and Cause roles, we retain the edges that contain these attributes along with their connected nodes, resulting in a subgraph H (W, F) of EAG. For other roles, only the Product entity of dissociation event remains in this subgraph. A *causal relationship path* (CRP) is then defined to represent the causal

relationship between two entities within this subgraph. On the other hand, since CRP is designed to capture the causal interactions, i.e., interactions where a source entity has an effect (up-regulation, down-regulation, etc.) on a target entity, it is essential to identify the target entity and the effect on this entity. Therefore, we develop an algorithm in Table 1 to extract every causal relationship path in EAG. For this extraction algorithm, we first find all the simple paths between all the possible entity pairs (S, T). In each path, if an entity is the Theme participant of an event, the event will be recognized as a mechanism event and other events are causality events. We remove paths containing more than two entities. We check the edges in each path from source entity to target entity to make sure Cause comes before Theme in the path. For each target entity, we use the chaining substitution to identify its source entity. The second last node in the path must be the mechanism event of the target entity. As a result, two causal relationship paths (CRPs) can be extracted from the example sentence shown in Figure 6.

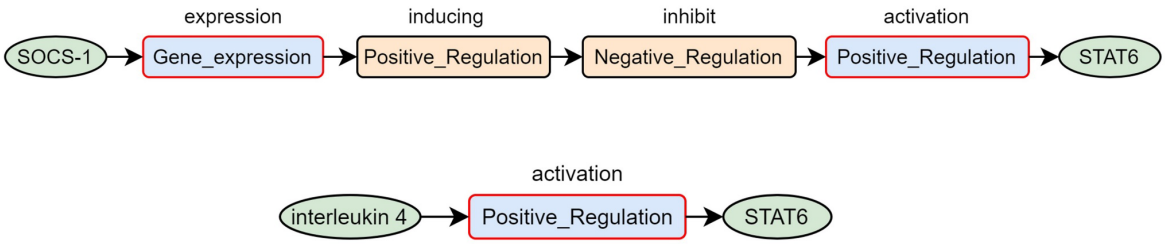


Figure 6: Causal relationship paths (CRPs) extracted from example sentence

3.2.2 Reduction of Causal Relationship Paths to Simple Event Structures

As mentioned before, we consider the events that express causal relationship as the causality event. For each CRP, path reduction is performed to assign different subtypes to events.

The reasoning behind this reduction step is that pathway representations or executable models are often hard to represent these causality events. Furthermore, we find that due to the complexity and ambiguity of natural language, even in the state-of-the-art nested event extraction systems such as DeepEventMine, an event in one sentence might be identified as both mechanism event and causality event using our methods. Therefore, a casualty event is relative to a specific CRP, which means in another CRP, this event could be notated as another subtype of event.

Intuitively, mechanism event is treated as our entry for the downstream translation, and keeping only the mechanism events is what makes the most sense from a systematic perspective, as a target entity must be affected by its related mechanism event in annotation. Thus, we define

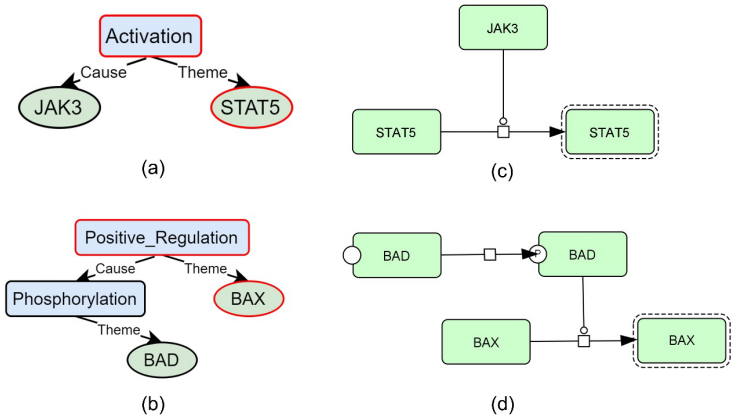


Figure 7: (a) The first simple event structure (b) The second simple event structure (c, d) Translated reactions from simple event structures

two simple event structures in this work. In Figure 7(a), the first expression "STAT5 is activated by JAK3" and verb "activated" is annotated as a flat event, we can simply translate this structure into the source entity ("JAK3") influences the target one ("STAT5"). In the other event structure "The phosphorylation of BAD causes an increase of BAX" in Figure 7(b), the entry event ("increase of BAX") is a nested event that has another mechanism event ("phosphorylation of

BAD") in its arguments. In this case, we denote these two events as source mechanism event (me_{source}) and target mechanism event (me_{target}) respectively. From this sentence, we can also extract a causal interaction that BAD positively regulates the target entity BAX. These event structures can be written as paths (n_{source}, me_{target}, n_{target}) and (n_{source}, me_{source}, me_{target}, n_{target}). Causality events in every CRP will be removed and the final sign of this interaction will be considered by combining the polarity of both causality and mechanism events. Event modifications will be added to the attributes of its mechanism event. Finally, every CRP extracted from EAG will be reduced to one of the simple event structures.

For the example sentence, consider the sentence text and entity pair (n_i, n_j):

S: "Interferons inhibit activation of STAT6 by interleukin 4 in human monocytes by inducing SOCS-1 gene expression.", (n_i, n_j): ("SOCS-1", "STAT6")

The original causal relationship path extracted from S with its event annotation graph EAG_s (Figure 5) is denoted as $CRP_{original}$: ("SOCS-1", "expression", "inducing", "inhibit", "activation", "STAT6")

Finally, the causal relationship path is reduced to the second event structure represented as $CRP_{reduced}$: ("SOCS-1", "expression", "activation", "STAT6"), along with a list of removed causality events CRP_{ce}: ("inducing", "inhibit")

3.3 Syntactic Mapping from EAG to Causal Interactions

As previous sections describe the methods of extracting causal interactions in EAG, we could translate complex events into multiple CRPs. There have been several representation formats developed for causal interactions. For example, SIGNOR database [28] introduces a structured format for the storage and analysis of causal relationships with human curation. A tabular format is proposed in CLARINET [29] to describe the directed signed interaction between regulator and regulated entity from machine reading output. In this thesis, we extend this tabular format with the curation manual from SIGNOR to represent the causal interactions. For the SIGNOR curation, curators will provide information about the effect (positive or negative) that the regulatory entity has on a regulated entity. In the following section, we will show our methods of mapping from EAG representations to causal interactions format.

In biological interactions, polarity attribute (the sign of interactions) is important in understanding the effects on target entity. The work in [30] analyzes the identification of polarity with some examples. For the statements: “The inactivation of Bad is sufficient to antagonize p38 MAPK”, REACH reading engine would extract a negative interaction between these two entities: (“Bad”, “antagonize: Negative_regulation”, “P38 MAPK”), which leads to a mischaracterization of polarity (e.g., activation or inhibition). Another extraction system (DeepEventMine) based on deep learning model is able to extract this interaction with nested events: (“Bad”, “inactivation: Negative_regulation”, “sufficient: Positive_regulation”, “antagonize: Negative_regulation”, “p38 MAPK”). This situation is hampered by the fact that statements might include interconnected events that affect the polarity.

Table 2: Categories for event annotation types

Positive Regulation Event	Regulation, Positive_regulation, Activation, Phosphorylation, Acetylation, Methylation, Deubiquitination, Catalysis
Negative Regulation Event	Negative_regulation, Inactivation, Dephosphorylation, Deacetylation, Demethylation, ubiquitination
Expression Event	Gene_expression, Transcription, Translation
Degradation Event	Degradation, Catabolism, Protein_catabolism
Localization Event	Localization, Transport
Association Event	Binding, Dissociation
Other Event	Conversion, Pathway

In order to detect the polarity of interactions, we classify the events into different categories and the polarity is assigned according to Table 2. For example, the class of Regulation with its subclass Positive_regulation is assumed as positive regulation events. Apart from negative regulation events, we assume that only the degradation event is negative, while the other categories are positive when considering the polarity of interactions. Furthermore, if an event modification is found, we will consider the polarity of an event with its event modification. If an event modification is negation, we will reverse the polarity of the event. If an event modification is speculation, we will annotate the extracted interaction as a hypothesis.

Table 3: Mapping rules from CRPs to Effect in causal interactions

Effect	Mapping Rules
up-regulates	$+CRP_{interaction}$
up-regulates activity	$+CRP_{regulation}$, me_{target} is Positive Regulation Event
up-regulates quantity by expression	$+CRP_{regulation}$, me_{target} is Expression Event
up-regulates quantity by stabilization	$-CRP_{regulation}$, me_{target} is Degradation Event
down-regulates	$-CRP_{interaction}$
down-regulates activity	$-CRP_{regulation}$, me_{target} is Positive Regulation Event
down-regulates quantity by repression	$-CRP_{regulation}$, me_{target} is Expression Event
down-regulates quantity by destabilization	$+CRP_{regulation}$, me_{target} is Degradation Event
form complex	$+CRP_{regulation}$, me_{target} is Association Event

We summarize the mapping rules for effect in Table 3. As mentioned before, a CRP will be reduced to a simple event structure, which can be written as: (t_{source}, (me_{source}), me_{target}, t_{target}). We count the number of events with their polarity along this path, and we represent the polarity using the symbols (+, -). Taking into account the polarity of removed causality events CRP_{ce}, the polarity of regulation $CRP_{regulation}$ is decided by the polarity of source mechanism events (me_{source}) and removed causality events (CRP_{ce}), while for the polarity of interaction $CRP_{interaction}$, we need to compute the polarity of all the events involved in CRP. In the following examples we show how we conduct a semantic interpretation of the sentences and translate them into causal interactions by their CRPs. Table 4 illustrates the results of generated interactions output.

Example 1:

S: "Interferons inhibit activation of STAT6 by interleukin 4 in human monocytes by inducing SOCS-1 gene expression.", $CRP_{reduced1}$: ("SOCS-1", "expression", "activation", "STAT6"), CRP_{ce1}: ("inducing", "inhibit")

-$CRP_{interaction1}$: ("expression", "inducing", "inhibit", "activation"),

-$CRP_{regulation1}$: ("expression", "inducing", "inhibit"), $me_{target1}$ is Positive Regulation Event→ (effect: down-regulates activity, sign: negative)

$CRP_{reduced2}$: ("interleukin 4", "activation", "STAT6")

+$CRP_{interaction2}$: ("activation")→ (effect: up-regulates, sign: positive)

Example 2:

S: “The inactivation of Bad is sufficient to antagonize p38 MAPK.” $CRP_{reduced}$: (“Bad”, “inactivation”, “antagonize”, “p38 MAPK”), CRP_{ce}: (“sufficient”)

+$CRP_{interaction}$: (“inactivation”, “sufficient”, “antagonize”)→(effect: up-regulates, sign: positive)

Example 3:

S: “Expression of IkB-alpha inhibited the upregulation of VCAM-1 expression.” $CRP_{reduced}$: (“IkB-alpha”, “Expression”, “expression”, “VCAM-1”), CRP_{ce}: (“inhibited”, “upregulation”)

-$CRP_{interaction}$: (“Expression”, “inhibited”, “upregulation”, “expression”),

-$CRP_{regulation}$: (“inhibited”, “upregulation”), me_{target} is Expression Event→ (effect: down-regulates quantity by repression, sign: negative)

Table 4: Output of causal interactions for three example sentences

Name	Regulator Entity	Regulated Entity	Sign	Effect	Mechansim
Example 1	SOCS-1	STAT6	negative	down-regulates activity	Activation
Example 1	interleukin 4	STAT6	positive	up-regulates	Activation
Example 2	Bad	p38 MAPK	positive	up-regulates	Negative_regulation
Example 3	IkB-alpha	VCAM-1	negative	down-regulates quantity by repression	Gene_expression

3.4 SBML-Compatible Reaction Network

We developed a SBML-compatible reaction network which followed the specifications of SBML and reimplemented some core functions of python version of libSBML [31]. The structure of our network format was depicted in Figure 8. The rationale of the format relied on the needs of integrating biological events scattered in different literature into a coherent system. Furthermore, many SBML-based graphical editing tools, such as CellDesigner, had a more fine-grained view of species and reactions. Thus, we designed such a format as a bridge to facilitate a better conversion from events to reaction models. The main elements of this format were Species which represented the entities of the model, and Reactions that described the mechanisms changing the state of involved species. Here, apart from the id attribute that gave species and reaction a unique identifier in the model, a *uniqueAttributes* attribute was defined to add the properties before creating a species. For the mapping from biological entities to species, we could refer a species to an instance of the same continuant with this attrbute. An *aliasName* attribute was intended to be used for representing the equivalence relations from annotation. A ComplexSpecies was a subclass of Species which was used to represent complex entity. It had a *participants* attribute that contained its constituents. We also implemented the compartment class, and a species was associated with a compartment. Each reaction could be mapped from a biological event. We remained the same specifications for reactant and product references, while we added a *modificationType* attribute to ModifierSpeciesReference. This attribute was used to describe different control that modified the reactions. SBML did not have a specific definition for modifier, instead SBML used the *sboTerm* attribute that supported the terms from the Systems Biology Ontology (SBO). However, this *sboTerm* values were ignored in some models and during the format conversion, this information might be loss in some cases.

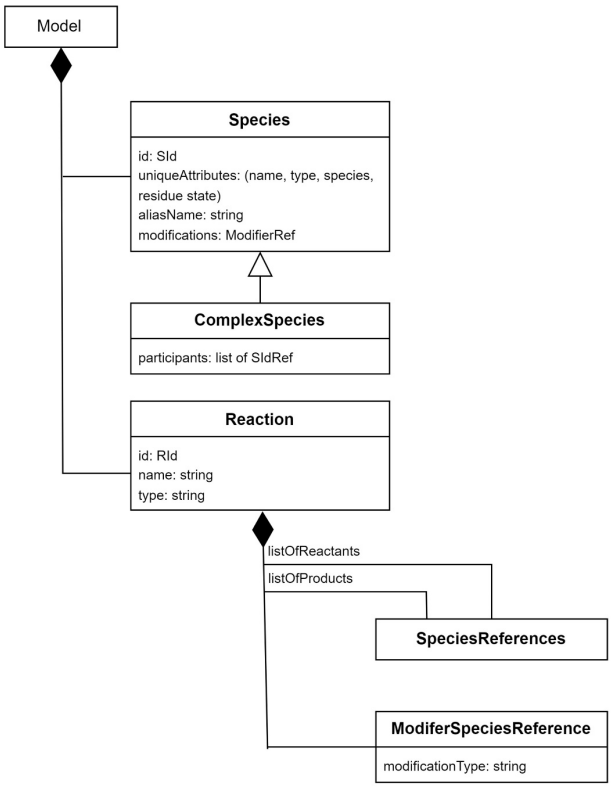


Figure 8: The structure of our SBML-compatible reaction network with fine-grained attributes

3.5 Mapping from EAG to Reaction Model

In the following subsections, we will introduce our mapping algorithms from EAG to our developed network format. This representation format can be used to generate output files in SBML format.

3.5.1 Mapping from Entities to Species

Table 5: Interpretation and comparison of state transitions

Pathway/Reaction Model	Event	
SBML/CellDesigner	GENIA	PC
in:Compartment$_1$ → in:Compartment$_2$	Localization	Localization
residue:state:∅ → residue:state:Phosphorylated	Phosphorylation	Phosphorylation
residue:state:Phosphorylated → residue:state:∅	Dephosphorylation	Dephosphorylation
residue:state:∅ → residue:state:Methylated	Methylation	-
residue:state:Methylated → residue:state:∅	Demethylation	-
residue:state:∅ → residue:state:Ubiquitinated	Ubiquitination	-
residue:state:Ubiquitinated → residue:state:∅	Deubiquitination	-
Species:state:inactive → species:state:active	Positive regulation, Activation	Positive regulation, Activation
species:state:active → Species:state:inactive	Negative regulation, Inactivation	Negative regulation, Inactivation

As we discussed before, biological entities in an EAG are related to instances in specific biological contexts, and events can be mapped to reactions that describe that state transition of species. However, natural language does not give us explicit distinctions among instances of the same continuant that corresponds to different biological contexts. Therefore, in Table 5, we use the state transitions map proposed in [32] to ensure that each biological entity can be mapped to a specific instance of continuant. For example, a reaction changing one residue state into Phosphorylated would map into the event, Phosphorylation. In other words, we could identify instances and create a list of species to represent them. Therefore, the first step of mapping algorithms is to initialize an empty reaction network and create species for each biological entity. In our SBML-compatible network format, a species is uniquely defined by its *uniqueAttirbutes* attribute: species name, species type, species state and its residue state. The name and type of entity annotation are mapped to species name and type. If an entity has a mechanism event to determine its state transition, then the specific state is added to distinguish between the species. Note that in a paragraph, entities across sentences and abstracts might be identified as the same instance, and

the number of species would be less than that of biological entities. In the reaction network, we create a unique *SId* for each species and map this species id to multiple annotation-IDs of the entity. For each entity, if there is no specified annotation for its compartment, we annotate the compartment as “default”.

3.5.2 Mapping from Events to Reactions

As we already reduce the CRPs in EAG to simple event structure, in this step reactant, product and modifier are added the reaction model on the roles of events. For the BioNLP Shared Task, Theme role is applied to capture reactants, and Product role is used for products. Cause role and regulatory events are defined to capture modifiers. For the mapping algorithm, we extend the existing mapping methods proposed in [9] by our representation formats. In Figure 7(c) and Figure 7(d), we show two examples of translated reactions from our simple event structures. The first example is a direct mapping from event annotation to a SBML-style reaction, while in the second example CRP (“BAD”, “phosphorylation: Phosphorylation”, “increase: Positive_Regulation”, “BAX”), the Cause is a source mechanism event (“Phosphorylation”) and the product of this event is used as a modifier. If the products and reactants for reactions are not explicitly annotated, we will automatically create the species for them. As noted in the previous subsection, we will create a phosphorylated BAD since the source mechanism event in CRP is a Phosphorylation event, and the species state of the target entity BAX is set to active. In other words, using our representation formats, only the mechanism events are mapped to reactions. Moreover, Localization Events and Expression Events are handled differently from other mechanism events. For the Localization Events, they will have the following additional roles to describe the compartment of involved entities.

AtLoc: location in which the Theme entity of a Localization event is localized. The Localization events are not involving movement. For this role, we create a compartment instance and associate the Theme entity with this compartment.

FromLoc/ToLoc: localization in which the Theme entity of a Localization event is transported from/to. This type of event (e.g, Transport) involves the movement of entity and we create a reaction where the Theme entity is initially located in the compartment described by FromLoc role (reactant), and ends up in the compartment described by ToLoc role (product).

For the Expression Events (e.g, Gene_expression, Transcription, Translation), when we map these events to reactions, they do not have the reactant species. In this work, we use the conversion methods proposed in [9]. For Transcription events, if the type of Theme entity is RNA, we map the entity to the product. If the type of Theme entity is DNA, then it gets mapped to the modifier of the Transcription reaction. In our network format, we create a *modificationType* for each modifier species and we use the polarity of $CRP_{regulation}$ to represent this attribute. For example, if the polarity of $CRP_{regulation}$ is positive, we annotate the modification type as catalysis. If the regulation type of CRP is negative, then we assume the modification type is inhibition. In Figure 9, we show the network format of Example 1 in section 3.3. Since there are two CRPs extracted from its EAG, a Positive_Regulation reaction is created that contains two modifiers SOCS-1 and interleukin 4. Modification types are decided by their corresponding $CRP_{regulation}$, and to complete the reactions in this example, we create a new STAT6 species (s3) with active state and set the reactants of Gene_expression event as empty.

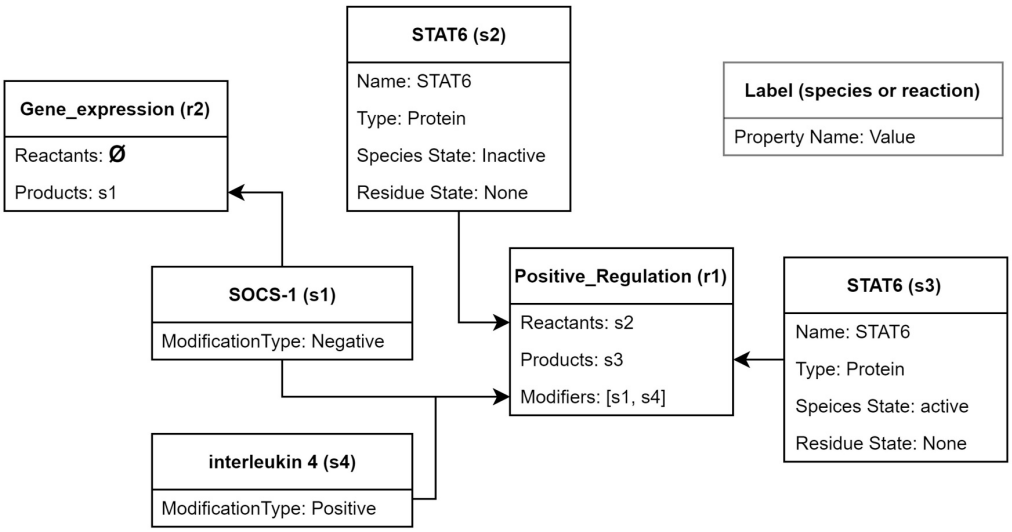


Figure 9: Reaction network of example 1 in section 3.3.

3.5.3 Reaction Network Contraction

For a large-scale corpus or multiple sentences translation, an additional step we term as network contraction is required since the events will be mapped to separate reactions without integrating into one. In the previous two steps we can translate the EAG into the SBML-compatible network format and species are uniquely identified by the *uniqueAttributes* and *participants* attributes. Moreover, in Genic Regulation Network inference algorithm [33], the resolution of interaction relations requires a traversal of the graph of annotations with rules to identify the agent and the target of an interaction relation. In our translation framework, the use of CRPs to detect the modifiers can avoid the traversal of the graph of annotations and extract the interaction relations with high accuracy. In this step, reactions with the same species are first simplified into one reaction. Then, we will remove the redundant edges between contracted event node and its Theme species. The final result is composed of big networks and relatively small connected components.

3.6 Translations from Reaction Network to Element-Based Model

In biological system studies, several modeling approaches can be applied during the process of model assembly. For example, ordinary differential equations can be used to model biochemical reactions when kinetic parameters are available. However, this type of model is limited in size due to the expense of describing mechanistic details. On the other hand, element-based models have been shown to efficiently simulate large systems, without the need for a complex parameterization process [4]. In this work, we aim to assemble an element-based model in the tabular format proposed in [25].

As we have already generated a reaction network from EAG, the aim is to convert this reaction network into a element-based model. Since our reaction network is compatible with SBML/CellDesigner and the element is regulated by logical update functions in the target format, we combine the methods in CaSQ [34] and the rules in [25] for our translation task. The translation is done in the following step:

Step 1: First, we modify the graph-rewriting rules in CaSQ to let them be compatible with the output tabular format. Most Species will be mapped to elements and some species need to be deleted. The network topology is computed as a directed graph, with one element corresponding to each species in the network.

Rule 1: For the state transition reaction, if a reactant and its product have the same species type, and if both the reactant and the product have the same name, then the reactant is deleted. When creating the species, we have already checked the uniqueness of instances using the *uniqueAttributes* and *participants* attributes.

Rule 2: If the reaction is annotated as association, and if one of the reactants is annotated as receptor, then the complex species is deleted, and the receptor is positively regulated by the other reactant (Figure 10).

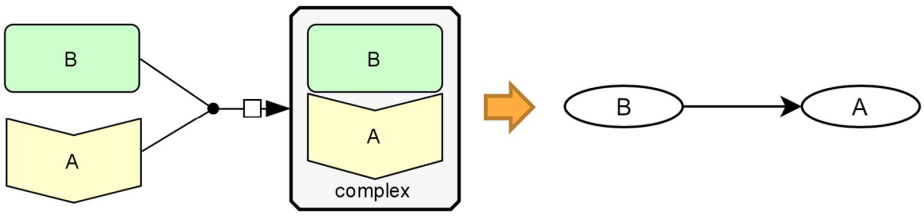


Figure 10: An Illustration of the rule 2. If two species are involved in a heterodimer association, and one of the reactants is a receptor, then the complex product is removed and the receptor is regulated by the other reactant.

Rule 3: If a reaction is annotated as association, if none of reactants are annotated as receptor, then the complex species of this reaction is deleted, and modifiers are rewired to have the reactants as the products. If the removed complex appears in another reaction, we list all the components of complex as regulator for the product of that reaction. Figure 11 illustrates the combination of rules 1 and 3. The formation of complex AB is catalyzed by C, and we create two rows: one for element A, and one for element B. The logical update function for element A is (C AND B), while the regulatory function for B is (C AND A). In addition, species D is regulated by the complex AB, then the positive regulation rule for D is (A AND B).

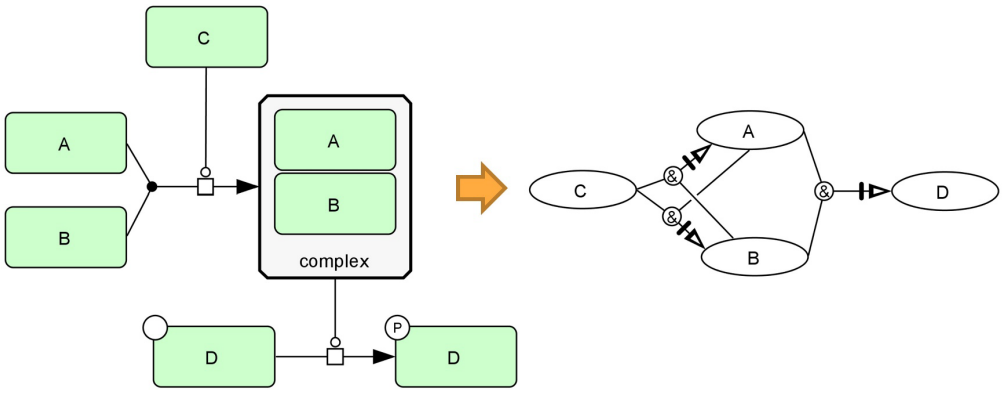


Figure 11: Combination of rule 1 and rule 3. Complex species is deleted and regulation of downstream elements is replaced by its components.

Rule 4: If a reaction is annotated as transport, we keep both the reactant and the product, and then the reactant is mapped as a positive regulator for the product.

Step 2: The logical update functions are computed based on the methods in CaSQ. For each element, we use the logical OR operator for all reactions producing it, and a reaction is on if all reactants are on, all inhibitors are off and one of the catalysts is on.

Step 3: As CaSQ suggests, model refinement is optional for the removal of unconnected components. For the names of element, we make them more precise by adding the original type of the species (e.g, Protein, RNA) and compartment (separated by an underscore).

The graph-rewriting rules are used to identify the same components in different states and decide when species can be merged or discarded. For the translation of element-based model, we replace the formation of complex species with its subcomponents. By the rules 2 and 4, we also retain components that might be involved in network motifs [35]. For instance, the case of translocation motif can be translated from the expression: “The binding of ligand X and receptor

Y regulates the translocation of Z from cytoplasm to nucleus". The reactant (Z_cyto) and product (Z_nuc) of the transport event are mapped to corresponding elements to model this process.

4.0 Results

4.1 Casual Interactions Extraction Performance

We show the performance of our translation framework by evaluating the polarity attribution accuracy of the dataset proposed in [30]. The dataset was constructed using distant supervision by aligning events extracted from biomedical literature by REACH reading system with polarity labels curated from the SIGNOR database. The dataset contains hundreds of sentences associated with protein-protein interaction events, and the *controller* (Cause), *trigger word* (Predicate) and *controlled* (Theme) are provided with both the interaction and predicate polarity label. The total number of sentences is 139 in the *test* dataset, and there are 77 cases in which the interaction polarity contradicts with the predicate polarity.

In our studies, we apply three NLP extraction systems (TRIPS, REACH and DeepEventMine) to extract the event annotations from each sentence in the *test* dataset and then use our methods in section 3.3 to extract the causal interactions and their polarity. Then our task is transformed into a binary classification of event polarity in a sentence. The predicted polarity is used to evaluate the true polarity of each signed interaction in the *test* dataset. As mentioned before, sometimes NLP extraction systems cannot detect the relations between events and entities (Figure 2) which means the translation framework will not identify any interactions within an EAG. Thus, we ignore such EAG as we assume the NLP extraction systems are expected to recognize both the entities and the relations accurately. For machine reading systems TRIPS and REACH, we filter out the causal interactions by comparing the regulator and regulated entity with *controller* and *controlled* in the dataset. We also apply the approximately names matching technique using the

Levenshtein-based string distance to reduce the amount of mismatches of these two entities. For DeepEventMine system, we feed the golden entities (*controller*, *controlled*) into the NER part of the pipeline. Then, we extract their interactions with the polarity labels for comparison. To avoid the influences of duplicated event subtypes, we assume that the extraction is correct as long as one of the predicted polarity extracted from a sentence is consistent with the true polarity. we compute the number of predicated polarity between regulator and regulated entity if an interaction can be found for these two entities.

Table 6: Polarity classifcation results by the number of predicted polarity on the test dataset

Tools	Number of predicted polarity	Precision	Recall	F1
TRIPS/DRUM	25	0.7	0.875	0.778
REACH	103	0.705	0.741	0.723
DeepEventMine	50	0.853	0.967	0.906

As shown in Table 6, the deep learning-based system DeepEventMine outperforms (with the F1-score of 0.906 on polarity classification) the machine reading systems, which indicates that deep learning approaches can better identify the events in a sentence, especially the complex events. The average performance of machine reading systems can achieve the score of 0.7. Since the dataset is first constructed by the extraction of REACH, almost every interaction event can be extracted using the REACH engine, while TRIPS performs poorly with only 25/133 interactions extracted. In section 3.3, we mention that the identification of interaction polarity is usually influenced by negative terms and complex event structure which is hard for NLP systems to capture. In this result, we briefly investigate how well event annotations from NLP systems can be converted into interactions and in such a way promote the polarity identification.

4.2 Translations for MTOR pathway events

To evaluate our translation framework for generating reaction models, we choose two different datasets related to mTOR pathway evaluated in [8] as our examples.

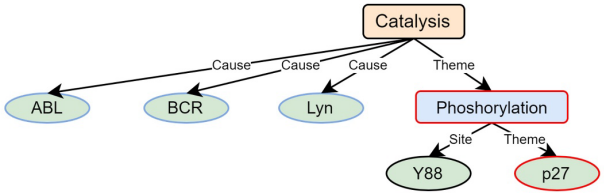
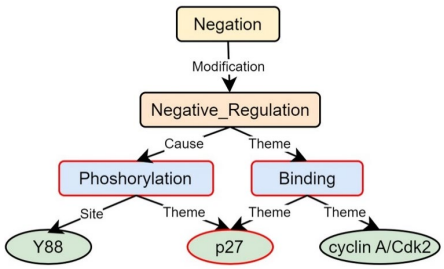
MTOR-HMN is a human curated mTOR pathway map [36]. The pathway is collected by Payao pathways, which is a community-based SBML model platform and all pathways are drawn on CellDesigner by manual curation.

MTOR-ANN consists of 60 PubMed abstracts associated with the mTOR pathway in MTOR-HMN. These abstracts are relevant to association and dissociation events in the pathway which is aimed at promoting the automatic extraction of these events. This corpus [32] follows the BioNLP ST event representation. Compared to the output of event extraction systems, the annotation density in this corpus is relatively high, which indicates that a wealth of information is contained in the abstract. We would like to see the performance of our translation methods with human-level extraction capabilities and how much pathway information we can extract from this corpus.

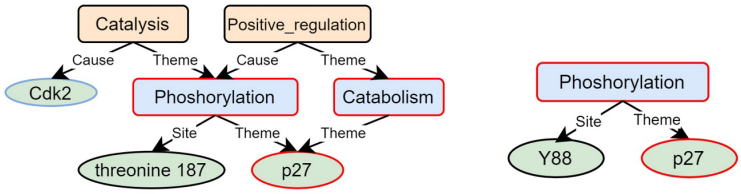
In MTOR-HMN, the paper (PMID:17254966) is a referenced document for the state transition of p27. Table 7 shows the sentences of the abstract describing reactions in MTOR-ANN and the EAG generated for each event. In sentence (1), which is followed by the sentence (2), p27 is phosphorylated by three modifiers, ABL, BCR and Lyn on residue Y88. Then a species is created with the name p27 and residue state set to Ø. An implicit product of this Phosphorylation event has a Phosphorylated residue state. In sentence (2), "Y88 Phosphorylation" refers to the phosphorylation of p27 on site Y88, which causes the negative regulation of another binding event. In this case, since there exists a phosphorylation event of p27, we could identify that the phosphorylated p27 does not prevent the binding of unphoshorylated p27 with cyclin A/Cdk2. However, in sentence (3), entity p27 has two Phosphorylation events on site Y88 and threonine

187, which makes the state of entity p27 unclear. That is because for the text span "phosphorylated on threonine 187", the Theme of this event is annotated as p27, not "Y88-phosphorylated p27", which indicates that sometimes even manual annotations could not capture the distinctions among instances of the same continuant and result in an erroneous interpretation of its surrounding biological contexts. In order to avoid the disambiguation of annotations, in our network format we only assign two values to the residue state, on and off, and omit the site information for simplicity. Therefore, in sentence (1), (2) and (3), we could identify the instances of the continuant p27 and map them to species.

Table 7: Sentence describing reactions in PMID: 17254966

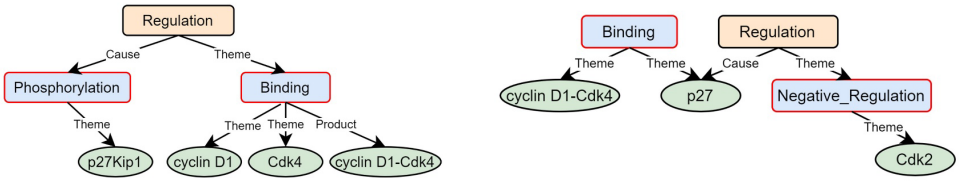
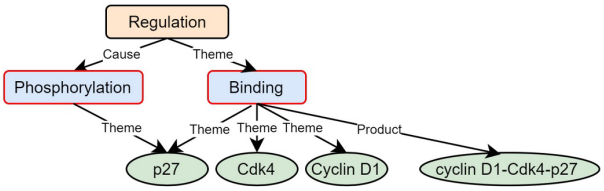
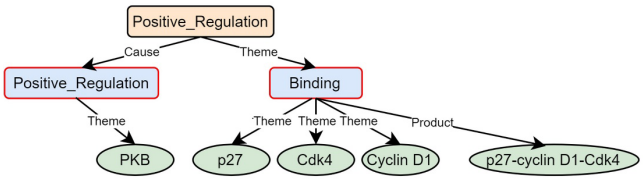
PMID: 17254966
(1) A conserved tyrosine residue (Y88) in the Cdk-binding domain of p27 can be phosphorylated by the Src-family kinase Lyn and the oncogene product BCR-ABL. 
(2) Y88 phosphorylation does not prevent p27 binding to cyclin A/Cdk2. Instead, it causes phosphorylated Y88 and the entire inhibitory 3(10)-helix of p27 to be ejected from the Cdk2 active site, thus restoring partial Cdk activity. 

(3) Importantly, this allows Y88-phosphorylated p27 to be efficiently phosphorylated on threonine 187 by Cdk2 which in turn promotes its SCF-Skp2-dependent degradation.



For the entity with complex type, besides the *uniqueAttributes* mentioned above, the *participants* attribute is added for each entity when creating the species. For example, in MTOR-HMN, the reference (PMID: 18710949) describes a heterodimer association reaction. Three species p27, Cyclin D and CDK4 are bound together to form a complex named Cyclin D-CDK4-p27. In the annotation of the referenced abstract in MTOR-ANN, these three reactants are annotated, but the complex species is annotated with different names ("p27-cyclin D1-Cdk4" or "cyclin D1-Cdk4-p27"). Therefore, we use the *participants* attribute to identify the instances of this complex entity. The *participants* attribute is unique which means that if complex entities have the same constituents, we only create one ComplexSpecies for them. Other entity names are added as aliasName attribute for this species. Table 8 shows some sentences related to the association reactions. In both sentences (1) and (2), p27, Cdk4 and Cyclin D1 are the Themes of the binding event. The entity "cyclin D1-Cdk4-p27" and "p27-cyclin D1-Cdk4" appearing in the product role has the same participants, which indicates that these two entities are actually the same complex species with different names.

Table 8: Sentences describing reactions in PMID: 18710949

PMID: 18710949
(1) Phosphorylation of p27Kip1 regulates assembly and activation of cyclin D1-Cdk4. p27 mediates Cdk2 inhibition and is also found in cyclin D1-Cdk4 complexes. 
(2) The present data support a role for p27 in the assembly of D-type cyclin-Cdk complexes and indicate that both cyclin D1-Cdk4-p27 assembly and kinase activation are regulated by p27 phosphorylation. 
(3) Here we show that PKB activation and the appearance of p27pT157 and p27pT198 precede p27-cyclin D1-Cdk4 assembly in early G(1). 

Furthermore, authors usually omit annotating the product of binding events. If a binding event does not have a Product entity, the name of complex species is created by other Theme entity names separated by an underscore character '_'. In the first sentence (1), p27 and cyclin D1-Cdk are bound together. Although the product of this binding event is not explicitly stated, we can learn

that this complex appearing in the product role is composed of p27 and cyclin D1-Cdk, and we name it as p27_cyclin D1-Cdk. Similarly, in the reference (PMID: 18710949), we can find that cyclin D1-Cdk have two subcomponents (cyclin D1 and Cdk4). Thus, the product of previous binding event is a complex species formed by three components, p27, cyclin D1 and Cdk4. By detecting the pariticipants of the ComplexSpecies, we can use this attribute to determine which entities belong to the same complex species. A special annotation in mapping the entities is the equivalence relations. Among the equivalent entities, we randomly choose one to create a species and add other entity names to its aliasName attribute.

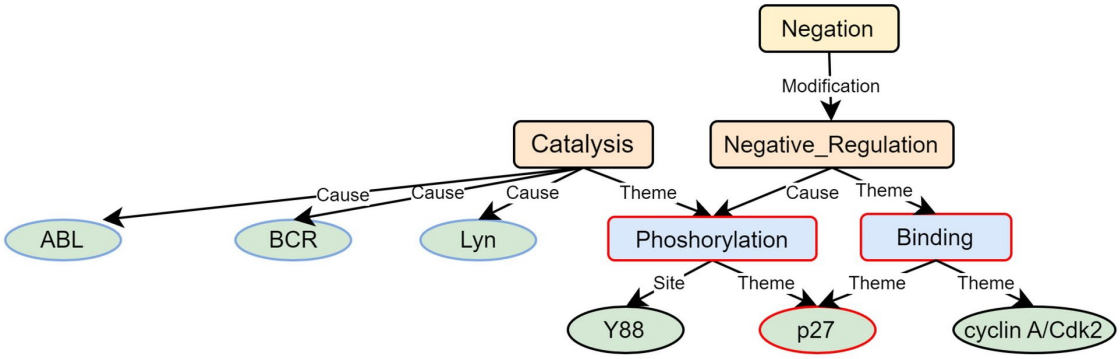


Figure 12: event contraction for the sentences in PMID: 17243966

For the mapping from events to reactions, we follow the algorithm described in section 3.5.2. For example, In Table 7 the Theme entity p27 of Phosphorylation event is mapped to a unique p27 species. The graph displayed in Table 7 are connected components of EAG translated from the reference abstract, and nodes representing mechanism events with the same Theme species, type are contracted into a single node. Then, we will remove the redundant edges between contracted event node and its Theme species. The contraction of two Phosphorylation events in sentence (1) and (2) is shown in Figure 12. Finally, all the event annotations in the abstract (PMID: 17254966) can be translated into a reaction network and converted to a SBML file (Figure 13).

The final graph only has two subgraphs, which means that no relationship between entity p27 and p27Kip1 could be found from the annotations. Except for p27Kip1, we can see that events are translated into a single network of reactions and the state transition of p27 is modified by four species (ABL, BCR, Lyn and Cdk2). The phosphorylated p27 is involved in a degradation process of p27 as a catalysis.

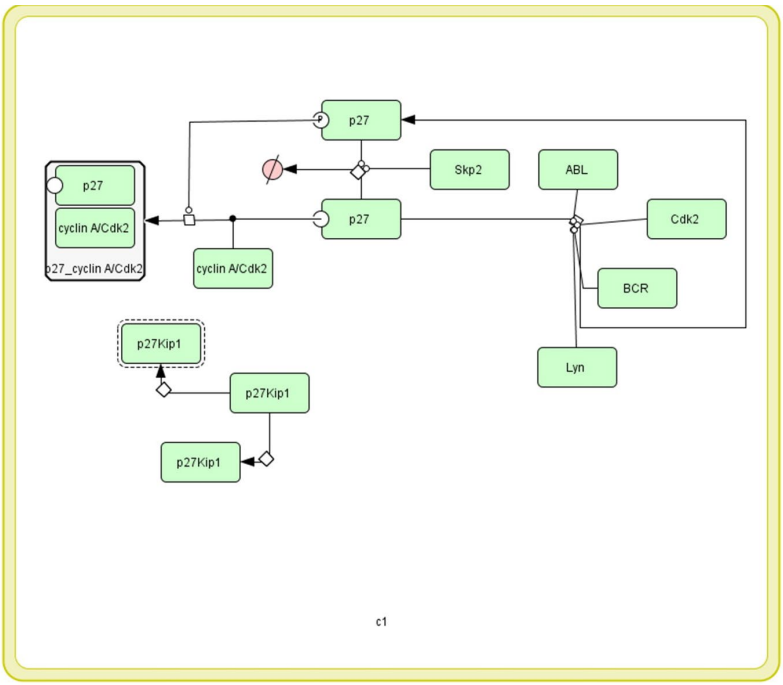


Figure 13: Generated SBML file (PMID: 17243966) which is visualized in CellDesigner

The following table shows the number of species, reactions for two different mTOR pathway datasets, and our generated reaction network (MTOR-RN) from MTOR-ANN. In the step of creating species, we identify the equivalent entities using the attributes in our network. In the next two steps, we map all the events to reactions and contract the reaction node. As a result, in MTOR-RN the number of species is 866 and the number of reactions decreases to 419.

Table 9: Number of species, reactions for the different datasets and generated reaction network

	# species	# reactions
MTOR-HMN	2242	777
MTOR-ANN	2457	857
MTOR-RN	866	419

5.0 Challenges for Current Translation

Building large-scale biological models from scratch is a time-consuming work that requires a significant amount of expertise. One of the challenges of this task is that models might be constructed differently even based on the same knowledge or experiment data. For example, if we want to build an element-based model by translating events from literature, there are multiple options for interpreting biological events in the form of models. In Figure 14, consider the sentence "The phosphorylation of BAD by MAPK causes an increase of BAX" which contains nested interactions, and in this case, substitution is required as "MAPK" is a Cause entity for the "phosphorylation" event. Here, we present two options for handling this sentence.

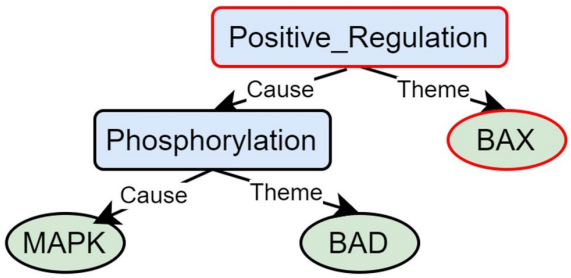


Figure 14: Example sentence that can be handled using different substitution methods.

Chaining Substitution. The first option would be to assume that the source entity is responsible for the target mechanism event. This is often the case with natural language and it would seem logical to assume that MAPK and BAD or BAD and ERK have a causal relationship. However, this may not be correct in modeling. For example, when the phosphorylation of BAD has other regulators, the increase of these regulators might cause the production of downstream

elements, which may be in contradiction with the original sentence. In general, by using only the output from the first event, one loses event specificity.

Set Substitution. Another option presented here maximizes the specificity of the input by combining two entities from the source mechanism event. That is to say the regulatory event is regarded as “Phosphorylation of BAD by MAPK”, therefore the influences of source mechanism event can be written as “BAD AND MAPK” using logical operator.

Another challenge is still the identification of biological entities. For protein modification, we do not translate this information, as we find that most Phosphorylation events do not have Site information. This means that we need more information from external sources to identify the modification of species or entity. If an entity is not involved in any Localization event, it is also difficult for us to identify the compartment. This means that in actual modeling, elements with different compartments might be the same components.

6.0 Conclusion

In this work, we present a framework for automating the translation from biological event annotations to other representation formats. We introduce two representations: an event annotation graph for representing the complex semantics from event annotations, and a SBML-compatible network format as a bridge for conversion from events to a reaction model. We have proposed several methods to extract the causal relationships between entities and the semantic annotations (e.g, effect and mechanism on relationships). Event annotation graph provide a graphical representation of events that describe the detailed semantics from annotations.

The extraction of causal relationship paths (CRPs) remove the need to traverse the graph of annotations and could extract the interaction relations with high accuracy. We examined the capabilities of our mapping algorithms using sentences annotated as complex event structure. The SBML-compatible reaction network enables the integration of piece information on biological events scattered on publications and can be converted into other representation formats such as SBML. Lastly, the translation framework facilitates rapid construction of biological models and enables a reliable extraction of useful information with the given event annotations.

Bibliography

[1] Kim, J.-D., T. Ohta, and J.i. Tsujii, *Corpus annotation for mining biomedical events from literature.* BMC bioinformatics, 2008. **9**: p. 1-25.

[2] Miwa, M. and S. Ananiadou. *NaCTeM EventMine for BioNLP 2013 CG and PC tasks*. in *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013.

[3] Trieu, H.-L., et al., *DeepEventMine: end-to-end neural nested event extraction from biomedical texts.* Bioinformatics, 2020. **36**(19): p. 4910-4917.

[4] Albert, R. and R.-S. Wang, *Discrete dynamic modeling of cellular signaling networks.* Methods in enzymology, 2009. **467**: p. 281-306.

[5] Ohta, T., et al. *Overview of the pathway curation (PC) task of bioNLP shared task 2013*. in *Proceedings of the BioNLP shared task 2013 workshop*. 2013.

[6] Cohen, P.R., *DARPA's Big Mechanism program.* Physical biology, 2015. **12**(4): p. 045008.

[7] Gyori, B.M., et al., *From word models to executable models of signaling networks using automated assembly.* Molecular systems biology, 2017. **13**(11): p. 954.

[8] Spranger, M., S.K. Palaniappan, and S. Ghosh, *Measuring the State of the Art of Automated Pathway Curation Using Graph Algorithms-A Case Study of the mTOR Pathway.* arXiv preprint arXiv:1608.03767, 2016.

[9] Spranger, M., S.K. Palaniappan, and S. Ghosh, *Extracting biological pathway models from nlp event representations.* arXiv preprint arXiv:1608.03764, 2016.

[10] Fluck, J., et al. *BEL networks derived from qualitative translations of BioNLP Shared Task annotations*. in *Proceedings of the 2013 workshop on biomedical natural language processing*. 2013.

[11] Becker, E.W., K.N. Bocan, and N. Miskov-Zivanov. Nested Event Representation for Automated Assembly of Cell Signaling Network Models. in Formal Methods. FM 2019 International Workshops: Porto, Portugal, October 7–11, 2019, Revised Selected Papers, Part II 3. 2020. Springer.

[12] Kim, J.-D., et al. *From text to pathway: corpus annotation for knowledge acquisition from biomedical literature*. in *Proceedings of The 6th Asia-Pacific Bioinformatics Conference*. 2008. World Scientific.

[13] Kim, J.-D., et al., *GENIA corpus—a semantically annotated corpus for bio-textmining.* Bioinformatics, 2003. **19**(suppl_1): p. i180-i182.

[14] Kim, J.-D., et al. *Overview of genia event task in bionlp shared task 2011*. in *Proceedings of BioNLP shared task 2011 workshop*. 2011.

[15] Stenetorp, P., et al. *BRAT: a web-based tool for NLP-assisted text annotation*. in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 2012.

[16] Allen, J., et al. *Effective broad-coverage deep parsing*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018.

[17] Valenzuela-Escarcega, M.A., G. Hahn-Powell, and M. Surdeanu, *Description of the Odin event extraction framework and rule language.* arXiv preprint arXiv:1509.07513, 2015.

[18] Hucka, M., et al., *The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.* Bioinformatics, 2003. **19**(4): p. 524-531.

[19] Demir, E., et al., *The BioPAX community standard for pathway data sharing.* Nature biotechnology, 2010. **28**(9): p. 935-942.

[20] Funahashi, A., et al., *CellDesigner: a process diagram editor for gene-regulatory and biochemical networks.* Biosilico, 2003. **1**(5): p. 159-162.

[21] Xu, J., J. Jiang, and H.M. Sauro, *SBMLDiagrams: a python package to process and visualize SBML layout and render.* Bioinformatics, 2023. **39**(1): p. btac730.

[22] Miskov-Zivanov, N., D. Marculescu, and J.R. Faeder. *Dynamic behavior of cell signaling networks: model design and analysis automation*. in *Proceedings of the 50th Annual Design Automation Conference*. 2013.

[23] Telmer, C.A., et al. *Dynamic system explanation: DySE, a framework that evolves to reason about complex systems-lessons learned*. in *Proceedings of the Conference on Artificial Intelligence for Data Discovery and Reuse*. 2019.

[24] https://melody-biorecipe.readthedocs.io/en/latest/index.html

[25] Sayed, K., et al. *Recipes for translating big data machine reading to executable cellular signaling models*. in *Machine Learning, Optimization, and Big Data: Third International Conference, MOD 2017, Volterra, Italy, September 14–17, 2017, Revised Selected Papers 3*. 2018. Springer.

[26] Björne, J. and T. Salakoski, *TEES 2.2: biomedical event extraction for diverse corpora.* BMC bioinformatics, 2015. **16**(16): p. 1-20.

[27] Miwa, M., et al., *Extracting semantically enriched events from biomedical literature.* BMC bioinformatics, 2012. **13**(1): p. 1-24.

[28] Perfetto, L., et al., *SIGNOR: a database of causal relationships between biological entities.* Nucleic acids research, 2016. **44**(D1): p. D548-D554.

[29] Ahmed, Y., C.A. Telmer, and N. Miskov-Zivanov, *CLARINET: Efficient learning of dynamic network models from literature.* Bioinformatics Advances, 2021. **1**(1): p. vbab006.

[30] Noriega-Atala, E., et al. *Understanding the Polarity of Events in the Biomedical Literature: Deep Learning vs. Linguistically-informed Methods*. in *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications*. 2019.

[31] Bornstein, B.J., et al., *LibSBML: an API library for SBML.* Bioinformatics, 2008. **24**(6): p. 880-881.

[32] Ohta, T., S. Pyysalo, and J.i. Tsujii. *From pathways to biomolecular events: opportunities and challenges*. in *Proceedings of BioNLP 2011 Workshop*. 2011.

[33] Bossy, R., P. Bessières, and C. Nédellec. *Bionlp shared task 2013–an overview of the genic regulation network task*. in *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013.

[34] Aghamiri, S.S., et al., *Automated inference of Boolean models from molecular interaction maps using CaSQ.* Bioinformatics, 2020. **36**(16): p. 4473-4482.

[35] Sayed, K., C.A. Telmer, and N. Miskov-Zivanov. Motif modeling for cell signaling networks. in 2016 8th Cairo International Biomedical Engineering Conference (CIBEC). 2016. IEEE.

[36] Caron, E., et al., *A comprehensive map of the mTOR signaling network.* Molecular systems biology, 2010. **6**(1): p. 453.