REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
07-09-2018	Final Report	3-Jul-2014 - 2-Jul-2018

4. TITLE AND SUBTITLE

Final Report: R3E: Reading, Reasoning and Reporting Explanations

5a. CONTRACT NUMBER

5b. GRANT NUMBER

W911NF-14-C-0109

5c. PROGRAM ELEMENT NUMBER

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES

Smart Information Flow Technologies (SIFT)
211 N. 1st. St., Suite 300

Minneapolis, MN 55401 -2078

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

66063-NS-DRP.4

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Mark Burstein
UU	UU	UU	UU		19b. TELEPHONE NUMBER
					781-799-3603

RPPR Final Report
as of 20-Sep-2018

Agency Code:

Proposal Number: 66063NSDRP **Agreement Number: W911NF-14-C-0109**
INVESTIGATOR(S):

Name: Mark H. Burstein
Email: burstein@sift.net
Phone Number: 7817993603
Principal: Y

Organization: **Smart Information Flow Technologies (SIFT)**
Address: 211 N. 1st. St., Suite 300, Minneapolis, MN 554012078
Country: USA
DUNS Number: 103477993 EIN:

Report Date: 02-Oct-2018 Date Received: 07-Sep-2018
Final Report for Period Beginning 03-Jul-2014 and Ending 02-Jul-2018
Title: R3E: Reading, Reasoning and Reporting Explanations
Begin Performance Period: 03-Jul-2014 **End Performance Period:** 02-Jul-2018
Report Term: 0-Other

Submitted By: David McDonald Email: dmcdonald@sift.net
Phone: (781) 718-1964

STEM Degrees: 0 **STEM Participants:** 0

Major Goals: SIFT's R3 (Reading, Reasoning, Remembering) project in DARPA's Big Mechanism program was focused on the problems of (1) developing an automated reading system for scientific articles capable of extracting semantic descriptions of relevant details about mechanisms, and (2) interpreting those descriptions in the context of an evolving background model of the larger mechanistic system being studied – a process we call localization against a model. That is, we sought to develop methods to identify how descriptions of mechanisms and the entities involved in those mechanisms in texts relate to parts of a previously known but incomplete model, so that the model could be extended or revised with the new information described in those texts.

We developed a means of extracting these forms from the summary descriptions accompanying parts of the Reactome model and built an enhanced representation we called BioPax+ that contained the key content.

We did experiments in the summer/fall of 2016 that showed how we could use the information we had extracted about the structure of protein complexes when active to help localize phrases in articles against our enhanced Reactome model.

We worked to provide the CURE consortium system with the same functional information we used in R3 to localize functional descriptions, and they now use it directly in their Assembly process, where it helps to identify statements with the same intent produced in different texts or by different readers on the same texts, rather than directly comparing single descriptions to a prior model.

In addition, we also extended the R3 reader, built around Sparser, so that it could act as an additional reader for the CURE system. Since Sparser is designed for semantic information extraction, it is much faster than TRIPS/DRUM, and even slightly faster than REACH, in that it can process most articles it reads in under 1 second on a MacBook Pro. By June of 2017, we were processing and providing HMS with native INDRA/CURE formatted output for more than 17,000 open text articles.

Accomplishments: We accomplished all our major goals. Details are provided in the report.

Training Opportunities: Nothing to Report

RPPR Final Report
as of 20-Sep-2018

Results Dissemination: We published three peer-reviewed articles that were presented at conferences of the AAAI, IJCAI, and ACS (Advances in Cognitive Systems). We made multiple presentations at each of the PI meetings held by DARPA. We collaborated extensively with the Sorger and Fontana labs at the Harvard Medical School as part of the CURE Consortium.

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: The Sorger Lab at the Harvard Medical School is the lead performer and integrator in the CURE consortium. We provided them with data cataloging the active proteins in the Reactome data base. We provided them with the semantic interpretations of the 17 thousand sentences they identified as relevant to the problem.

When they that our system for reading journal articles as accurate as they other systems they worked with and also significantly faster (roughly one article per second), we provided them with executables that they mounted on their Cloud and subsequently ran it on more than a million articles.

PARTICIPANTS:

Participant Type: PD/PI
Participant: Mark Burstein
Person Months Worked: 2.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co PD/PI
Participant: David McDonald
Person Months Worked: 6.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Staff Scientist (doctoral level)
Participant: Scott Friedman
Person Months Worked: 3.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Staff Scientist (doctoral level)
Participant: Alex Plotnick
Person Months Worked: 6.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

RPPR Final Report
as of 20-Sep-2018

Participant Type: Other Professional
Participant: Laurel Bobrow
Person Months Worked: 3.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Consultant
Participant: Rusty Bobrow
Person Months Worked: 3.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Faculty
Participant: James Pustejovsky
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Contract Number	W911NF-14-C-0109
Title	R3E: Reading, Reasoning and Reporting Explanations
Contractor's Name	Smart Information Flow Technologies, LLC
Contractor Address	319 1st Ave, Suite 400, Minneapolis, MN 55401-1689
Deliverable	Final Technical Report August 1, 2014 – July 31, 2018
Status	Does not contain classified or sensitive material

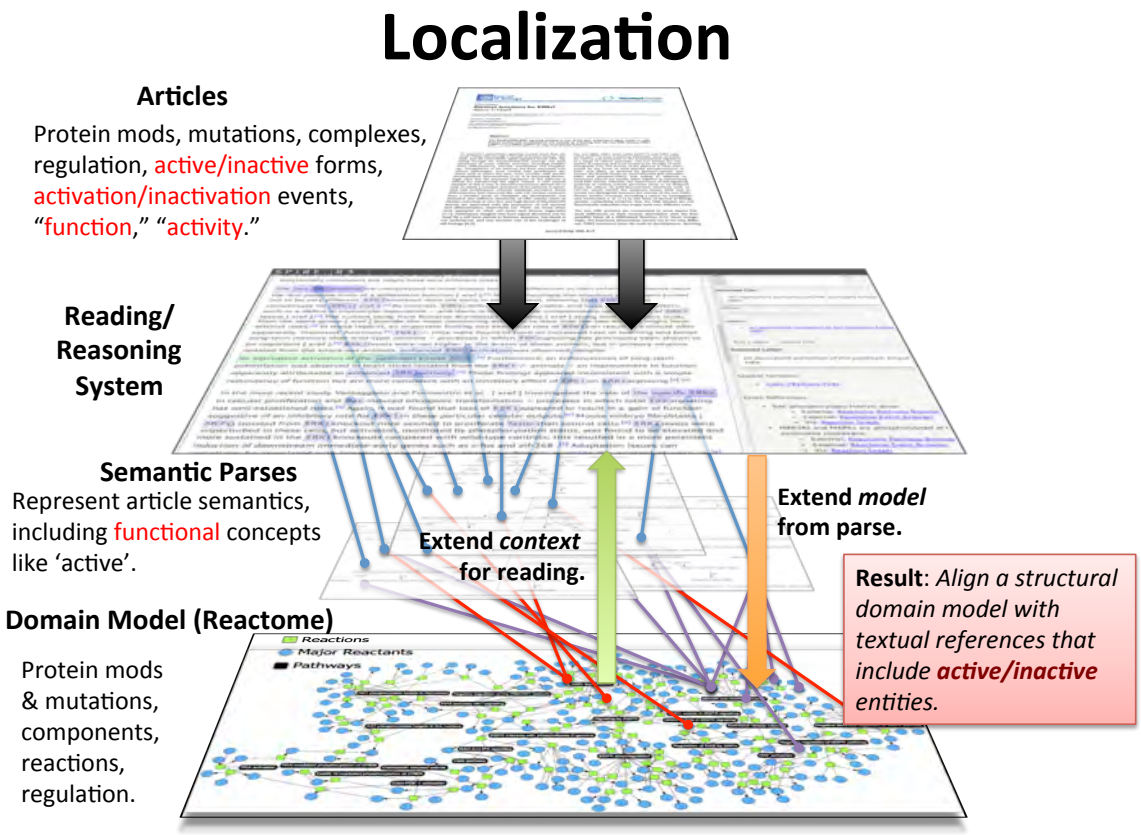


Table of Contents

Table of Figures

Abstract

SIFT's R3 (Reading, Reasoning, Remembering) project in DARPA's Big Mechanism program was focused on the problems of (1) developing an automated reading system for scientific articles capable of extracting semantic descriptions of relevant details about mechanisms, and (2) interpreting those descriptions in the context of an evolving background model of the larger mechanistic system being studied – a process we call *localization against a model*. That is, we sought to develop methods to identify how descriptions of mechanisms and the entities involved in those mechanisms in texts relate to parts of a previously known but incomplete model, so that the model could be extended or revised with the new information described in those texts. Our reading system, R3/SPARSER was developed largely during the first two years of the project based on a previously existing system, SPARSER (McDonald 1996), and emphasized multi-sentential reading and co-reference. Since SPARSER was designed for semantic information extraction, it is extremely fast, indeed much faster than TRIPS/DRUM, and even slightly faster than REACH, the other two readers used by the CURE consortium to assemble its mechanism descriptions. SPARSER can process most articles it reads in well under 1 second on a MacBook Pro, and frequently under ½ second. By June of 2017, we were processing and providing HMS with native INDRA/CURE formatted output for more than 17,000 open text articles. By the end of the project Sparser had read close to 1 million articles.

R3's Localization system, developed beginning in the second year, demonstrated the utility of maintaining mechanistic models at multiple levels of abstraction, specifically to include functional descriptions of underlying mechanisms. These descriptions thus included not just chemical reactions, but what functional states, such as "active", were arrived at and what functions were made possible by entering those states. R3 initially extracted these key functional forms from the summary descriptions accompanying parts of the Reactome model we were extending. It then built an enhanced representation of the overall model in a representation we called BioPAX+ (based on the BioPAX representation used in the Reactome model). Our work provided the CURE consortium system with the same functional information collected by R3 to localize functional descriptions, which they used productively in their Assembly process, where it helped to identify statements with the same intent produced in different texts (or by different readers on the same texts), enabling the accumulation of a larger and more accurate model. In our final year we developed a "gold-leaf" (that is, not really gold) standard reference set against which we did our final evaluation of the localization system.

1. Summary

Machine reading does not end with a parse or even with a semantic interpretation. When we read to inform ourselves, we use our current model of the world to guide our interpretation of the text, and then we reconcile this interpretation with our model to determine its consistency with our prior beliefs and perhaps to accept and incorporate the new information. Our interpretation might corroborate, extend, or conflict with our prior model and perhaps cause us to revise or extend it. We refer to this model-centric activity as *reading with a model*. This is the central goal of our ongoing work on the Reading, Reasoning, and Reporting (R3) cognitive system, as part of DARPA's Big Mechanism. R3 reads articles in molecular biology to extend and revise its models of biological mechanisms, specifically those having to do with signaling pathways.

A central capability— and research challenge— for cognitive systems that read with a model is *localizing* (i.e., recognizing and retrieving) in that model, the entities and events mentioned in the text, in order to begin the process of reconciliation. Localization allows the system to establish a mapping between the interpreted text and the model and to enable bidirectional information flow between the model and the text interpretation process.

Localization is an important part of "pre-assembly", the reasoning that is needed to establish how new information gleaned from texts is understood and disambiguated in the context of prior knowledge, and the reader's current focus. It determines how the new information in the text would be integrated with that which had previously been assimilated into a big mechanism model. Localization has an impact on the interpretation of the text itself, since it can help in disambiguating references to proteins and processes that are only vaguely referred to, but where the context strongly suggests what specific elements of the model were meant.

The goal of R3 was to enable transfer of information about mechanisms gleaned from the text into a model (**interpretation-to-model**), either to extend the model or to identify and annotate conflicts. If localization first establishes correspondence between parts of the model and the text, it can also improve the text interpretation process (**model-to-interpretation**) by making the reading system aware of details about known entities and processes (such as their types and relations to other mentioned entities) so that when they are mentioned only by reference, those references are not overly vague or ambiguous. If known entities and events in the text are not localized correctly within the model, then the interpretation is less successful since new related information is not properly integrated.

Building a system that reads and localizes to a pre-existing biological pathway model that has been developed and curated by domain experts involves many domain-general and domain-specific challenges, including:

- Texts frequently use the same word to reference different types in the model. For instance, "RAS" can refer to a protein, a gene, or a larger multi-protein complex, even within a single article.
- Texts may describe things at different levels of abstraction than the model. For example, authors frequently talk about the abstract *functional properties* of entities (e.g their "activation"), although the background model may only describe the entity interactions and molecular structures.
- One process or entity may be a component of many larger processes or entities in the model.

Some of these challenges are due to information mismatch between journal articles and domain models, where the model has no ontological categories or relations to represent the level of events in the text. For instance, articles frequently mention post-translational modifications, including phosphorylation (binding a phosphoryl group to a molecule). Another is the formation of complexes with multiples of the same molecule. (Dimerization is the binding of two like molecules to form a dimer.) These and other types of events are present in the formal model, but only implicitly: there are no categories for these events in the formal BioPAX model, so one must compare the reactants and products and infer these events. To rapidly recognize these events when they are mentioned in the text, we must extend the model by describing them explicitly.

Furthermore, biology articles often describe processes at a functional level, but at the time of this work, the background BioPAX Reactome domain model does not represent entity function. In the cell signaling domain, processes and entities are functionally described as being "switched" on or off, and processes or entities can "activate" or "inhibit" other processes or entities. Entities are "activated" when their structure or their bindings enable them to act as catalysts, inhibitors, or scaffolds in other reactions. Activation conditions vary across entities: proteins like MEK and ERK are activated when they are phosphorylated; others are deactivated when they are phosphorylated; others are activated when they are dimerized; and so forth. This functional language allows biologists to compactly refer to entities with causal affordances without needing to know their exact structure.

To capture the specific states of activation for the particular proteins in our model, we needed a source of information about the complexes involving those proteins when they were considered "active" or "inactive." We found our source embedded within the model itself: Expert biologists wrote textual summaries about each reaction in the model, and associated these summaries with the reaction entities. For example:

- "SOS1 is the guanine nucleotide exchange factor (GEF) for RAS. SOS1 *activates* RAS nucleotide exchange *from the inactive form* (bound to GDP) *to an active form* (bound to GTP)."
- "EGFR phosphorylates PLC-gamma1, thus *activating* it."
- "*Activated MAPK proteins* negatively regulate MAP2K1:MAP2K2 heterodimers..."

These examples of reaction summaries show how the language used for human consumption conveys the functional states of their primary participants, rather than their structural changes, which is described by the model. R3 learns functional knowledge—which it needs for localizing functional references in articles— by parsing these summaries and then analogically transferring knowledge into the associated events in its domain model. When subsequent texts describe entity function, R3 can identify the corresponding structures in its extended model.

What we demonstrated was that an important key to successful localization of textual biological mechanism descriptions was having an effective means of interpreting descriptions that involved the *functional* states of molecules (e.g., active vs inactive) in terms of their underlying molecular structure, since that was how the mechanistic models we sought to relate these texts to would describe the proteins. We developed a means of extracting these forms from the summary descriptions accompanying parts of the Reactome model and built an enhanced representation we called BioPAX+ that contained the key

content. We did experiments in the summer/fall of 2016 that showed how we could use the information we had extracted about the structure of protein complexes when active to help localize phrases in articles against our enhanced Reactome model. We also adapted R3 so that it could use either R3's own SPARSER semantic parsing system or (new this year) IHMC's DRUM parser to demonstrate localization against Reactome. We collected data about the effectiveness of this vs. localization without the added information and showed that the improvement was highly significant. We describe these preliminary results again in this report, and show new evidence from more extensive experiments performed this year.

Overall, R3 integrates deep semantic parsing, ontology mapping, interpretation-to-model structure-mapping, and functional reasoning. Semantic parsing allows R3 to extract precise descriptions and determine entity types from local lexical context. R3's ontology mapping and localization allows it to transfer its semantic interpretation into other ontologies to identify any and all corroborating events and entities. R3 extends structure-mapping methods to support wide-scale event recognition and retrieval. Finally, R3's mechanism-level reasoning allows it to reason about functional factors— such as what it means when an article describes an entity as active— despite lacking direct functional knowledge in its initial domain model.

During the final years of the Big Mechanism program, we focused primarily on the scaling up of the R3 reading system and the localization of mechanisms found during reading against a model. That is, we wanted to have R3 identify how descriptions of mechanisms and the entities involved in those mechanisms, as found in texts, relate to parts of a previously known model.

We worked to provide the CURE consortium system with the functional information we used in R3 to localize functional descriptions, and HMS group within CURE now uses these R3 results directly in their Assembly process, where they help to identify statements with the same intent produced in different texts or by different readers on the same texts, rather than directly comparing single descriptions to a prior model. In CURE, the assembly process is unifying descriptions extracted by three different parsers (TRIPS/DRUM, REACH and R3/SPARSER) reading many of the same articles. Being faster, REACH and SPARSER read most of the articles, while TRIPS is used to parse key sentences only. We supported that process by providing key pieces of information about what activation means for different proteins, enabling HMS to compare different ways of referring to those complexes. We extended R3's coverage from just the EGFR signaling subset of Reactome to all of Human Reactome to support CURE's range of models, which required that we re-engineer parts of R3 so that it could read and process all of the Reactome model at once, and employ additional means of extracting functional information from the model.

In addition, we enhanced the R3/SPARSER reader so it could be integrated with CURE as an additional reader. Since SPARSER is designed for semantic information extraction, it is much faster than TRIPS/DRUM, and even slightly faster than REACH, in that it can process most articles it reads in under 1 second on a MacBook Pro. By June of 2017, we were processing and providing HMS with native INDRA/CURE formatted output for more than 17,000 open text articles. Our recent investigations indicate that SPARSER is producing higher precision than REACH on the information that both extract, and that we agree on a substantial portion of the results, which significantly improves the confidence scores that

CURE produces during assembly. By the end of the program CURE had used Sparser to read approximately one million articles.

Section 2 describes our approach to extracting and recognizing biological events and interactions from text, focusing on challenges for natural language understanding. Section 3 describes the R3 approach to localizing extracted bits of mechanisms against a prior semantic model. Section 4 describes empirical evidence of our reading and localization systems. Section 5 details our main activities supporting the CURE consortium.

2. The R3/Sparser Technical Article Reader

R3's reading engine, SPARSER, is extremely fast, processing a typical article in a half a second. From the beginning of the project, with the assistance of consultant Rusty Bobrow, we worked to adapt SPARSER to the problem of biomedical texts. Its original application was doing information extraction on texts that were written in a 'sublanguage' – a specific set of conventions that authors follow when writing about a particular narrow topic, such as quarterly earnings financial reports. Working from a semantic model of the terms and relations in these topics, SPARSER can automatically generate a semantic grammar that is tailored to the conventions of the sublanguage. In this mode it produces high precision results, while ignoring aspect of the text that are not in its semantic model.

SPARSER parses into a referential model of structured objects in a typed lambda calculus, rather than a set of predicate logic forms (McDonald, 2000). SPARSER's categories are taken from an ontology (a linguistically annotated domain model) whose upper structure is based on Dolce (Gangemi et al., 2002) and Pustejovsky's model of events (Pustejovsky, 1991). There is a middle level with ontological models for location, time, people, measurement, change in amount, and such. This core is extended with a ontology of biomedical phenomena that is deliberately designed to be close to how these phenomena are described in articles in order to simplify the parsing process. Individuals (instances of categories) represent the entities, events, and relationships that are identified when a text is read. Individuals are unique: the parsing process guarantees that every individual with a particular set of values for its properties is represented by a single object (Maida & Shapiro, 1982, McDonald, 2000). This guarantee is managed by a description lattice that coordinates partial descriptions as described below.

Adjustments to accommodate biology

Biomedical research articles are written to be read by other professional biologists who are presumed to have the requisite technical background. The brief mention of a well-known mechanism *("RAS/RAF/MEK/ERK Pathway"*) is sufficient to evoke all of the details of the mechanism in the mind of the reader. This lets them effortlessly fill in information gaps that cannot be supplied by standard discourse techniques *("activated upon GTP loading and deactivated upon hydrolysis of GTP to GDP"* — loaded onto or hydrolyzed from what?). We need to have knowledge sources that enable our systems to do this too.

Like other science authors, biologists must keep their articles within length limits, so they use compaction techniques such as describing events using nominalized verbs and packing information into them as prenominal modifiers, *e.g., "EGFR and ERBB3 tyrosine phosphorylation," "mitogen-induced signal transduction."* This changes the usual grammatical cues (such as one would use on newswire text) and requires adopting knowledge-rich analysis techniques if parses are to be accurate.

A further property of biomedical text is that logically related information is usually distributed across multiple sentences. The following example is typical. The classification of the sites are given in the first sentence and their identity in the second. *"We observed two conserved putative MAPK phosphorylation sites in ASPP1 and ASPP2. The ASPP1 sites are at residues 671 and 746, and the ASPP2 sites are at residues 698 and 827."* In R3 we have enhanced our discourse history to let us combine information from both

sentences into a single, logically complete, representation that specifies the binding sites on ASPP1 and ASPP2 where MAPK catalyzes phosphorylation.

While the authors of biomedical journal articles do appear to be using a sublanguage of sorts, it is enormous, and it was not practical to attempt to build the full model of it that would be required to generate an adequate semantic grammar. As a consequence, we developed a new, more syntactic grammar with explicit interpretation rules, and a new control structure that delimits each successive sentence and makes a series of passes over it to develop the full analysis. By performing a succession of minimal analyses, we reduced the number and kind of decisions being made in each pass so that subsequent passes have accurate information about the entire sentence, allowing it to make decisions with more information than the prior, strictly left-to-right approach.

We now also support a suite of repair rules that are applied when composition rules fail. For example, if the interpretation ends with a set of stranded constituents, a set of pattern-directed rules takes over the search for alternatives. The overall effect is that we are able to use syntactic rules with more assurance, particularly within the long noun phrases that are a hallmark of biomedical articles. In the future, this process can be further refined using statistical evidence.

SPARSER's precision depends on having a thoroughly defined vocabulary. The quest to find better ways to identify and sufficiently define the technical vocabulary in the texts dominated our efforts in the first few months of the program., and continued throughout the program as we moved to be able to handle the vocabulary in the million-article corpus addressed by the CURE consortium. We bulked up our coverage by consulting public databases to accumulate a list of drug names and cell lines, and directly incorporated the UCD entity recognizer to assist with this process. We further analyzed available training materials to explicitly define many additional proteins and pathways. We also developed tools to extract "grounded" definitions of biological vocabulary (using identifiers from various standard databases such as HGNC, UNIPROT, Reactome) from the results of running both REACH and TRIPS on very large sets of articles (see section below on Extending/aligning biochemical vocabulary).

Reading with the aid of micro-models

We determined early on that it is impossible to thoroughly read a scientific article in molecular biology without first providing the system with the basic knowledge that biologists take for granted. Because the knowledge is 'obvious', it is left out of the text of the article as a normal aspect of writing texts in this genre. To compensate for this, we needed to provide the parser with pre-built models of the language and assumptions of simple 'Bio Mechanisms' such as switches, activation, binding, forming complexes, movement within the cell, changes in conformation, and the like.

To that end, Professor Pustejovsky developed a Generative Lexicon process model of the event structure of a molecular switch, and Dr. McDonald worked out a design for implementing it as a pre-instantiated model that is incorporated into the discourse history when any of its lexical triggers is encountered. So, for our example – "*Ras acts as a molecular switch that is activated upon GTP loading and deactivated upon hydrolysis of GTP to GDP*" – the model is added the moment that *Ras* is encountered by the parser. The

model incorporates links that are followed to fill in the missing information, namely that the substrates of the two actions are Ras proteins, and that that each action changes the state of the "switch" from off/inactive to on/active, or vice versa. Given this general background knowledge, when the denotation of, e.g., "*GTP loading*" is retrieved by the parser it will already include the fact the substrate is Ras, and enable the association of the 'active' state with a downstream effect.

Such models bridge the gap between the structural status of entities (e.g., their phosphorylation status, molecular subcomponents, and molecule bindings) and the salient functional capabilities of entities within a larger system (e.g., their ability to translocate and activate other entities). As part of this effort we developed a means of extracting these micro-models from the textual summary descriptions that accompanied parts of the Reactome model. With these available to its reading operations, R3 can use textual references to "*active RAS*" or "*RAS function*" to marshal important background knowledge about the structure of RAS when in the functionally active state, and the events that constitute RAS' function when active, respectively.

Biological vocabulary development

To facilitate the rapid buildup of Sparser's biology vocabulary, we developed tools to simplify the process of defining the language–concept definitions for new vocabulary. Relationships are defined in terms of type-restricted variables; the linguistic facts about the verb that realizes that relationship (or its nominal or adjective equivalent) can be stated compactly by associating variables with syntactic relationships (subject, complement of a specific preposition, etc.). The value restriction on the variable becomes a constraint on the semantic categories of the constituents that can validly fill the role. For example, the definition of the ontological category for 'encode' looks like this, including its various surface lexical forms.

```
(define-category encode           ;; name of the semantic category
   :specializes bio-process       ;; parent category
   :binds                         ;; list of semantic slots or relations
      ((encoder gene)             ;;  as (<relation> <type restriction>)
       (encoded protein))
   :realization       ;; specifies how syntactic & semantic rules get generated
    (:verb "encode"         ;; generates all verb forms.
     :noun "encoding"       ;; corresponding nominal forms.
     :etf (svo-passive)     ;; set of syntactic rule patterns to use
     :s encoder             ;; the slot filled by the syntactic subject
     :o encoded             ;; the slot filled by the syntactic object
     :of encoded            ;; the slot filled by preposition 'of'
           ))
```

So for active forms of the verb **encode**, associated here with the semantic category of that same name, the subject binds the event to role **encoder** (see :binds), which must be of semantic type **gene** while the object of the verb is the **encoded** role which must be a **protein**. These roles are semantically specializations of **agent** and **patient.** This same definition generates a set of rules for handling the association of these two roles with the correct syntactic elements of sentences in a variety of syntactic arrangements using active and passive forms of the verb, and also nominalizations.

Multi-sentence event reference resolution using a description lattice

We used Sparser's long-standing ability to resolve pronouns and definite noun phrases ("*…in RAS mutant cells. … in these cells)* as we moved to longer texts. For regular pronouns (*it*), we draw on the context in which it occurs to provide the type information that will let us search the discourse history for a full description of its referent. For non-pronominal references, which are rampant in the biomedical literature, we needed a different technique. Typically, the first reference to an entity will describe it in some detail, and then subsequent references will be more concise, using only a couple of characteristics to make clear what previous thing was being referred to. We needed a way to search for and identify the best prior matching entities. The solution we arrived at dove-tails with an independent related problem.

From the beginning of the project, we were concerned with problem of how to relate a protein, say BRAF, to the plethora of variations and contexts in which it appears in an article. If we are talking about its V600E point mutation, for example, we should not treat the mutation as simple property of BRAF in the same way as we treat the different ways it is spelled or its ID in the OBO ontologies. The mutated version behaves differently from the wild-type (not mutated) BRAF in how it reacts in particular experimental conditions and so should be represented as a distinct individual.

We concluded that we needed to design and implement a systematic way to keep track of entities like proteins in their various states given their different cell and organism type variations, with different mutations or post-translational modifications, in different cell lines, and so on. We based our design on the lattice structure previously developed for Sparser that handled partially-saturated individuals, and the incremental description refinement procedure worked out by Rusty Bobrow and Bonnie Webber. (see Bobrow & Webber (1980)). The resulting *description lattice* is essentially an incrementally constructed subsumption lattice that is augmented with each mention of a new entity in an article. This lattice provides, among other things, a way to quickly identify and to refer uniquely back to every previously seen, semantically compatible description.

The entities, events, and relationships that have been identified when a text is read are represented as individuals -- instances of categories in our lexically annotated semantic model. Individuals are unique: Every individual with a particular set of values for its properties is represented by a single object as guaranteed by the parsing process (see McDonald (2000) and Maida and Shapiro (1982)) and occupies a specific node in the description lattice.

Event reference resolution

Even in the simpler prose style of Reactome curators' comments on reactions, information is distributed across multiple sentences, providing new attributions about previously mentioned events, and using explicit and implicit connectives to introduce causal and temporal relationships between events—all key competences for reading about mechanisms. Our description lattice is key to how we now pull together semantically coherent descriptions across multiple sentences and sections of documents. For example, to fully understand a reaction its full set of reactants, the binding sites involved, and the nature of the products, the system must be able to find all of those bits of information which are typically described over the course of a number of sentences. Understanding both event and entity co-reference to the level where we know which state the proteins are in at each

point, and which parts of the protein have been modified (by the current or previous reactions) requires us to carefully maintain these distinctions.

Unique among the CURE readers, SPARSER does cross-sentence reference resolution by comparing the partial semantic descriptions extracted from each sentence (pronominal references are handled by another mechanism). Since later references tend to be less detailed, the process looks for recent mentions of descriptions that include the properties of the entity from the current sentence for which a referent is sought.

Consider this text from a figure caption. Rhetorically, it is a summary of a research result that is framed as a comparison of what happens in two different experimental conditions: untreated cells and cells that have been treated with an experimental drug (AZD6244). While detecting the parallel structure of the two sentences[1] is an important part of understanding what is being said, here we focus on the two phosphorylation events.

"In untreated cells, EGFR is phosphorylated at T669 by MEK/ERK, which inhibits activation of EGFR and ERBB3. In the presence of AZD6244, ERK is inhibited and T669 phosphorylation is blocked, increasing EGFR and ERBB3 tyrosine phosphorylation and up-regulating downstream signaling."

The description lattice is built up incrementally through composition, starting from the head. In the first sentence, our grammar takes the head to be the verb "*phosphorylate*," which is represented by an individual whose category inherits from post-translational-modification, which in turn inherits from caused-bio-process (then bio-process and perdurant). The compositions and their corresponding individuals (lattice nodes) are illustrated below. The original individual for the event of phosphorylation, with the *identifying index* 6560[2] goes through six successive compositions. At each step an attribute is added, for instance the residue T669, and a new individual is created to represent that combination, in this instance the individual 6651.

```
6560: phosphorylate
   "phosphorylate"

6651:  6560 + site = residue-6636
   "is phosphorylated at T669"

6652:  6651 + agent = pathway-6537
   "is phosphorylated at T669 by MEK/ERK"

6653:  6652 + substrate = protein-1438
   "EGRF is phosphorylated at T669 by MEK/ERK"

6654:   6653 + causally-related-to = inhibit-6650
   "EGRF is phosphorylated at T669 by MEK/ERK which inhibits …"

6655:  6651 +  context = cell-line-6642
    "In …, EGRF is phosphorylated at T669 by MEK/ERK which inhibits …"
```

[1] Both put the context in an initial prepositional phrase, which is followed by one or two main events, and ending with the consequence of that event.

[2] These index numbers on the individuals are incremental because this was run on a freshly loaded system where this was the first run of a text with any of this content. The numbers are arbitrary. They correspond to the count of individuals created so far.

Each of these compositions adds a *mention* to the discourse history that records the individual and its location in the text. If the individuals were created in a set of successive compositions, then their mentions are chained to reflect that. Similarly, the individuals themselves are linked 'up' to the individual they were derived from and 'down' to any individuals that build on them.

Our system knows that the phrase "*T669 phosphorylation*" in the second sentence is related to the first event because the index for the individual that represents the phrase is 6651, just as in the first event.[3] When the analysis of the second sentence is finished, an inspection process notices that the phosphorylation event is incomplete: it doesn't locally tell us the substrate protein that got the phosphate or the agent that caused the event. This prompts a search for this information in the discourse history. Using discourse relations as a guide (theme, focus, locality, document section, we ask at what other places in the text was description (phosphorylate + site=T669) mentioned,). This query takes us to the mention in the immediately prior sentence, where the chain of individuals locally linked to that first mention includes the missing substrate and agent.

This is enough to license R3 to identify the description formulated on the first mention to identify the other properties that the second has, and to copy or unify over any non-conflicting properties. Some issues remain. One is that the first event mention is not only linked to its logically required terms (agent, substrate, site), but also to a particular context ('untreated cells') and causal consequence ('inhibiting EGFR and ERBB3 activation').

We capture the event sequences in the two experimental conditions, marked in the first case by "***which*** *inhibits activation of EGFR and ERBB3*", and in the second case simply by the comma at the beginning of "**,** *increasing EGFR and ERBB3 tyrosine phosphorylation*". These event sequences are critical to interpreting the mechanisms involved and the comparisons being made. Then, similar comparisons will also be possible between article statements and the background BioPAX model.

Recognizing these event sequences allowed us to more thoroughly interpret BioPAX model comments, many of which describe downstream effects of the reaction being labeled. In particular, it let us identify which downstream events are considered significant outcomes of the reactions being labeled – effectively identifying 'affordances' from the many things that are causally connected to that event in the model.

Recognizing types of activity and causal effects

We made a number of other modifications to SPARSER to improve its semantic output on the complex sentences that appear in Reactome comments. Of particular importance is that we recognize correctly who the agents and reactants are, and handle event modifiers properly, such as in "*the ubiquitinase activity of BRAP*" as in

> *"Binding to activated RAS stimulates the ubiquitinase activity of BRAP, promoting autoubiquitination and relieving the inhibition of KSR1."*

Here, 'ubiquitinase' indicates the specific type of activity that is stimulated (the catalytic activity for ubiquitination of another protein by BRAP). Correctly identifying the kind of activity implied by a modifier is important for correctly interpreting these comments and associating them with the model.

[3] All instances that describe the same set of attributes are always represented by the identical individual.

A second class of improvement had to do with the interpretation of implicit chains of events (and sometimes, the implied causal sequencing of those events). The sentence above can again serve as an example. The subject of the sentence, "*Binding to activated RAS,*" is itself an event, with an implicit result that "stimulates" (that is, causes an increase in activity). We have an event causing ubiquitination activity of BRAP and subsequently 'promoting autoubiquitination' (implicitly of KSR1) and relieving inhibition on (i.e., activating) KSR1. As part of identifying the activities that are enabled, we also needed to ensure that the model tracks not just the proteins, but also the complexes they were involved in. There are a number of inferences required to fill in the implied event arguments, and identify that:

- BRAP is the protein binding to activated RAS — missing subject
- BRAP:RAS complex is thus activated as a ubiquitinase — interpret the modifier of 'activity'
- BRAP:RAS complex can then catalyze the autoubiquitination of KSR1 — infer complex formation as subject of 'promoting autoubiquitination' in the implied conjunction due to the presence of the comma
- Autoubiquitination of KSR1 leads to reduced inhibition of KSR1 activity (whatever that turns out to be) — Conjunction of clauses with same (event) subject as previous verb 'promoting' implies causation or temporal sequence; reduced inhibition of KSR1 is reduced inhibition of activity of KSR1 (metonymy)

To support the identification of these critical connections between sentence elements and thereby enable R3 to properly match the content of these sentences to the background model, we did two things. (1) We expanded the amount of post-parse inference that we do and (2) we extended SPARSER's new `repair' phase of analysis to deliberately identify patterns over multiple constituents (e.g. clause + comma + participle). This let us rearrange the initial analysis to incorporate constituents and determine the correct scope of conjunctions, among other things.

Contextual Interpretation

Our discourse component resolves pronominal and definite references using its structured history of entity and event mentions. This is the same facility that organizes searches to expand partial descriptions of entities to full ones and in general to link individuals as they appear in different parts of an article.

Our system distinguishes between the *direct* (or *base*) interpretation of a phrase (which the parser produces simply by applying syntactically guided interpretation rules to the lexical items directly present in the phrase) and the *contextual* interpretation of the phrase. The system first performs a direct interpretation of all of the phrases in a stretch of text (typically a sentence), and then performs a post-analysis pass drawing on the context provided by the model formed from the interpretation of all of the prior text. This produces (in a bottom-up, recursive manner) the *contextual* interpretation of each phrase. There are many components of this contextual re-interpretation, including:

- pronoun resolution (not shown in the examples above, but of critical importance)

- local semantic role disambiguation, e.g. the choice of the *agent* or *substrate* role for *MAPK* in the ambiguous noun phrase "*MAPK phosphorylation sites*"
- conjunction distribution: creating expanded interpretations of phrases that contain conjunctive elements such as "*conserved putative MAPK phosphorylation sites in ASPP1 and ASPP2*".
- elliptical phrase expansion, e.g. interpreting "*T669 phosphorylation"* in terms of "*EGFR is phosphorylated at T669 by MEK/ERK*". Note that this takes into account the equivalence of the normalized interpretations of "*T669 phosphorylation"* and "phosphorylated at T669")

Epistemic structure and tagging

R3 also needed to recognize the larger-scale rhetorical structure in articles. We began the process by cataloging the phrases that carry only rhetorical content. (e.g. "*We reported recently that*", "*However*", "*Here we show that*"). We developed a paragraph-level and sentence-based state machine that provides the inferential basis for judging the epistemological status of mid-paragraph sentences which typically do not incorporate any rhetorical marker phrases. The pattern set and epistemic state analysis machinery were integrated into the analysis pipeline and classified sentences as known, conjectured, or new.

The mechanism works by tracking phrases that indicate whether the statement is intended factually or as conjecture, whether or not it is a reference to a prior work, and its tense. It uses phrases like those in the table below, along with the section of the document the sentence was in, to identify the first two of these:

Discourse Role	Rhetorical Phrases Indicating Role
Conjecture	it is likely that, a possible explanation
Known Result	it has been shown, recent evidence suggests
New Fact	these results indicate, our data suggest that
Methodology	we conducted, we performed
Motivation	to determine whether, we investigated whether

Knowing what section the sentence is from is particularly important when the statement is unmarked and in present tense. For example, with the sentence "*ASPP2 belongs to an evolutionarily conserved ASPP2 family of proteins alongside ASPP1 and iASPP."* Being in the present tense, it is considered to be conveying known information, especially if it is in the introduction.

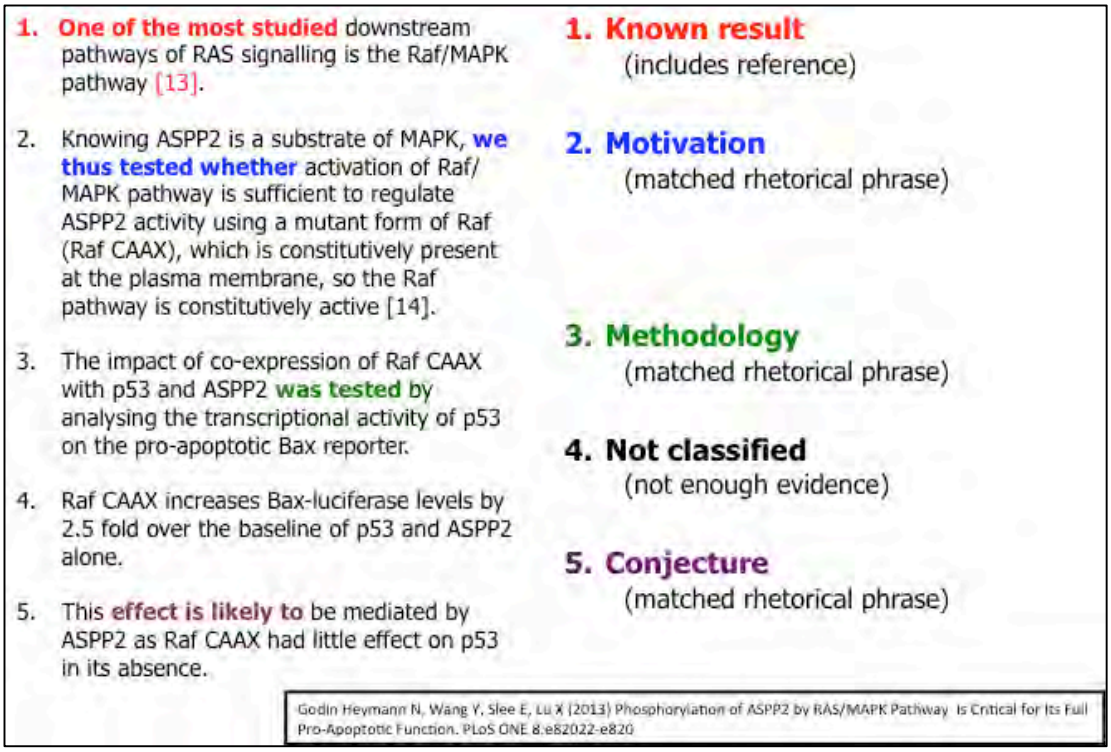


Figure 1: Phrases for identifying epistemic stance of sentences.

Use of R3/Sparser as an additional reader in CURE

During the later years of our work, we did a number of things to improve R3/SPARSER, to provide the CURE system with an additional reader that improved the diversity of the attempts to recognize useful reaction statements that are compared and combined by the CURE assembly process. CURE assembles matching individual statements from different readers to assess the certainty of its model elements. Adding a third reader increased the likelihood of the extractions being accurate. Our work to improve SPARSER for this was as follows:

- We extended the SPARSER lexicon and associated semantic forms with the information used in TRIPS and REACH so that CURE could utilize SPARSER output as an additional source of reading input. This included
    - Unifying the semantics behind our protein vocabulary with those from the other sources, and agreeing with HMS on the choice of identifiers from various standard databases (HGNC, UNIPROT, Reactome) to use for different entities.
- We developed code to produce the sentence output from SPARSER in an XML form that could be consumed by CURE.
- We fixed a number of SPARSER interpretation errors discovered in the process of reviewing the CURE corpus we received.
- We processed a set of 30,000 sentences that had been found by REACH to be relevant to CURE, and identified differences in the information derived by our

analysis, in the process identifying events that had been missed, false positives that REACH had generated, and SPARSER false negatives.

- We subsequently processed over 17,000 articles that constituted the set available in NXML format that CURE was using. Later, after SPARSER was directly incorporated into CURE it was used to read over 1 million articles.
- We developed code to automatically align and compare the outputs of SPARSER, TRIPS and REACH.

Extending/aligning biochemical vocabulary

We used several different approaches to align our vocabulary for proteins and other entities with the descriptions produced by REACH and by TRIPS/DRUM. We began by developing code to compare event descriptions produced by REACH to SPARSER event representations for the same text. Comparing their bio-chemical entities to ours in this fashion allowed us to extend our definitions so we would have comparable results when combined in CURE. We identified entities based on their involvement in sentences describing reactions and got groundings for over 500 additional bio-chemical entities, including cell lines, tissue types, genes, protein families, simple chemicals, and cellular processes through this work. This allowed us to extend our associations for these entities to standard databases including PFAM (Protein Families), and Cellosaurus (a catalog of Cell Lines).

It also became clear that certain lexical items (particularly in Reactome) were specified as being used as the names for several distinct proteins. We took those words and defined them as de-facto protein families (the same word could be taken as indicating any of a specified number of proteins). These de-facto protein families indicate usage patterns by biologists, even though they do not specify a functional or homology basis for the notion of family, nor a standard identifier for the family.

However, we encountered some conflicts with a subset of REACH's protein definitions, especially related to protein family names, and we looked to the TRIPS lexicon for additional guidance. The TRIPS/DRUM TextTagger agent is used by the TRIPS parser to retrieve definitions for new lexical entries. It calls a variety of databases and returns definitions based on how well they match the given term. To facilitate our grounding of terms, we developed a procedure to call the TextTagger with items we believed to be proteins and extracted the definition with the highest match score. By doing this, we gained definitions for 1,770 proteins. As a further step, since we were expecting additional large amounts of data from HMS, we developed a full TRIPS agent that could interact with the TextTagger using KQML and its best definitions for terms. This was used to process the object references in the 30,000 sentences we processed.

As we were anticipating the need to process 30,000 articles, not sentences, we also developed API support and installed SPARSER on our workhorse server so that we could address such volumes of text for the future needs of CURE.

Early in 2017, while we produced a draft set of semantic output on the 27k sentence corpus, we also analyzed some of the differences we were finding between SPARSER output and the REACH output for those sentences. To do this we processed the REACH output JSON files and aligned their output with the text of the sentences. We compared the

SPARSER semantic output with those results. Our results showed that there were a number of cases where:

- We produced events that REACH missed
- REACH produced "pseudo-events" that were false positives (and SPARSER didn't produce the events)
- REACH produced events or had event participants that SPARSER missed

We then adjusted SPARSER's grammar and subcategorization frames to extend our vocabulary and coverage to include the newly discovered problems.

Overall, we discovered through our comparisons that there were cases where REACH was inferring relationships between proteins because they were found in the same sentence, though they weren't directly connected by the text. There were other cases where assertions were missed due to weak co-reference resolution and a few other issues. This gave us some confidence that the addition of the SPARSER output to the other reading methods would help to correct errors during the assembly process.

Shortly afterwards we also worked extensively on resolving issues to effectively run the 17,000+ article corpus for the joint extraction test with Ben Gyori at HMS, and to support localization. This involved relatively little interesting "conceptual" work, but a lot of debugging and engineering.

We developed new tools for adding the vocabulary for all of the proteins involved in the Homo Sapiens part of Reactome. Related to the localization work, we discovered that many of the "proteins" in Reactome were explicitly labeled "activated ___" and that if we used those as "uninterpreted protein names", this caused problems. We then developed ways to extract only the appropriate vocabulary from Reactome. We identified all the likely protein names from the result of parsing the 17,000+ articles, and developed a tool that would add definitions for those entities from the TRIPS lexicon if they were missing. We also made use of Uniprot's API to develop a tool that would vet protein definitions obtained from REACH and HMS against the set of aliases for the given Uniprot ID to avoid some erroneous definitions from those systems. Overall, we added over 10,000 bio-chemical entities to our vocabulary in the last year of the project. We also improved the extraction of the entities and relations used for localization.

Improvements guided by comparisons with other readers

In addition to extending the biological vocabulary, we did some in-depth comparison of SPARSER output against REACH and TRIPS, and dramatically reduced the number of errors in reading the 17,000+ articles. While REACH and SPARSER process articles at a similar rate (SPARSER takes about 0.5 sec/article), REACH over-generates because it does not take as much linguistic structure into account.

We ran the 17,000+ articles through SPARSER and provided the full semantics to HMS. They gave us back some statistics on which assembled statements were developed using information from data sources, and the TRIPS, REACH and SPARSER readers. Reviewing these results revealed that the routine HMS wrote to incorporate the SPARSER output missed the majority of useful statements, especially on recursively embedded statements (where the top-level structure of the sentence did not contain the useful information). A decision was then made to develop a native INDRA output format in order to ensure that the information extracted was properly utilized. This output form was completed in April, after

addressing some vocabulary issues and fixing some mapping problems between SPARSER output and INDRA representations of statements.

In May of 2017, we completed improvements to R3/SPARSER based on our comparison of Sparser output with that of REACH and TRIPS. We then developed a native INDRA formatted output (in JSON) for the extracted content of parsed articles. We also developed procedures to parallelize the running of the large corpus of articles.

At the end of the month, we provided a full revised set of results on the 17K articles to HMS/CURE. These results were included in the assembly process for the next evaluation of the full CURE system in June. The addition of R3 parsed output was shown to improve the overall results of the assembly process, as it reinforced some conclusions drawn by other readers, but also has a different set of strengths than those readers, especially when it comes to some forms of coreference resolution and unification.

We also improved the output of R3 descriptions of active forms gathered from Reactome, also by using a native INDRA object model, so that CURE would have all of the information that R3 collected about active forms from the full human Reactome model that we read. Of the ones supplied, 23 active form descriptions were used to improve the utility of statements acquired by reading and support the development of a coherent model. These too were reported in the June CURE evaluation report.

We delivered INDRA-natively formatted JSON containing the information SPARSER extracted from the 17,000+ articles. The CURE assembly process now combines that information with that produced by REACH and TRIPS, and some additional data sources that HMS incorporated. The addition of R3 parsed output improved the overall results of the assembly process, since it reinforced some conclusions drawn by other readers, while producing fewer false positives than REACH, so it tended to reduce the certainty of some conclusions drawn from that reader.

We did a more careful comparison of the SPARSER, TRIPS and REACH results on this large corpus, implementing some support tools that enable us to compare the precise passages that were the source of the “cards” each produced, and to align entities even if they were extracted with slight differences in grounding or in the text spans that each demarcated for those entities. Results of this assessment on a set of 800 articles are:

- REACH and SPARSER agree on 16,538 event extractions,
- SPARSER has 15,509 extractions that don't (exactly) match REACH, and
- REACH has 11,800 that don't match SPARSER

Looking at the head words for the unmatched events, SPARSER has 10,787 events that are not picked up by REACH, while REACH has 6,034 that are not seen by SPARSER. By sampling 10% of those REACH items not identified by SPARSER and examining them manually, we estimate that somewhat less than 30% of those REACH extractions were correct. Similarly sampling the ones that SPARSER found that REACH did not, we estimate that >80% were correct. (Note that these estimates are from relatively small samples of the 1000s of events.)

We used some of the information discovered in doing these comparisons to improve the quality of SPARSER’S output by fixing interpretation errors it had produced. These were reflected in the next round of contributions to CURE from R3.

R3 Reading and Localization Architecture Summary

R3's architecture and information flow is shown in Figure 2. Before processing complete biology articles, R3 bootstraps its domain model by inferring and representing events and segmenting its input biology model (typically a subset of Reactome) to facilitate retrieval. R3 then parses the English textual labels and comments embedded in its biology model which were written by biologists. These textual labels and comments describe events (e.g., activation, deactivation) and properties (e.g., active and inactive states) that are not represented directly in the formal structural model. To learn by reading, R3 uses structure-mapping to match the parser's semantic interpretation against its domain model. It uses analogical inference to transfer knowledge into the model and then it propagates the information throughout the model, where relevant. R3 then reads biology articles using the same semantic interpretation and structure-mapping processes: it parses the article to extract the semantics of entities and events, and then it matches those semantics against events and entities represented in its model. This computes the locale of parsed entities and events within the model for the purpose of bidirectional knowledge transfer.

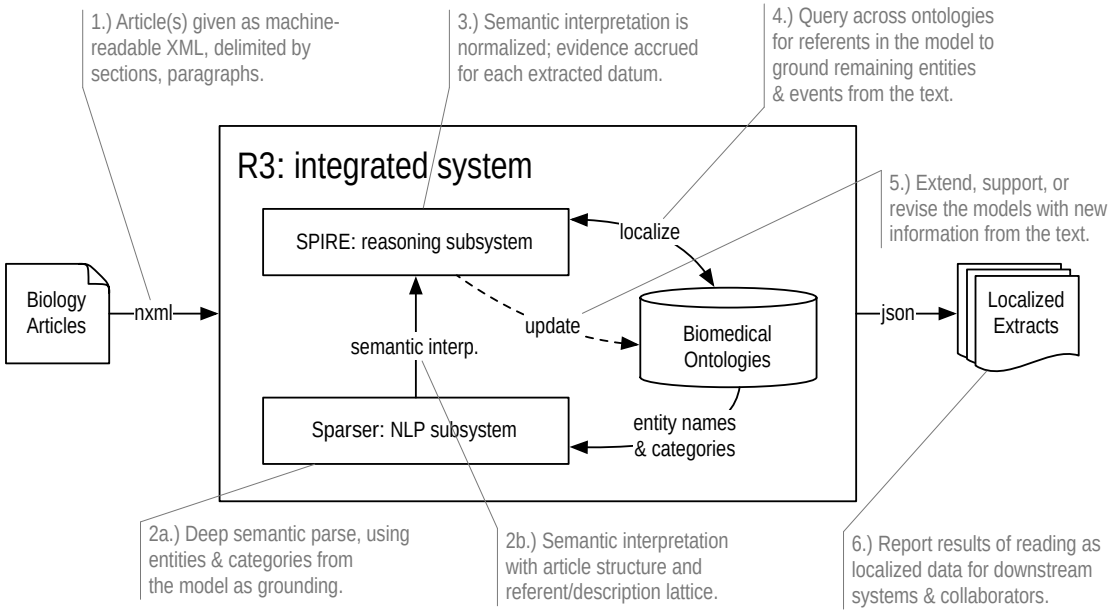


Figure 2: The R3 architecture, and the flow of information by which R3 reads articles, updates its mechanism models, and publishes extracted knowledge for human and machine collaborators.

3. R3 Approach to Localization of References to a Model

As we said earlier, a key issue for Big Mechanism is the problem of reading with a model. That is, determining whether the descriptions in the scientific article being read refer to known parts of the prior model, putting us in a position to identify properly how the model is extended by new information that is related to the known information. Since authors tend to use only partially complete descriptions to refer to information assumed known by their readers, it is important to try to use context wherever possible to not just form coherent semantic descriptions from text, but to see what those descriptions might be referring to. Only then can we properly place the new information into that same context.

Our initial work on localization of model fragments (typically single types of events) during year one consisted of parsing articles and attempting to directly match semantic descriptions to known events and entities in a large many pathway model extracted from the Reactome portion of Pathway Commons (reactome.org). Reactome contains a large network of reactions represented using a small OWL ontology created for a language called BioPAX. There were several key kinds of mismatches between the semantic representations from parsed texts and the representations of chemical reaction in the model's BioPAX ontology. Some of these differences had to do with the explicit terms used for kinds of reactions/reaction products, such as "activation", "inhibition", etc.

It became clear that articles often talk about processes at a functional level, in terms of triggers for and preventers of events or event sequences. Typically, in the signaling domain, processes are "switched" on or off, and proteins can "activate" or "inhibit" processes. Proteins are described as in an "activated" state when they are bound to other molecules in ways that enable them to act as catalysts in subsequent reactions.

This kind of association of functional states with molecules is so common that, in many cases, localization by matching is not effective unless the underlying model represents these molecular states explicitly. What makes a protein "active" can be many different things at the structural, molecular level. For example, proteins like MEK and ERK are *activated* when they are phosphorylated. Others are *deactivated* when they are phosphorylated. Some are activated when they are dimerized, etc.

To capture the specific conditions corresponding to activation for particular proteins in our model, we needed a source of information about the complexes involving those proteins when they were considered "active" or "inactive". We found that much of that information was available in the comments that curators had written for individual Reactome reactions or in the labels of complexes that contained the active forms of proteins. During year two, we decided to try parsing the textual summary statements associated with each reaction in the Reactome model as written by the original curators, so that we could identify what in the molecular descriptions was being referred to as active or inactive in each case. Some examples:

- "SOS1 is the guanine nucleotide exchange factor (GEF) for RAS. *SOS1 activates RAS nucleotide exchange* from the *inactive form* (bound to GDP) to an *active form* (bound to GTP)."
- "EGFR phosphorylates PLC-gamma1, *thus activating it*."
- "*Activated MAPK proteins* negatively regulate MAP2K1:MAP2K2 heterodimers..."

These examples of reaction summaries show how the language used for human consumption conveys the functional states of their primary participants, rather than the chemical changes represented in the underlying formal BioPAX model. R3 learns what it needs for localization by matching the semantic interpretations of those descriptions against the associated model of the chemical reactions.

Enhancing BioPAX models with forward inferences for improved matching

Much of our work during the first two years was with the subset of Reactome that covers all of EGFR related signaling.[7] This includes as a sub-pathway the RAS/MAP kinase cascade.[8] During year three, we extended R3's coverage to all of Human Reactome, and extended localization process so that it can now identify references to pathways and other phrases that refer to more than single reactions.

Using our SPIRE inference engine, we enhanced the BioPAX model by running inferences that are similar to the ones we used in year one to embellish BioPAX signaling models for model comparison during card generation. These inferences do such things as make explicit the notation of post-translational modifications (as used for card creation), dimerization, translocation, etc. so that they are easier to identify during localization.

Native BioPAX is very expressive, and Pathway Commons contains valuable BioPAX content; however, BioPAX is not optimized for automated reasoning, e.g., using inductive, deductive, or approximate graph-matching techniques. During year two, we extended the BioPAX encoding by extending its OWL ontology with additional categories and relations. We colloquially refer to this extended encoding as *BioPAX+*. Figure 3 shows some of the extensions in BioPAX+ that enable it to capture functional relationships. Activated entities can activate or inactivate processes, and processes can produce products and other observable outcomes.

R3 analyzes the native BioPAX (one of its inputs) and encodes BioPAX+ categories for proteins with post-translational modifications and other commonly referenced protein classes and relations. Figure 4 shows a model of SOS-activated Ras nucleotide exchange. Native BioPAX (the input) is shown in black, and the automatically-inferred BioPAX+ (inferred by R3) is shown in blue. Some native BioPAX is hidden for clarity. R3 makes the following inferences:

- It labels complexes with repeated substructures **heterodimer** instances. In the native BioPAX, this was only evident from looking at protein-states and their stoichiometric coefficients.
- It detects phosphorylations on different proteins, reifies them as (e.g.) **phosphorylation-effect** and **tyrosine-effect** instances, and links them to the numeric phosphorylated sites. In the initial BioPAX, this was only evident from looking at the modification-type string, e.g., *04-phospho-l-tyrosine*, to determine that this is a tyrosine phosphorylation.

[7] http://www.reactome.org/PathwayBrowser/#/R-HSA-177929
[8] http://www.reactome.org/PathwayBrowser/#/R-HSA-5673001

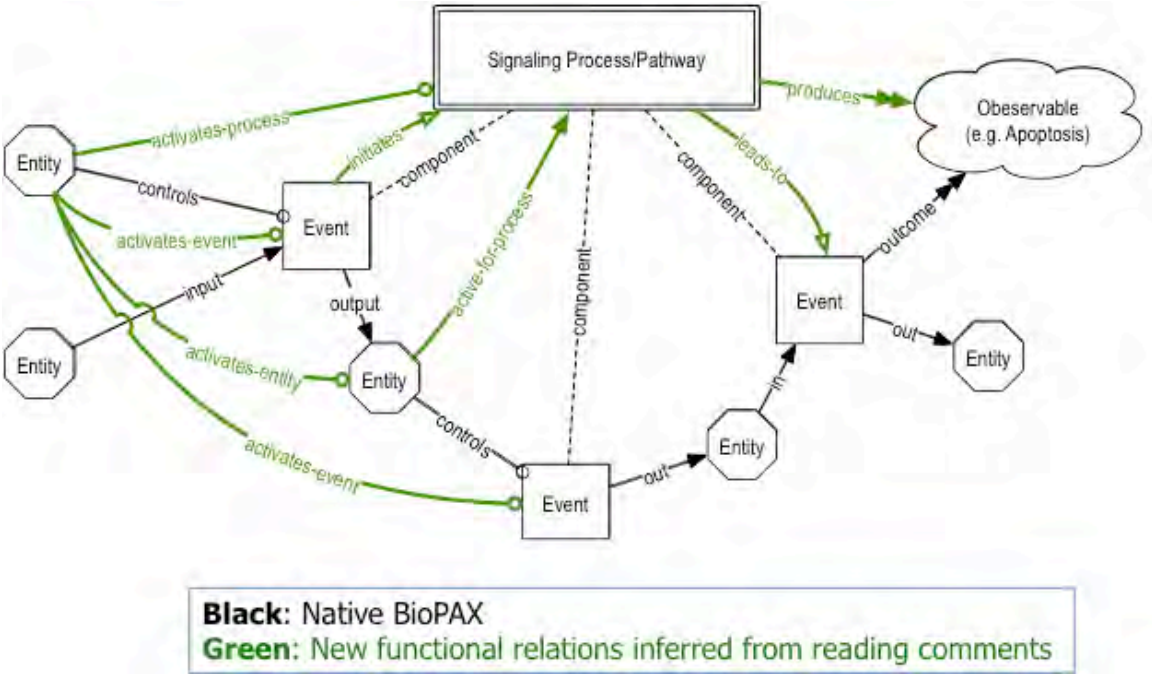


Figure 3: Extending the ontology of BioPAX

- It detects a nucleotide exchange and reifies a **nucleotide-exchange-effect** as a result of the reaction, and it relates it to the input/output nucleotides (GTP and GDP) and the input/output complexes (RAS:GDP and RAS:GTP).
- It encodes a **catalyzes-event** relation between the catalyst and the reaction.

All of these additions to BioPAX facilitate querying, graph-matching, and ultimately, localization during reading to support reading-against-a-model, because it brings the representations closer to the way people talk about those structures. It makes explicit the following events/effects and their relationships with reaction participants: association, dissociation, nucleotide exchange, nucleotide, homodimer, heterodimer, modification types (incl. phosphorylation, ubiquitination, acetylation, etc.), modifying amino acids (incl. tyrosine, threonine, serine, histidine, lysine).

Learning functional descriptions from Reactome comments

Biologists typically write about "activations" of proteins within pathways, since these proteins have pathway-specific downstream functionality. This "activation" language permeates biology articles, and it also permeates the English content within Reactome—including the *displayName* and *comment* fields— written by human expert curators. Figure 5 from Reactome shows a comment (Summation) from which we can conclude that EGFR is treated as active once it is dimerized.

The next step in R3's process infers which proteins within complexes are functionally referred to as the active or inactive states of those proteins. For example, which protein states are active kinases that trigger other events or outcomes. These functional states are not easily inferable from BioPAX OWL, because 'active' proteins can be characterized by many different properties, and BioPAX explicitly avoided such labels because activation can mean different things structurally in different cases. However, scientists refer to

proteins by their activity state all the time and we need to know which ones are so referenced in order to properly localize them to chemical states in the model.

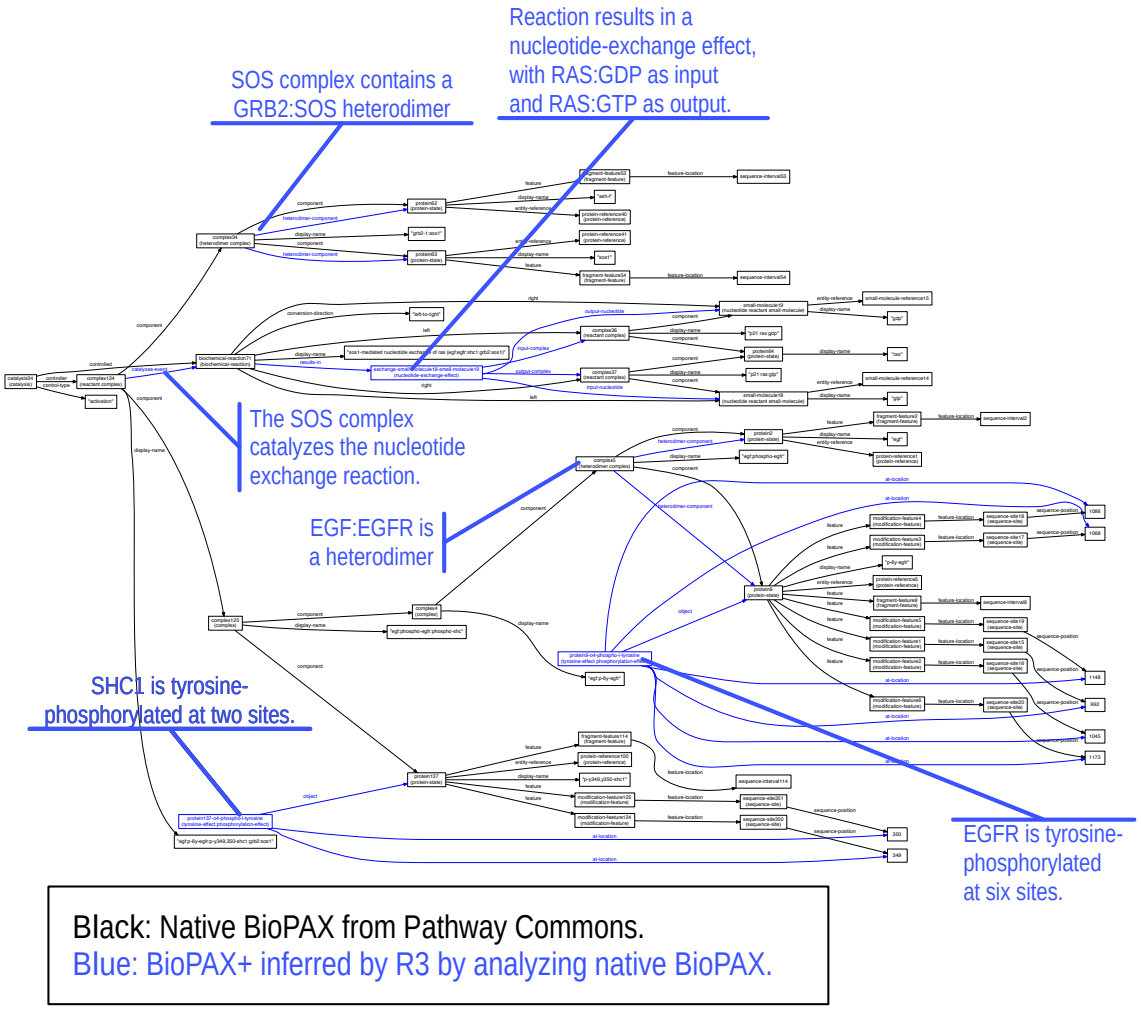


Figure 4: Enhancing BioPAX

Fortunately, the curators of BioPAX tended to refer to the elements in their model the way they would otherwise talk about them, and so we can glean from the comments associated with each reaction and complex which ones are ‘active’ and which ‘inactive’. Making those descriptors explicit as a functional level of our model by mining the comments was what we achieved during year two. Figure 5 shows an example of a piece of the Reactome model and a summary comment on one reaction within that.

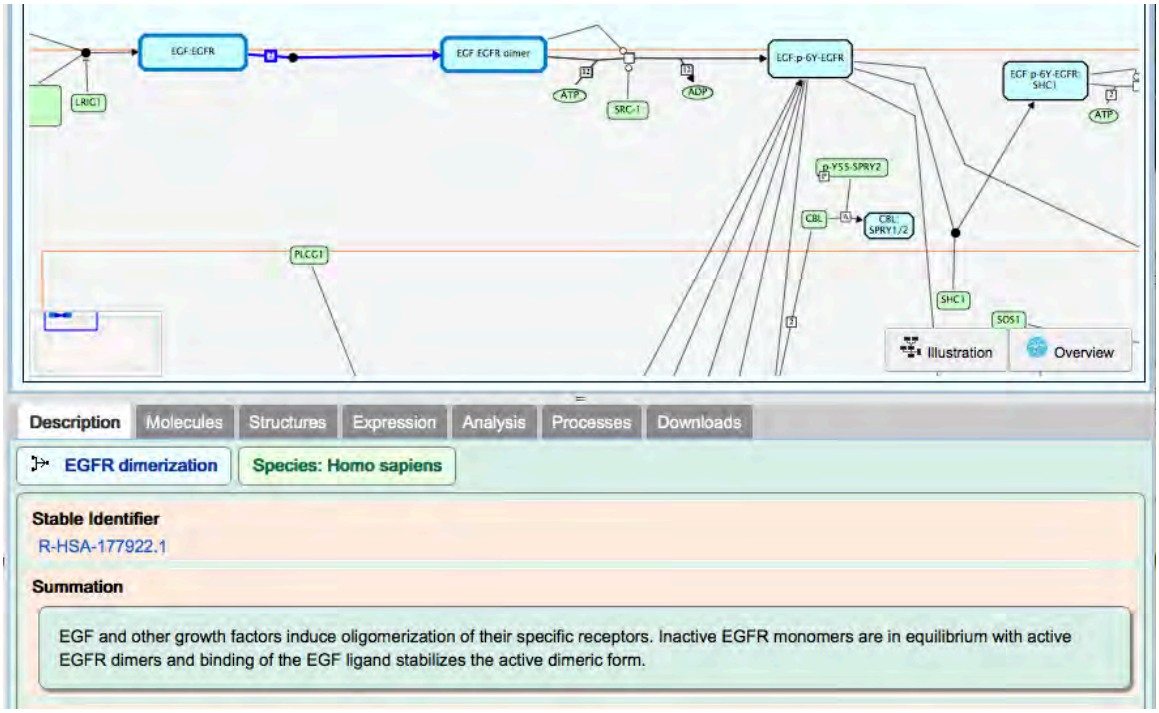


Figure 5: Reaction view from Reactome browser (EGFR dimerization is the reaction name and the Summation field at bottom is the comment on the reaction)

During this step in R3's processing, the BioPAX+ re-representation of a Reactome model is used as the background model (without the functional descriptors that this step will add). A semantic parser (SPARSER or DRUM) reads the comments on a complex or reaction and then R3 compares it to the formal BioPAX+ representation for that entity using approximate graph-matching to find maximal common subgraphs between the semantic interpretation and the model. These graph alignments enable R3 to identify where *new* information from the semantic interpretation can extend the background model.

R3 extracts the activation knowledge (included related causal effects and conditions) from comments and display-names and uses it to extend the representation of BioPAX+ to identify active and inactive states, and activating and de-activating events.

Curator comments with "activation" language include the following:

1. "SOS1 is the guanine nucleotide exchange factor (GEF) for RAS. SOS1 activates RAS nucleotide exchange from the inactive form (bound to GDP) to an active form (bound to GTP)."
2. "EGFR phosphorylates PLC-gamma1, thus activating it."
3. "Activated MAP2K phosphorylates MAPK on threonine and tyrosine residues in the activation loop…"
4. "Activated MAPK proteins negatively regulate MAP2K1:MAP2K2 heterodimers…"
5. "…Inactive EGFR monomers are in equilibrium with active EGFR dimers and binding of the EGF ligand stabilizes the active dimeric form."
6. "Active PLCG1 hydrolyses PIP2"
7. "Calmodulin activates Cam-PDE 1"

8. "MAP2Ks and MAPKs bind to the activated RAF complex"
9. "EGFR activates PLC-gamma1 by phosphorylation"

These texts implicitly include things like the relationship of the before and after conditions in the activation described in (1), the effects of activation in (2), (3) and (6) described by the pattern "activated <x> <does y>", and the identification of phosphorylation with activation in (2) and (9).

Extending the BioPAX ontology with extracted function/activation knowledge

After producing a deep semantic parse of a comment, R3 learns from the recognized events and entities. Using ontology mapping, R3 re-represents the parsed description as BioPAX+ in order to directly compare it to the structural model from Reactome, which is also now represented in BioPAX+. Since the article and the model may represent events with different granularities, the SPIRE ontology-mapping rules must occasionally generate new event symbols to represent the mismatch. For example, if R3 reads, "X phosphorylates Y and Z," it must create separate phosphorylation events for Y and Z, with X in the agent role of both, in order to localize them independently: these may or may not correspond to the same event in the model.

Once the text and model are made comparable, R3 uses SPIRE's graph-matching capability for two central operations in model localization:

1. **Retrieval**: Given a semantic description extracted from text, recognize and retrieve all potentially corresponding entities and events from the model.
2. **Transfer**: Given a semantic description extracted from text and a description of an entity or event from the model, match the two descriptions and suggest the transfer of entities and relations into the model.

The core graph-matching operation involves computing one or more mappings between two representations. Each mapping is a maximal common subgraph (MCS) between the two representations, where each entity is a node, each relational assertion is a node, and each relational argument is a position-labeled edge. Each of SPIRE's MCS mappings describe correspondences (i.e., tuples describing isomorphic nodes across graphs), a score that rates the quality of the correspondences, and inferences describing complements of the MCS (i.e., non-isomorphic relations and entities) that can be projected from one graph to the other.

Graph-matching inferences are not necessarily deductively sound, since they are based solely on structural similarity; however, in previous work, we have shown that these inferences can be practically used to revise beliefs and models (Burstein, 1988, Friedman et al., 2012). As we illustrate below, graph-matching inferences are practical for extending the model while reading. They reduce the space of legal mappings—thus making the problem more tractable than traditional MCS optimization problem—by adding two additional constraints:

1. **Identicality**: Category nodes can only correspond to other category nodes with identical categories, and relation nodes can only correspond to relation nodes with

identical predicates. This allows symbol arguments (e.g., referring to entities or events) to correspond to non-identical symbols.

2. **Parallel connectivity**: If two relation or category nodes correspond, their arguments must correspond, in sequence. Applied globally: if two nodes correspond, so must their reachable subgraphs.

These two constraints drastically decrease the solution space, so SPIRE's greedy MCS algorithm is plausible and effective. Guaranteeing an optimal MCS solution is out of scope for R3 due to tractability: the decision problem for MCS is widely known to be NP-complete. As we demonstrate below, a greedy algorithm produces practical results for R3's model localization.

After parsing and mapping a comment or a display-name of a biochemical reaction, R3 then performs constrained graph-matching to find literal similarity—and not just a structural analog—between its BioPAX+ representation of the parse and the BioPAX+ model of the reaction. We demonstrate this with the parse of this comment:

"SOS1 is the guanine nucleotide exchange factor (GEF) for RAS. SOS1 activates RAS nucleotide exchange from the inactive form (bound to GDP) to an active form (bound to GTP)."

Figure 6 shows this comment as a description of the highlighted reaction in the Reactome browser.

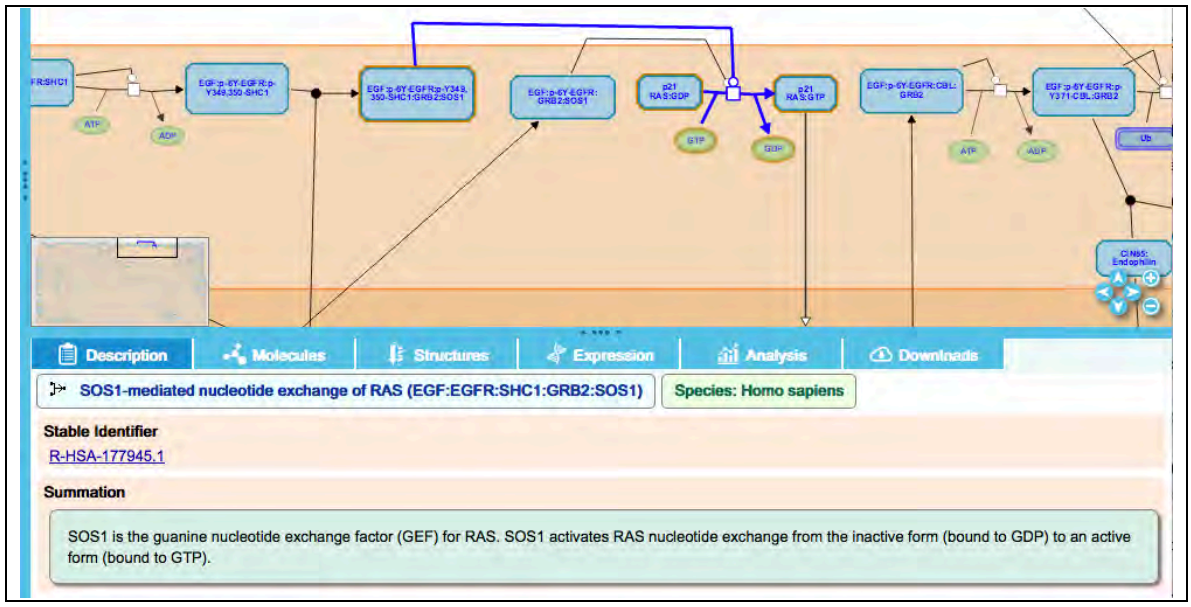


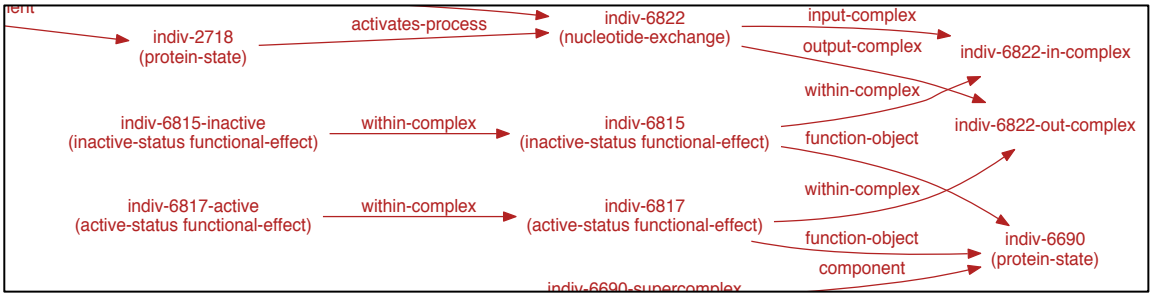
Figure 6: Reactome view of reaction (highlighted blue edges) with Summation comment parsed by R3

The matching process maps the parsed "RAS nucleotide exchange" event to the nucleotide-exchange event in the model, and other entities/events accordingly. R3 identifies unmapped entities and relations from the parse, which can be transferred via a graph-projection into the model.

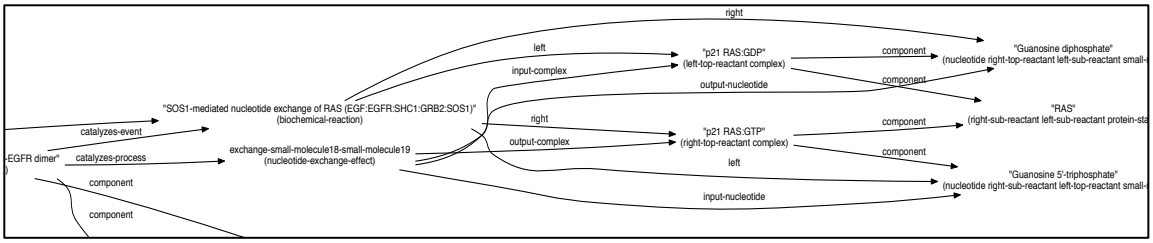
The result is shown in Figure 7, below, with the native BioPAX shown in black, the inferred BioPAX+ shown in blue, the overlapping (i.e., isomorphic) entities and relations from the parse and the model shown in purple, and the newly-imported knowledge from

the parse shown in red. The semantic representation from the text (Figure 7a) and the corresponding portion of the BioPAX model (Figure 7b) are the inputs to graph-matching, which computes the maximal common subgraph (shown in blue in Figure 7c, using the symbol names from the model). The complement (i.e., non-isomorphic portion) of the semantic interpretation provides graph-matching inferences (shown in red in Figure 7c) that R3 transfers into the model.

a.) Output of Semantic Parse



b.) Existing Event in Model



c.) Extended Event in Model: (**model complement, isomorphism, parse complement**)

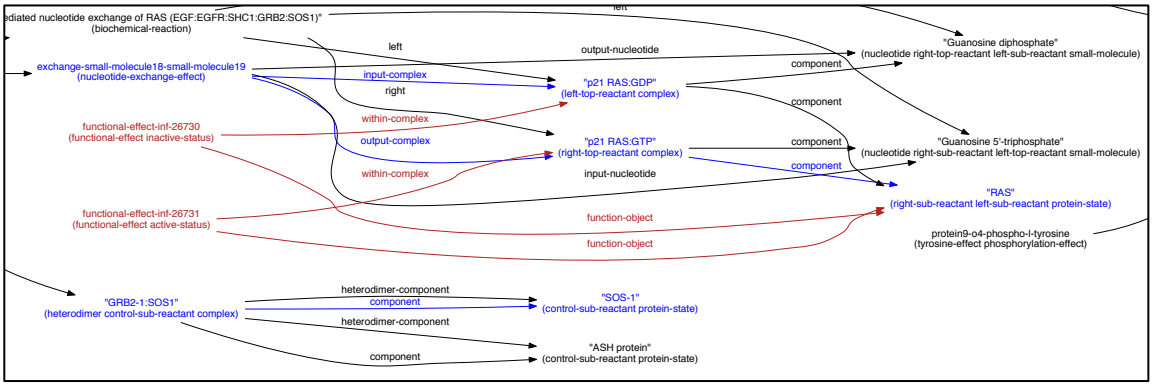


Figure 7: Information (red) added to the model from a comment.

The interpretation includes the following statements (and others) about a nucleotide exchange and a RAS protein:

```
(isa indiv-6822 nucleotide-exchange)
(input-complex indiv-6822 indiv-6822-in-complex)
(output-complex indiv-6822 indiv-6822-out-complex)
(name indiv-6690 "RAS")
```

These relations and events were not previously described in the BioPAX, but are now present in R3's BioPAX+. Further, the information about active and inactive RAS (i.e.,

protein64 within *complex37* and *complex36*, respectively) can be generalized beyond the context of this reaction, so that R3 can resolve "active Ras" to *protein64* within *complex37*, or any other nominally active form of RAS in the BioPAX+.

R3 concludes that the nucleotide exchange event as parsed in the text is isomorphic to a known nucleotide exchange event in the model. Also, the protein, input, and output complexes of this event in the text are isomorphic to the respective protein, input, and output complexes of the event in the model. The semantic interpretation also contains novel (i.e., non-isomorphic) information that the RAS is inactive in the input complex of the nucleotide exchange and is active in the output complex:

```
(isa indiv-6815 inactive-status)
(function-object indiv-6815 indiv-6690)
(within-complex indiv-6815 indiv-6822-in-complex)
(isa indiv-6917 active-status)
(function-object indiv-6817 indiv-6690)
(within-complex indiv-6817 indiv-6822-out-complex)
```

The isomorphic structure (shown in blue in Figure 7c) provides a scaffold to transfer this novel information (shown in red in Figure 7c) into the model, importing new categories and relations describing existing entities and events in the model, and generating new symbols for novel events and entities. R3 thus transfers inferences that describe protein function and behavior, such as active and inactive forms of proteins, and processes that activate and deactivate proteins. By generalizing from these results, R3 learns that p21 RAS is active when bound to GTP and inactive when it is bound to GDP.

Propagating learned information throughout the model

The protein complexes that R3 learns are active forms can each occur in many places in the model. R3 ensures (somewhat conservatively) that the functional descriptions it associates with these structures are automatically propagated to each such reference in the knowledge base so that, e.g. all of the times that the same "active protein" catalyzes a reaction are labeled.

Consider the following curator's comment on a reaction in Reactome: "*Binding to activated RAS stimulates the ubiquitinase activity of BRAP, promoting autoubiquitination and relieving the inhibition of KSR1*." This interpretation of "*activated RAS*" allows R3 to make the following extensions to its BioPAX+ Pathway Commons model:

```
;; There is an active form…
(ISA inferred-status ACTIVE-STATUS)
;; …of protein64 (p21 RAS)…
(FUNCTION-OBJECT inferred-status protein64)
;; …when it is bound within complex37 (p21 RAS:GTP).
(WITHIN-COMPLEX inferred-status complex37)
```

Note that there was no mention of p21 RAS:GTP within the comment, but the protein entity isomorphic to the "active RAS" interpretation was p21 RAS, which happens to be bound to GTP in this specific reaction. R3 makes the *conservative* inference that the "activated" protein RAS was active in this *specific* context (i.e., bound to GTP), since p21 RAS is not necessarily active in isolation. R3 stores these inferences about `protein64` within `complex37` alongside the rest of the imported BioPAX that describes the reaction.

That way, when R3 reads references to this reaction, such as "*BRAP binds active RAS*," it will have the knowledge necessary to resolve to this reaction.

Importantly, the information gleaned from the comment we processed is more general than this singular reaction, so R3 subsequently propagates the inference throughout the model, to (1) descriptions of complexes that contain `complex37` as a subcomponent (and thus `protein64`) and (2) descriptions of reactions that contain `complex37` (and thus `protein64`) as a reactant, controller, or component. In total, R3 applies this single inference to 16 other reactions, and 15 other (containing) complexes within the EGFR subset of Reactome.

Initially, during year two, R3 found 19 comments or display-names of reactions within the EGFR signaling portion of Reactome that contained information about activations.[9] From these, it made 19 extensions to the functional knowledge (i.e., active or inactive forms of proteins) about RAS, RAF, MEK, ERK, SRC, SHP2, EGFR, Cam-PDE, and PLC1. For all of these, R3 extracts and labels the "active" forms for RAS and RAF. It also extends the model with "inactive" forms that were explicitly stated in the comments and display names, e.g., "*PAQR3 binds inactive RAFs*." These inferences, after propagation, affected 182 reactions and complexes in the entire EGFR signaling subset of Reactome we have been working with, which contains a total of 128 reactions and 911 complexes. Many of these 182 complexes and reactions contained *multiple* functional inferences, since they describe multiple notionally "active" components: active RAF phosphorylates (activates) MEK; active MEK activates ERK; etc. Later, we discuss the impact that this had on our ability to localize mentions in an article in our section on evaluation.

Finding Activation/Deactivation cycles

R3 can also trace the activity of these proteins through different events, from where they are activated to where they are deactivated, in a cell-location specific fashion. Essentially, R3 traces the reaction chart with the functional knowledge in-hand, and ignores reactions where nothing interesting occurs (i.e., no activation or deactivation or translocation). We call this the *functional event structure* of the protein, and we use our knowledge representation based on Pustejovsky's Generative Lexicon (GL) theory to represent this structure so that R3 can localize and reason about entire sub-pathways of activity.

Once all of the active and inactive forms of proteins referenced in the summaries are identified, R3 analyzes each active protein to generate GL-compatible definitions of the entities that include specifically their *affordances,* the things that they can participate in when in an active state. This involves identifying their structural preconditions, location preconditions, and molecular binding preconditions for "active" status. In total, R3 generated 14 entries to describe active variants of RAS (3), RAF (4), MAP2K (3), and MAPK (4). There were multiple affordances recorded for each protein family, since within the EGFR signaling subset of Reactome, "active RAS" refers to GTP-bound HRAS, KRAS, and NRAS of the RAS family. Similarly, "active MAP2K" can refer to a phosphorylated homodimer of MAP2K1, a phosphorylated homodimer of MAP2K2, or a phosphorylated MAP2K1/MAP2K2 heterodimer. R3 collapses these across protein location: Reactome duplicates proteins across cellular locations, e.g., in the cytosol, at the

[9] A sample of a BioPAX reaction display-names with "activation" language is: "Phospholipase C-gamma1 binds to the activated EGF receptor".

plasma membrane, etc. R3 automatically fused duplicates in this work. It then uses the functional knowledge in its lexemes to build an event graph of protein activation and protein function, as shown in Figure 8 below.

The event structure describes the active and inactive forms of molecules across cellular locations, as well as the biochemical-reactions (the circles with "R") that activate, deactivate, and translocate them. The triangular arrowheads indicate input and output reactants to the reactions, and the circular arrowheads indicate direct regulatory relationships— such as catalysis— between entities and reactions.

Figure 8 shows the automatically-generated R3 event structure of active RAS, RAF1, MAP2Ks, and MAPKs once it has assimilated the comments and reaction structure of the "Signaling by EGFR" subset of Reactome. This takes R3's functional knowledge a large step toward properly localizing (and thus understanding) what articles actually *mean* when they say "MAPK activity." This automatically-generated event structure graph is a very small subset of R3's original BioPAX model, and it closely resembles the well-known RAS-RAF-MAP2K-MAPK activation cascade. It should be noted that some proteins with known active states were not found to have active states from comment reading, because no annotator mentioned it. For example, SOS-1, is not referenced as being "active" in any textual Reactome summary, despite having an "active" form in the wider literature, so R3 did not learn any functional knowledge about these proteins.

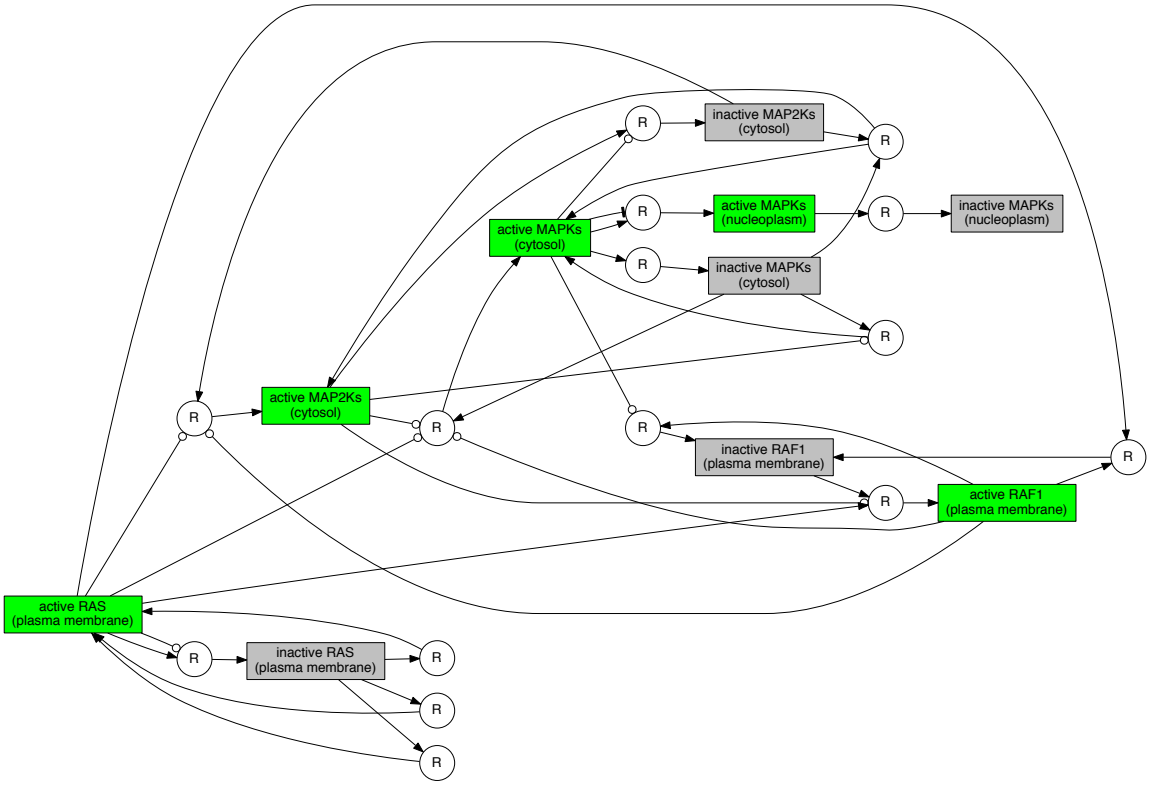


Figure 8: Generalized activate/deactivate relationships

Figure 8 also shows how Active MAP2K is deactivated in a reaction where active MAPK is a control (catalyst). By following the sequences of reactions resulting from a protein becoming active, we can determine the set of things that might be referred to as

consequences of the activation, and thus identify the set of related events that might be the subject of subsequent discussion in a research paper following the phrase that was localized. The green and grey boxes denote the explicitly mentioned protein variants described as active or inactive.

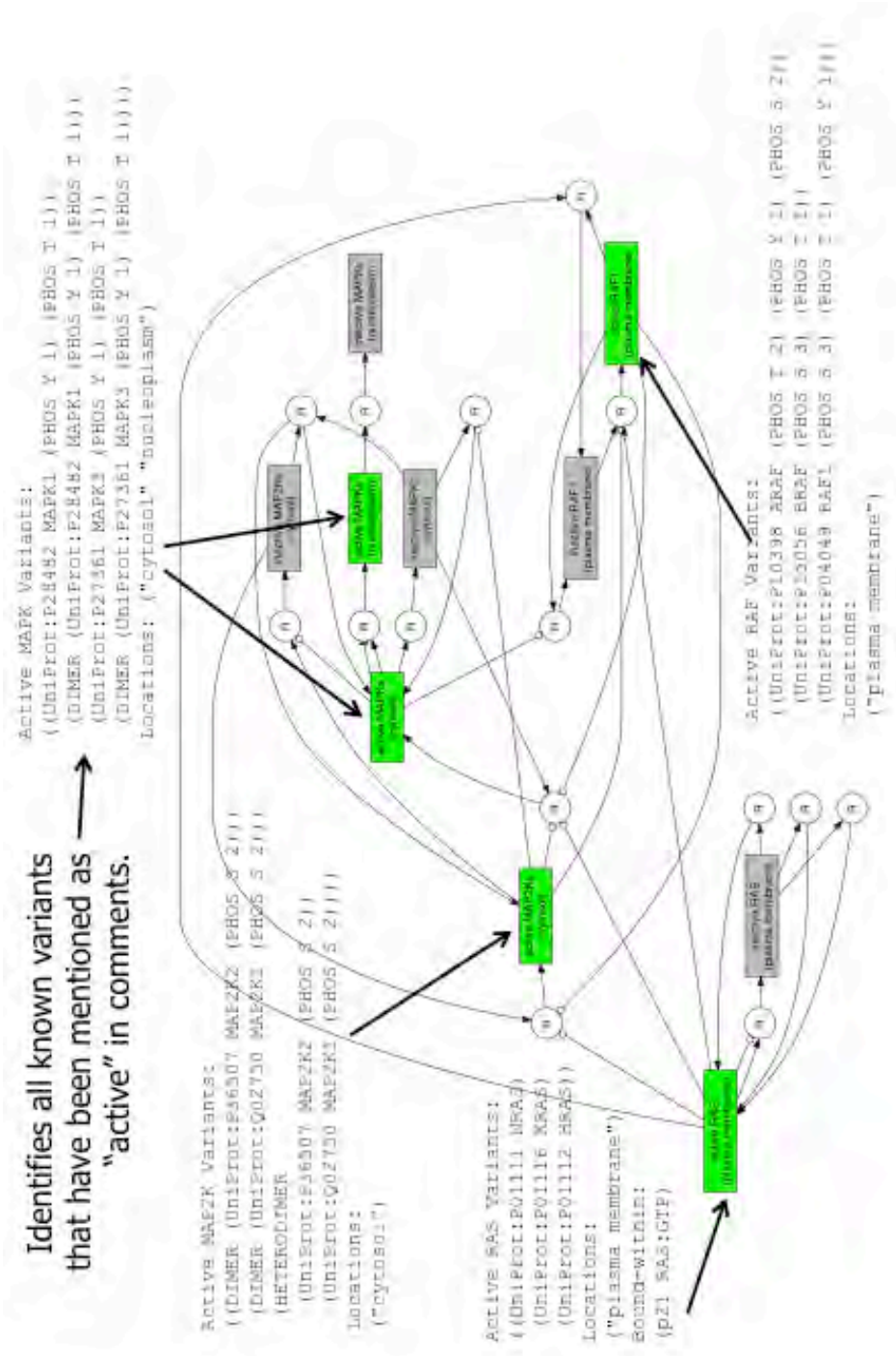


Figure 9: Detailed structural information captured about each active form

Figure 9 shows the same figure, but with sets of specific complexes noted as possible referring forms for the represented state ("active" or "inactive") of each protein or family. This information is critical to establishing precisely what protein complexes might be referred to in a particular context. This is the kind of information that we provided to the Harvard CURE group for use during assembly.

Matching for localizing referring expressions in text

A process very similar to the one described above that learns active forms from comments on model reactions is used to localize referring expressions in articles to events or complexes in the now enhanced model. That is, given a semantic parse of sentences from an article, we identify entities in the model to which the textual descriptions most likely refer. Importantly, localization is not limited to a *single* reaction or molecule; rather, our empirical results suggest that authors often refer to entire sets of molecules such as "active MAPK" or "active MAP2K," where both of these actually refer to a heterogeneous set of individual proteins, heterodimers, and homodimers in the model.

The set of all molecules and reactions from our BioPAX model fragment (the EGFR signaling subset of Reactome) against which we first demonstrated localization is shown in the Figure 10 to provide a sense of the scale of the set of possible retrievals. Each node in the graph is either a molecule (blue) or a reaction (green), and edges between them indicate participation in reactions, set-to-member relations over molecules, and component relationships of complexes.

Though analogical matching has been used widely to perform data fusion and forms of abductive inference, event recognition and localization requires much tighter matching than with analogies: R3 should *not* retrieve events that are *similar* to an event described in a scientific article; rather, R3 should retrieve descriptions that could refer to the *same* events. This is a problem of partial matching for *recognition* rather than one of *analogy*.

Typically, texts will talk about specific proteins playing roles in various reactions or causal processes when in fact, and in the underlying model, these proteins are in various states of binding with other, unmentioned molecules forming complexes. Hence, it is critical that the structure matching be able to identify the states of these proteins in these larger complexes as reactants, products or catalysts when relating the extracted text semantics to the model.

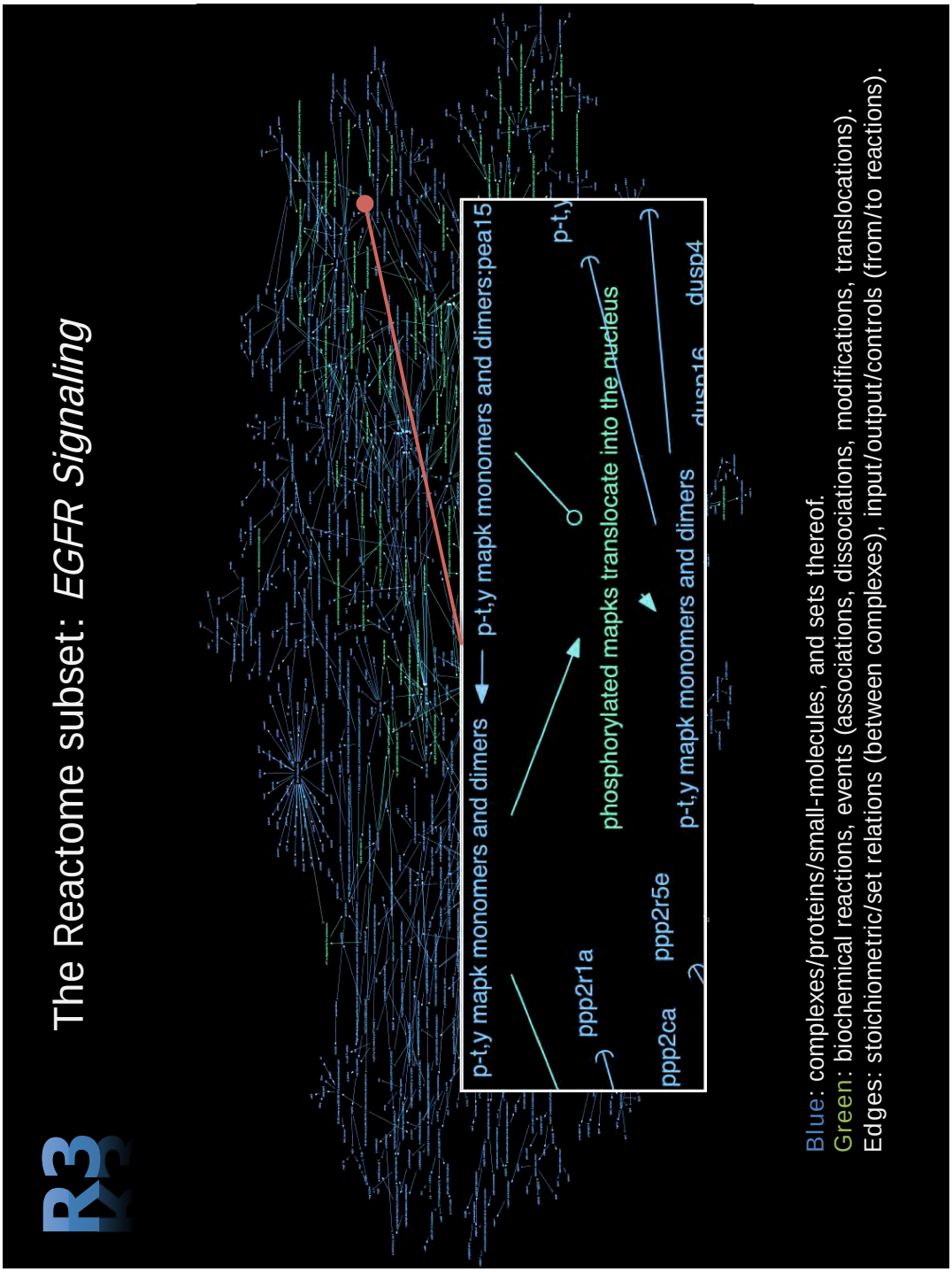


Figure 10: Overview of EGFR subset of Reactome

The matching process for localization required the extensions outlined below.

- **Identifier intersection**. Like any graph-matching optimization algorithm, if SPIRE's maximal common subgraph (MCS) can add a correspondence to its mapping, it will. This maximality bias yields higher-scoring mappings, but it can also produce erroneous results in an entity or event recognition setting. For instance, without additional constraints, the event "SOS1 activates RAS" will map nearly perfectly to the event "MEK activates ERK"; but, this is undesirable, so an additional constraint requires that entities must plausibly co-refer. This allows the parsed individual whose identity was established by namestrings such as {"SOS1," "SOS1 HUMAN," "SOS-1"} to correspond to the model entity with namestrings {"UniProt:Q07889 SOS1," "SOS1," "Son of sevenless protein homolog 1"}given the "SOS1" intersection. This significantly increases recognition accuracy and reduces the search space for mappings.
- **Dependency constraints**. Adding constraints on entities during mapping— such as only permitting two phosphorylation events to match if the phosphorylated entities also match— reduces erroneous mappings. We use a domain-general mechanism for specifying and mapping with dependencies, but with domain-specific rules for asserting these dependencies, e.g., properties depend on their object role-filler. During the mapping process, when the object role-filler is selected for the mapping, the events are added to the search space.
- **Category and predicate subsumption.** If R3 reads, "SOS1 activates RAS nucleotide exchange", it will assert **`(activates-process txt-SOS1-ent txt-RAS-NE-ent)`** to describe this relationship between the SOS1 referent **`txt-SOS1-ent`** and the nucleotide exchange referent **`txt-RAS-NE-ent`**. However, in the corresponding reaction in the model, R3 has described this relationship with greater specificity since SOS1 is a subcomponent of the catalyzing complex e.g**. `(catalyzes-process-as-component mdl-SOS1-ent mdl-RAS-NE-ent`**). The matching must allow for this mismatch in level of specificity between functional activation and biochemical catalysis.

In R3's relational hierarchy, the **activates-process** relation from the text is a superordinate relation of the **catalyzes-process-as-component** relation in the model. SPIRE's graph-matching algorithm supports non-identical relation matches and non-identical category matches albeit with a diminished score. The score is computed as the *Jaccard index* of the predicates' or categories' superordinate locales, which we define as the set of superordinate predicates or relations reachable in an upward walk of constant length k. The Jaccard index between locales is computed as $|(A \cap B)| / |(A \cup B)|$, so it is 0.0 (i.e., not allowed) for nonintersecting locales, 1.0 for identical locales (i.e., identical predicates or categories), and within the interval (0, 1) for non-identical predicates with intersecting locales. For R3, we use a locale distance of k = 3, including the relation or category itself and all relations or categories within two upward traversals. The choice of k value is sensitive to the depth and specificity of the ontology.

Retrieval and Localization

After mapping the semantic information extracted from an article into BioPAX+, R3 localizes it by retrieving all matching entities and processes in the model. R3 uses a two-

stage similarity-based retrieval algorithm: Given a probe (i.e., the process or entity description) and a library (i.e., a set of entity and process descriptions from the model), the first stage is an efficient feature vector dot-product between the probe and each context to filter low-similarity descriptions, and the second stage is the graph-matching recognition algorithm described earlier. The result is a similarity-ranked list of a subset of the model library. R3 uses some additional graph-matching constraints to ensure that the explicitly described entities and relations are in the mapping (e.g., for "MEK-directed phosphorylation of ERK," the MEK, ERK, and phosphorylation event are all required); otherwise, the mapping operation terminates with a score of zero. R3 thereby identifies and ranks portions of the domain model according to their structural similarity to the extracted information. As we show in the evaluation section below, this localization approach recognizes entities and processes with high precision and recall; however, it does not account for context and causal locality.

Localizing References to Signaling Pathways

Another important extension of the localization process was the development of a means of localizing references to entire pathways. When a pathway is mentioned, it often establishes the context in which subsequent references are to be interpreted. Articles frequently refer to pathways in different ways, as exemplified by these three examples.

1. "the Ras/Raf/Mek/Erk signaling pathway,"
2. "the Raf/Mek/Erk pathway,"
3. or even more succinctly, "the RAF/MAP kinase cascade."

These all refer to the same pathway—which is a reified entity in Reactome that is associated with a specific set of reactions—but the phrases contain between two and four proteins, and are not identical. Rather than doing a pure name-match, we implemented an initial approach to *bottom-up localization*, whereby R3 localizes a *subset* of the phrase, and then uses those results to constrain higher-level localization of the pathways themselves. For instance given this text.

[the [Ras]^{1}/[Raf]^{1}/[Mek]^{1}/[Erk]^{1} signaling pathway]^{2}

R3 localizes the red components first, associates those model locales with the protein mentions, and then attempts to localize the entire (blue) pathway, given those localized components.

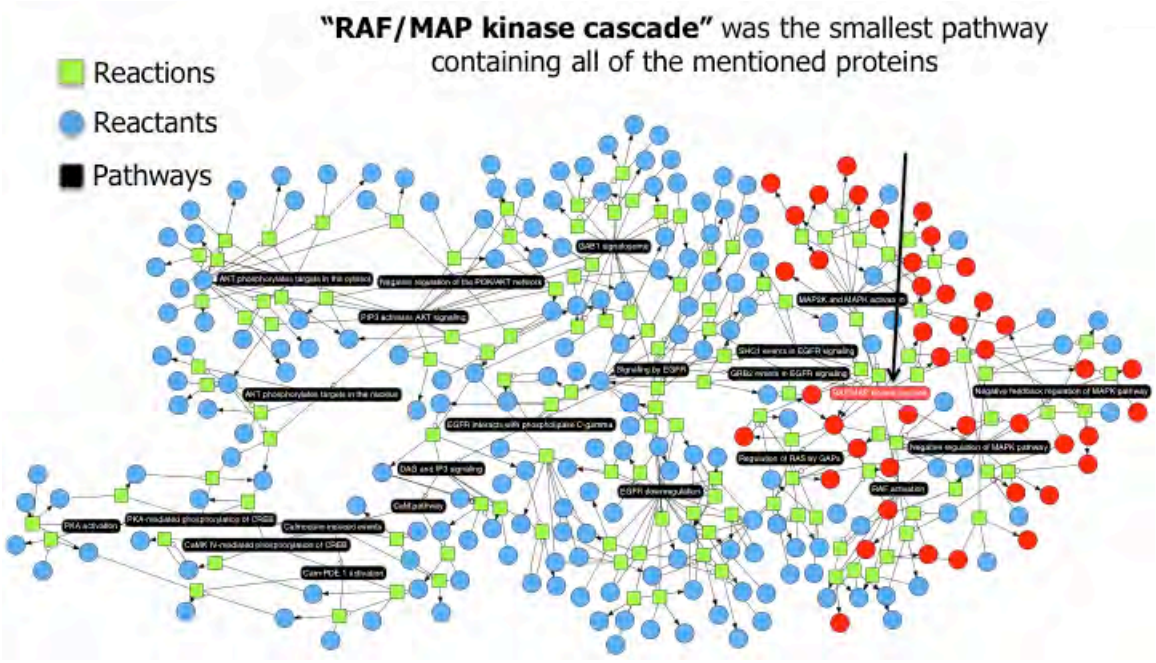


Figure 11: Localizing RAF/MapK pathway (mentioned entities in red)

R3 localizes the example pathway phrases 1-3 correctly using this bottom-up strategy, by finding pathways that contain all of the listed components in the model, and then ranking by size, smallest first. Intuitively, the largest superset pathway (e.g., the entire EGFR Signaling pathway) is relevant, but the subset of that pathway that contains all of these components (e.g., the actual RAS/RAF/MEK/ERK cascade) is a better answer.

Localizing Causal Chains

Causal assertions occur frequently in the literature. Causal assertions range from distant indirect relations that broadly establish the context of the paper, to the more specific relationships between reactions a short distance downstream, to the *very* specific relationships within a single reaction. Consider the following snippet from a Reactome comment about a reaction in the model written by a biologist:

> *"…Nucleotide exchange stimulates a conformational change in RAS to facilitate its interaction with RAF,* ***ultimately promoting the phosphorylation of downstream effectors MAPK3 and MAPK1 (also known as ERK1 and ERK2)*** *(reviewed in Cseh et al, 2014; Vigil et al, 2010)…"*

Figure 12 shows a graph of a portion of Reactome model for RAS that includes the reaction for this comment. Where other reactions are shown as green squares, this one is highlighted in yellow (near left with arrow). The bold-faced text refers to reactions (e.g., phosphorylations of ERK1/2) further downstream in the pathway, denoted by red squares. R3 automatically identifies these red reactions using its existing localization machinery; it identifies four phosphorylations of ERK1/2.

Localizing a causal pathway does not end here; presumably, the authors mention downstream reactions for a reason, e.g., to establish context and indicate other useful background knowledge that should be salient to the reader. For this purpose, R3 uses its

reaction graph to perform a connectivity search, identifying relevant reactions and reactants (ORANGE squares and circles, respectively) that it traverses in the path from the source to the downstream events.

Note that two of the four ERK phosphorylations (RED squares) are not reachable from the source (YELLOW) node in this graph, since these two are RAS-independent phosphorylations of ERK1/2, and were presumably not the reactions referenced by the authors. This demonstrates how a causal graph search can help constrain localization and marshal other relevant causal knowledge (e.g., RAF activation, MEK activation) into memory for coreference and assembly.

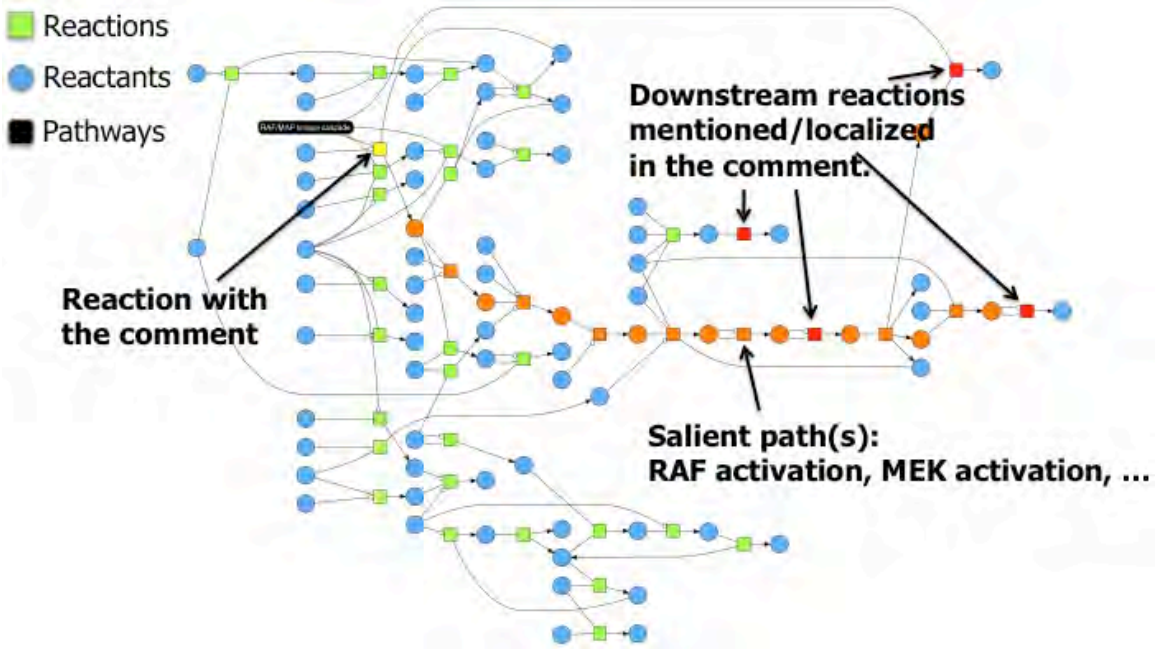


Figure 12: Localized pathway in orange

A User Interface for Viewing Articles and Localized Entities

During year two, we prototyped a user interface that would allow people to see what expressions were considered by R3 for localization and what model elements were considered reference candidates by these expressions. Some additional work on this interface was done during year three so that it could be used to collect expert annotation data on the process.

The Localization UI shows the phrases that R3 has related to our Reactome-based BioPAX+ model by localization reasoning. As illustrated in the figure below, it shows the article being read (using either SPARSER or TRIPS/DRUM), the phrases (in light purple) whose parsed semantic form corresponded to a localizable molecule, complex, or reaction in the model, as well as a means of marking (in green) phrases that the user wishes to annotate manually. Graphical views of the semantic interpretation of each phrase, and of the reactions that were found to match can be shown. The user can directly access Reactome views of the items that were successfully localized. Candidate localizations are shown in the right-hand panel, and you can either get a graph of R3's own internal

representation of those reactions (BioPAX+) or jump directly to the Reactome Website browser view.

We believe that this kind of interface could have great value, as it allows experts to review links to the model and to correct the errors that R3 may have made in that process. It can also be used as a form of Biocuration interface more generally. However, we ended up using other methods to collect our "gold standard" corpus, due to the need to work remotely with our expert, as discussed in the next section.

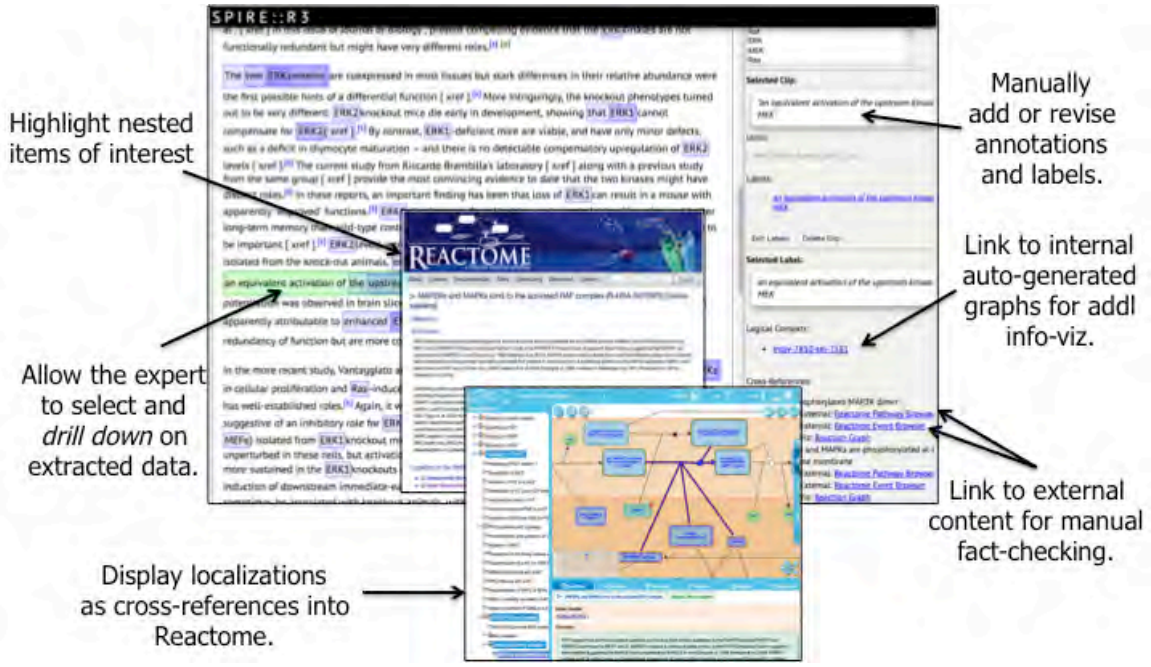


Figure 13: Overview of document viewer with localization functionality

4. Localization Evaluation Studies

Enhanced Evaluation of the R3 Localization System

We view localization as a traditional Information Retrieval (IR) problem, so we are evaluating it as such, using precision, recall, and/or f-measure. During the final year of the program, we endeavored to produce a gold standard for comparison with R3 by engaging a biologist to mark up a small set of articles. However, that effort was only partially successful as we detail later. The goal was to establish, given (e.g.) that an article mentions "active MAPK," which *exact set* of molecules in Reactome are being referred to, in the expert's opinion. As collecting that gold standard has been difficult and costly, our initial experiments during years two and three only tested the added value of localization against retrieval of possible referents based on the mere presence of the named proteins in an event or complex in the BioPAX model. We first review those studies before describing our final evaluation effort.

Prior to the Big Mechanism PI meeting in March 2016, we completed a first test of our parse-to-model localization for a small set of articles. The test consisted of taking statements mentioning processes or reactions (such as might appear in MITRE cards), and attempting to identify the reaction or protein complex in the BioPAX model that that event or protein was referring to. For this test we looked at 3 articles, containing 21 processes and 77 molecules (including duplicates), of which 15 were described in terms of their functional state of activation. Across this set of articles, we had R3 try to identify the events or proteins, as described semantically by SPARSER from the articles. Consider a phrase like "*enhanced ERK2 activation was observed*." Using only the original BioPAX model, we compared how well it could just identify mentions of ERK2 in the model. For the EGFR subset of Reactome that we were working with, this would find all of the items in red in the following figure (14a):

However, with the knowledge about when ERK2 was activated in the target model, R3 could narrow it down to the following mentions in red (yellow indicates the parts of the model that were enhanced by reading the comments on Reactome objects and events:

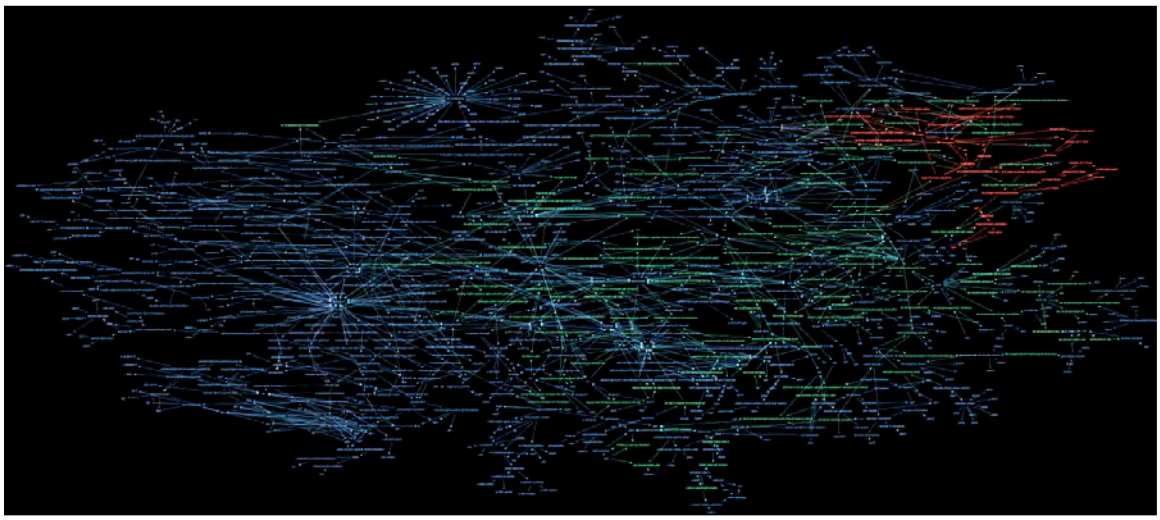


Figure 14a: Mentions of ERK2 in EGFR model (in red)

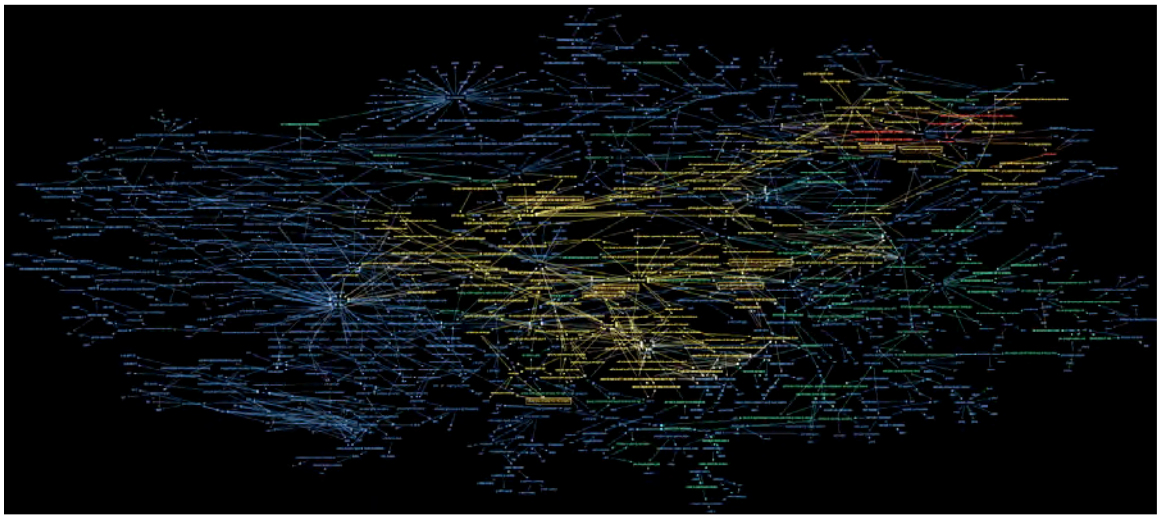


Figure 14b: Mentions of Active ERK2 (red) after labeling functional forms (yellow)

Zooming in on this figure we can more easily see what is mentioned. This is the set of complexes involving activated (phosphorylated) MAPKs, of which ERK2 is a member.

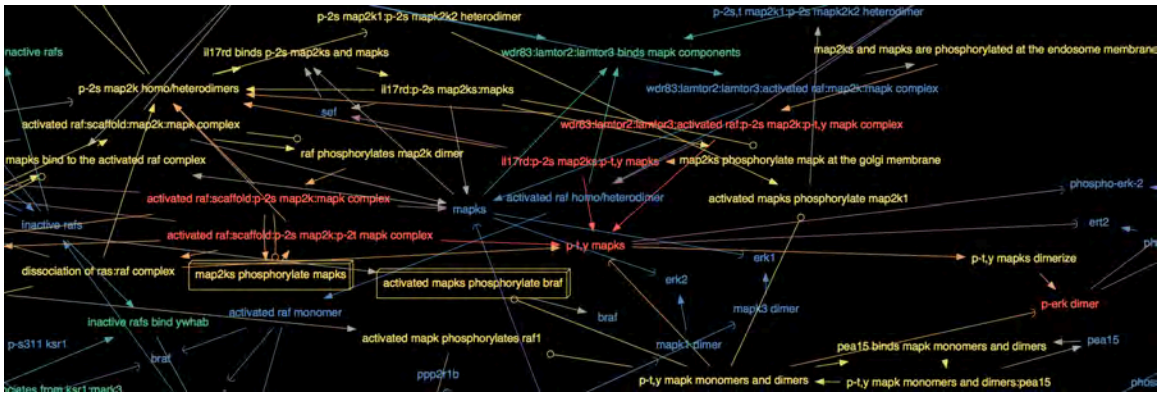


Figure 14c: Subset of Figure 14b showing identified activated Mapk (ERK2) forms

Overall, this process improved the F1 measure to 0.96 on the set of mentions that were localized, vs. only 0.46 when BioPAX was not enhanced by reading the comments on the model. This preliminary experiment demonstrated that R3 can learn functional knowledge by reading, and that this knowledge improves R3's subsequent ability to localize extracted knowledge as it reads.

Evaluating R3 Localization using DRUM

A more substantial experiment was conducted during year three (August/September, 2016) after we had completed the integration of the TRIPS/DRUM article reader with R3. For this experiment, IHMC provided us with DRUM parses of 11 articles that are referenced in the "Signaling by EGFR" subset of Reactome, so we were sure that those articles would contain references to elements of the "Signaling by EGFR" BioPAX+ model.

We integrated R3 with IHMC's DRUM parser to demonstrate the generality of our localization machinery and exemplify how other reading systems can integrate with R3. For any reading system, integrating with R3 involves writing an ontology-mapping from the reader's ontology into R3's extended BioPAX notation. Since DRUM's output uses a

normalized TRIPS ontology output —which is similar to SPARSER's output— writing this ontology-mapping was fairly straightforward.

With the EGFR signaling subset of Reactome that R3 had processed forming the model (i.e., what R3 would localize *into*), we read 11 articles cross-referenced by that portion of the model (i.e., *what* R3 would localize from. This portion of Reactome includes the well-known RAS-RAF-MEK-ERK cascade, EGFR stimulation, and ERK translocation into the nucleus. We chose articles cross-referenced by the model because, presumably, they would have high topical overlap with the model and therefore provide ample grist to test the localization process.

Below are the statistics for these 11 articles. Across all the articles, R3 mapped DRUM's output to our extended BioPAX notation, and then it identified 3,404 relevant *mentions* (i.e., phrases referring to molecules or interactions that R3 should try to localize), including proteins, complexes, modified proteins (e.g., "phosphorylated ERK"), active forms (e.g., "active ERK"), modification events (e.g., "MEK-directed phosphorylation of ERK"), binding events (e.g., "MEK association with ERK"), and activation events (e.g., "ERK activation"). The 3,404 mentions contain duplicates, since articles frequently mention the same protein many times.

Articles	11
Total Relevant Mentions:	3404
Averages per article	
Relevant mentions per article	309.5
Mentions ascribed to model	144.6
Model retrievals	3749.2
Mentions not found in model	164.8

As shown in the lower half of the above table, for each article, R3 averaged 310 localization attempts, finding just under half of them (145) in the Reactome EGFR Signaling model. To make these 145 successful localizations per article, R3 averaged 3,750 model retrievals, which means retrieving an average of 26 elements from the model for any mention it successfully localized. This is due to the high number of molecules described in Reactome. For instance, ERK is described as ERK1 and ERK2, and both of these are described in multiple modification states, in monomer, heterodimer, and homodimer forms, and all this is duplicated across cellular locations, for a massive Cartesian product of molecule descriptions. For this reason, R3 ranks its list of retrievals by numerical similarity, using the graph-matching score as its similarity metric. This means that if just "ERK" is mentioned, proteins (ERK1, ERK2) are frequently reported first, then modified variants of proteins, then dimers, and then larger compounds, e.g., RAF+MEK+ERK.

The second table, below, shows a breakout comparison of three different localization techniques:

1. **Full Localization using Graph-matching** (at left): this is our current setup, whereby R3 localizes by producing a semantic (extended BioPAX) graph for each mention, matching it against the model to identify candidates for retrieval, and then filtering and ranking by similarity.

2. **Localization using Protein name-matching only** (center): This is a simpler, overly eager approach, whereby R3 uses only protein names to retrieve complexes and reactions. This means "active MAPK" or "phosphorylated MAPK" would map to simply "MAPK," and "MEK phosphorylates MAPK" would retrieve any reaction that contains proteins matching both MEK and MAPK.
3. **Localization using Graph-matching without activation information** (right): this uses the same graph-matching approach as the first (left) condition, but without any information about active forms or activation events, which R3 learned by reading comments. For all mentions of molecules and reactions that do *not* involve active forms or activation (e.g., "MEK phosphorylates ERK"), the behavior is identical to the first (left) method. But for phrases like 'MEK activates ERK', it is equivalent to method 2.

	Full Localization using Graph-matching	Localization using Protein name-matching only		Localization using Graph-matching without activation information	
	# found in model	# found	% False Positives	# found	% False Positives
Protein mentions ascribed to model	1,477	1,482	0.34%	1,482	0.34%
Model retrievals per protein mention	40,863	41,870	2.41%	41,870	2.41%
Reaction mentions ascribed to model	114	133	14.29%	119	4.20%
Model retrievals per reaction mention	378	1,970	80.81%	1,625	76.74%

The solutions produced by the leftmost "Full Localization" condition are by far the highest quality, and we have vetted some of the results with experts. That said, we can easily identify the *additional* false positives incurred by the simpler competing approaches (center and right). We break this into protein mentions/retrievals and reaction mentions/retrievals:

- For proteins mentions, the competing less careful approaches both lead to a few additional ascriptions of references to the model, so that 0.3% of the proteins that it localizes in the model actually are not in the model (e.g., 'active SOS' finds all SOS references, even if we don't know which are the active ones). This also results in more erroneous retrievals from the model, so that 2.4% of all protein references retrieved from the model are incorrect. For instance, "active ERK" should retrieve a subset of ERKs, but these competing approaches instead retrieve all ERKs.
- The comparison is much more pronounced for events (i.e., reactions). Protein-matching erroneously ascribes mentioned reactions to the model 14% of the time, and when it retrieves reactions from the model, 81% of the retrieved entities are incorrect. This demonstrates the value of event structure in localization. Importantly, the graph-matching without activation knowledge (at right) *also* has a high error rate (77%), which demonstrates the value of activation knowledge, given the pervasiveness of "active" and "activation" language in the text.

We provided the information about active protein forms to HMS/CURE for use in their assembly process. The way in which they use it is different, but the effect is much the same in that they are identifying and trying to unify and add to their model different descriptions

of the same reactions, some of which are described functionally and others in terms of their biochemistry.

Developing a "gold standard" for evaluating localization

In order to further develop a meaningful evaluation, we set out to develop a sufficient set of articles annotated by an expert with the following:

1. Markings of all phrases that could potentially be identified in a Reactome model.
2. Labeling those phrases with the type of entity or relation referenced.
3. Identifying the possible referents in the model that the phrase could be indicating.

We decided that the most expedient method to do these tasks was to use an existing tool, MITRE's Callisto annotation tool, to accomplish the first two parts of the job. We hired Dr. Maciej Kotecki, a biologist from Prof. Cochran's lab as a consultant to do this work. Dr. Kotecki has worked with Profs. Cochran and Pustejovsky in the past doing a similar kind of annotation using Callisto, which meant he required little training for this step. From the prior project, Callisto had been configured with an appropriate set of labels for our task. This allowed us to more rapidly collect data for the first two of the three types of required information. Below is an example of using the Callisto tool in this fashion.

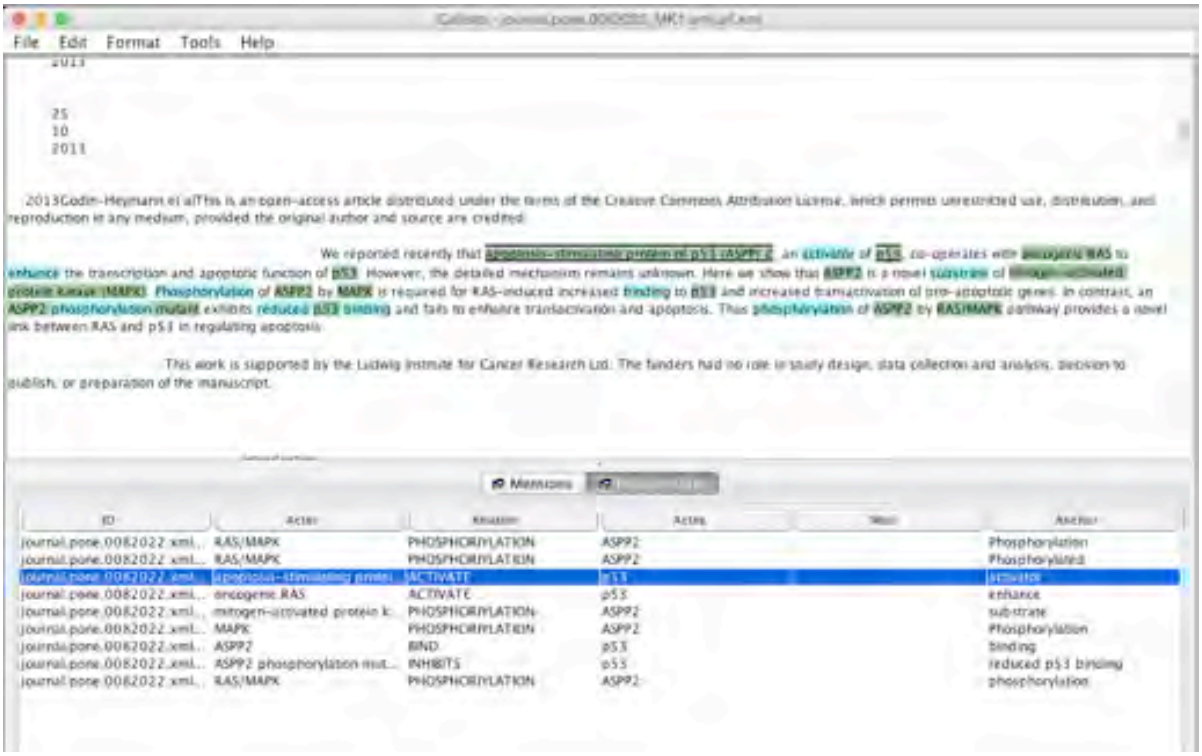


Figure 15: Annotating using MITRE's Callisto

Since the set of available labels in Callisto from Dr. Kotecki's previous project included such things as phosphorylations and activations, we were also able to use the data collected from his annotation of five articles to gather appropriate *potential* referents in the model that could be used for the third and final stage of the process.

To prepare for this phase of the process, we developed software to extract the annotated information from Callisto's native XML file format and convert it to a form that could be compared to the set of text regions marked as containing localizable phrases by our R3 system. The goal was to evaluate how accurately R3 identifies the entities and events that can be potentially referenced in the Reactome model, and to select a broad set of potential model references for our expert to select among. For each phrase that was marked in Callisto, we matched the entities involved against the Reactome model and found all entities and/or events involving those entities, depending on whether Dr. Kotecki had indicated the expression represented an entity or relation.

For the next phase of the process, localization, we constructed a new web interface that used the data from the Callisto annotation process as its starting point. Figure 16 (a and b) illustrate the process we asked Dr. Kotecki to follow. Figure 16a shows the initial process of selecting a phrase, previously marked in Callisto, for localization to the Reactome model. For each phrase and mention of that phrase in context, we asked him to choose whatever reasonable references to the model he could, given the context in which the phrase occurred. Step 1 is to select the article being worked on. Step 2 is to select one of the entities that was annotated in Callisto (each entity may occur multiple times in an article). Step 3 is to choose one of the mentions of the phrase (a specific location in the article) for localization. Callisto uses the notion of a "primary mention" for repeated entities, so that the annotator can group phrases referring to the same entity. We made use of these groupings so that the localization only has to occur once, though it can be changed for each additional mention if the intended reference has indeed changed.

Figure 16b shows our web interface for collecting annotator choices on localizations into Reactome for the specific mention he selected to focus on. In the top left corner, the specific mention is identified by the article, annotation (the set of identical entities) and the specific mention. Below that, the annotator can enter comments about his decision process or questions for the group. Entity mentions can either be about a specific entity/protein, or a family of proteins like RAF, or name a pathway. Checkboxes helped us to understand which he thought it was, as R3 can really only localize specific proteins or complexes in this context, as that is what Reactome emphasizes.

The table at the bottom of the screen is a list of all of the candidate references (nodes in the model) selected by R3 by a loose match against the model. As the goal was to give the annotator choices without overwhelming him, but not to eliminate any possibly relevant choices, we used a broader criteria for selecting potential references than used by R3, to ensure that the annotator could potentially include things we might have missed. This included searching for possible synonyms among the candidates. We discovered during the final analysis that at least one common synonym had been missed, and removed those choices from R3 as well, to make the results comparable.

As there could be many potential references for each phrase or mention, we provided some search and sort tools to help make the annotator's selection of references to the model easier. For entities, the user could enter a search string, which would find and bring all of those to the top of the displayed list. A button allowed all of those highlighted as matching to be selected simultaneously, and then individually unchecked if not appropriate. A second search could then find additional matches. A similar but more complex search was used with relations, as described below.

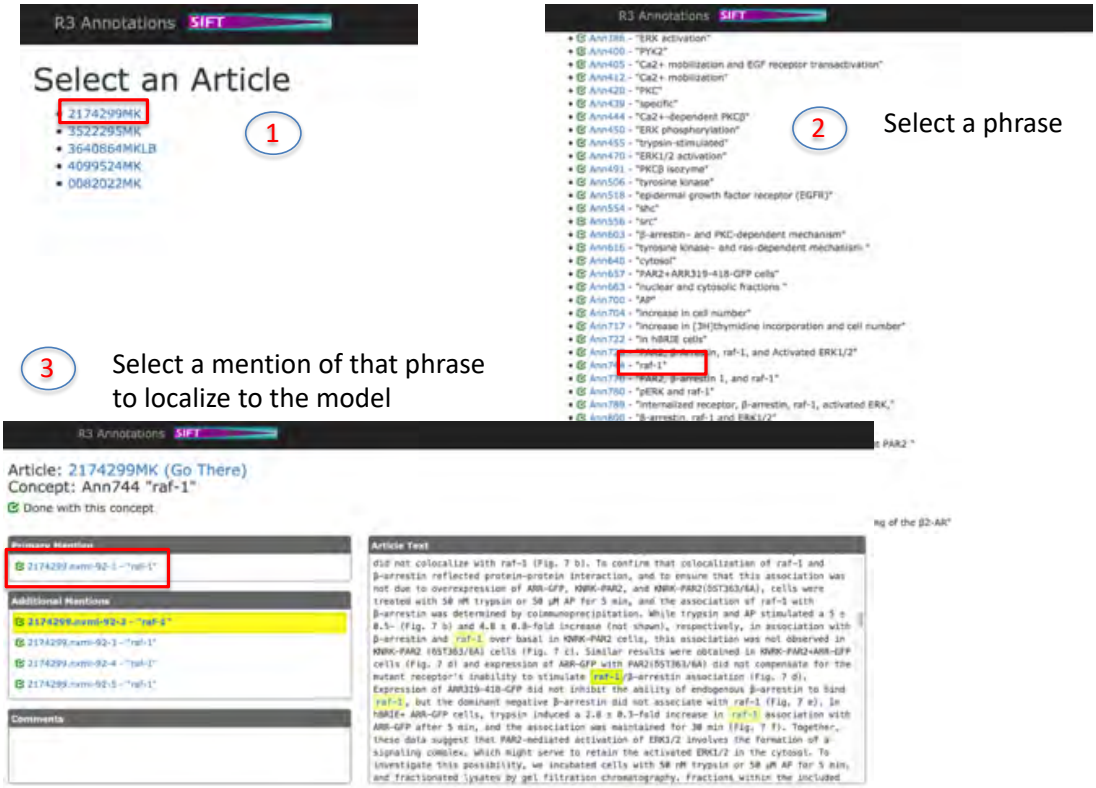


Figure 16a: Process for selecting a specific phrase to localize to the model

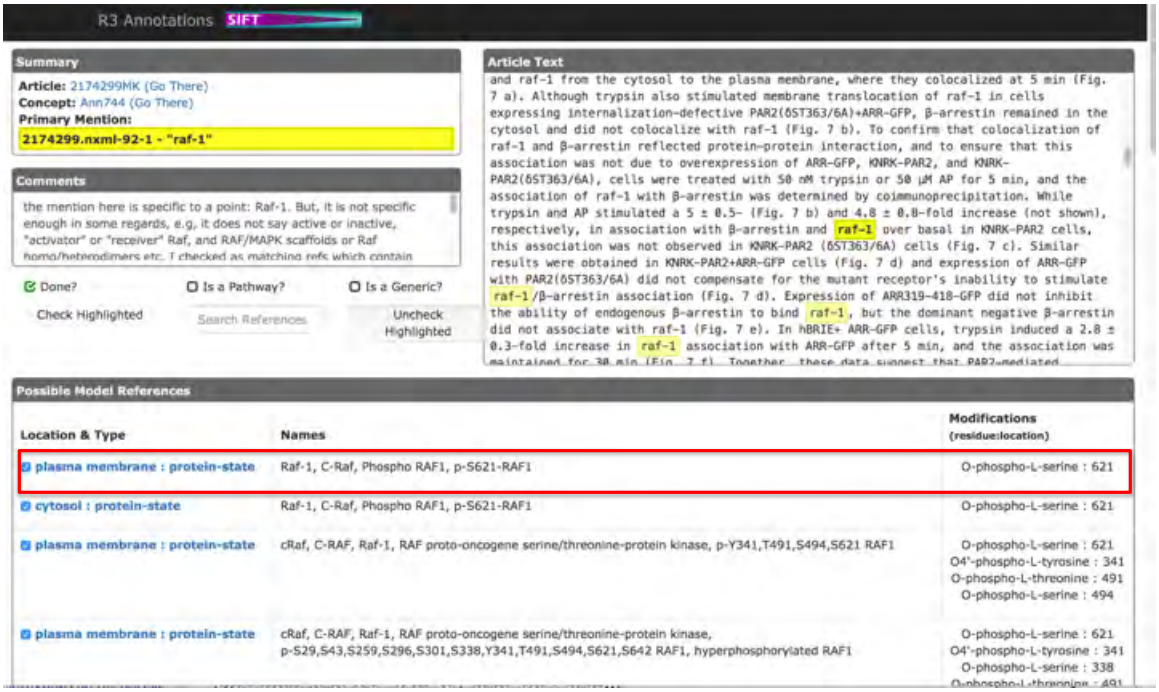


Figure 16b: Marking model elements corresponding to chosen phrases/mentions

Figure 16c shows the localization selection screen for relations. Similar to Figure 16b in many respects, but with more searching capability. Here, the relation being recorded is a regulation of BRAP by RAS. The annotator enters RAS and BRAP as search terms and finds three reactions in the model. Two of them are most appropriate, but all three can be checked by "Check Highlighted" button and then one unchecked if ultimately found to be inappropriate. The comment by Dr. Kotecki (MK) shows this process playing out.

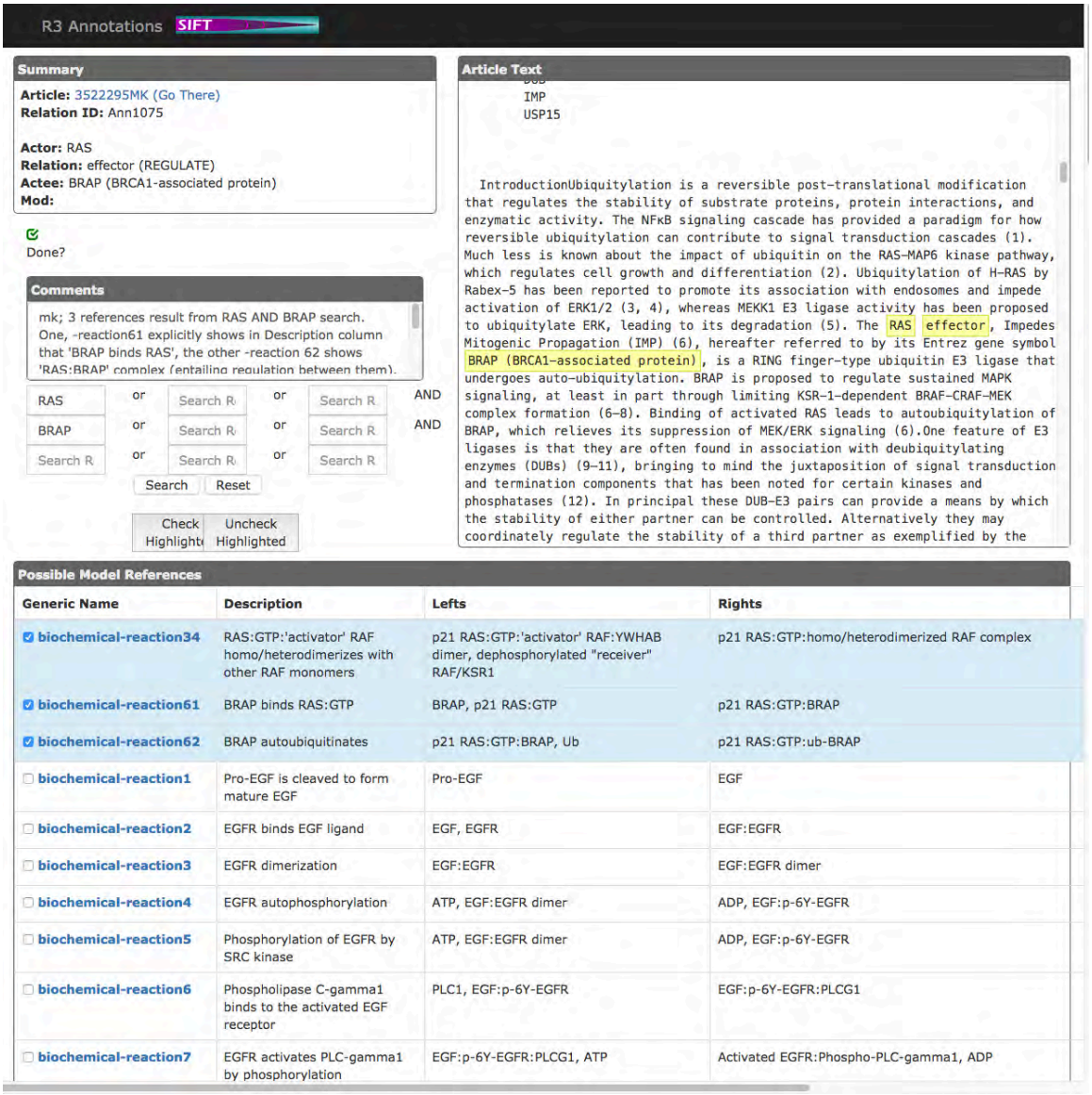


Figure 16c: Annotator Localization of Relations

The data is saved automatically on the server as the annotator works, so that work can continue at any future time and then can also be used in our data analysis which developed recognition and recall statistics on:

- Whether the text segments chosen by R3 and the expert agree or differ.
- Whether R3 and the expert agree on how the text segments highlighted are localized into the model.

Discussion of Results

Below we describe the results of the study. We would like to preface our remarks with a few caveats: This was a time-consuming and expensive study and, as no similar study has ever been done before, we learned a few hard lessons along the way. We discovered too late that we needed to give more precise instructions to the annotator so that certain kinds of systematic errors were not made. This would have improved the accuracy of our outcomes. First, we clearly did not give precise enough guidance on the initial part of the study, where Dr. Kotecki (MK) marked up articles using Callisto. We relied on his prior experience with the tool and did not notice until too late in the process that he made some decisions that were inconsistently applied and that ended up hampering our data analysis, and which could have been corrected with better instructions and by having a second annotator for corroboration if there was time and resources to do so.

The first problem was that he did not mark all entities mentioned in the text, but only or primarily those that were involved in relations. This became evident in the disparity in entities found by R3 as compared to our expert (a factor of 4 to 1, as shown in Figure 17).

The second problem was that some of the things that he marked as entities were not entities but nominalized pathways or events, as those were often the named causal antecedents or agents of the events being described. For example, for the sentence:

> *"In contrast, in KNRK-PAR2(δST363/6A) cells, AP induced a 2.4 ± 0.8-fold increase in [3H]thymidine incorporation and a 2.7 ± 0.5-fold increase in cell number, which is similar to that observed with serum."*

MK identified two relations, one with "[3H]thymidine incorporation" and one with "increase in cell number" as consequents of the process that occurred in an experimental condition in KNRK-PAR2(δST363/6A) cells. Neither of these consequences are biological entities. They are both processes. The first can be interpreted as producing a complex containing [3H]thymidine, but the second is not a molecular result, but rather an observable macro-level change.

Dr. Kotecki also marked conjunctions of entities as though they were a single entity, where R3 did the opposite. For example, in the sentence:

> *"Some examples are insulin receptor substrate-1, where phosphorylation by ERK1/2 leads to cellular insulin resistance (DeFea and Roth 1997), cytoskeletal proteins, and microtubule-associated proteins (MAPs; Sturgill and Ray 1986; Ray and Sturgill 1988), where phosphorylation is involved in cell migration, morphological alterations (Goedert et al. 1996; Klemke et al. 1997), and phospholipase A2 (Bornfeldt et al. 1997), which mediates many agonist-induced inflammatory responses."*

he marked the phrase “cytoskeletal proteins, and microtubule-associated proteins” as a single entity.

Conjunctions of processes are also beyond the current capabilities of R3, for example:

> *PAR2 couples Gαq/11 and phospholipase Cβ, leading to hydrolysis of*

phosphatidylinositol bisphosphate, Ca2+ mobilization, and activation of protein kinase C (PKC) and ERK1/2 (Déry et al. 1998)."

Here we have a conjunction of two proteins which should to be interpreted as bound by PAR2 (as opposed to acted on separately by PAR2), and three consequences of the PAR2 coupling. Dr. Kotecki formed a conjunction for the second and third consequences. For the purposes of our analysis, both conjunctions were ignored in the adjusted entity recall numbers.

We addressed the first issue as best we could in our analysis by only considering the entities R3 identified only if they substantially matched ones also marked by Dr. Kotecki. R3 was able to identify 78% of the entities annotated by Dr. Kotecki (% of MK), and 87% when the conjunctions were removed from his entity set, although as mentioned R3 had found many more entities than Dr. Kotecki overall.

The second problem mostly affected the analysis for relations. With regard to this issue, non-entities marked as entities due to their role in events, we considered partial matches between events/relations (with their subordinate roles for type, agent, antecedent and consequent). Thus, for example, if the event type and consequent were correct for a phrase we accepted that as indicating a corresponding event description. In the tables of Figures 17 and 18, **Span "Recall"** refers to the percent of text spans (contiguous words in the text) identified by Dr. Kotecki using Callisto that were also identified using R3, based on our heuristics for comparing matching spans. As mentioned previously, the average per article **Total** span recall for entities was .78. We made an adjustment to remove from consideration entities that R3 could not match without making changes to how R3 broke up spans covering conjunctions, which increased the average span recall to .87 (**Adjusted**).

Both R3 and MK localized approximately half of the entity spans each marked (% found), and only about one-third of the relations (28% for R3, 35% for MK). The others they were unable to find correspondences for. This was partly because the target model was only a portion of Reactome, and partly because only certain kinds of events or relations were relatable to the model.

Results of ENTITY Localization Analysis

	Entities Found		Localized		Spans Matched	Span "Recall"	
Article ID	**R3**	**MK**	**R3**	**MK**	**Total**	**Total**	**Adjusted**
2174299MK	1144	298	486	141	232	0.78	0.83
3522295MK	1085	218	428	107	170	0.78	0.96
3640864MK	990	245	670	218	216	0.88	0.92
4099524MK	607	190	433	128	122	0.64	0.75
3847091MK	649	183	223	55	143	0.78	0.88
Total	**4475**	**1134**	**2240**	**649**	**883**		
Average/Article	**895**	**227**	**448**	**130**	**177**	**0.78**	**0.87**
			% of found		% of MK		
			50%	57%	78%		

	# Matched & Localized		Localization	
Article ID	**R3**	**MK**	**Recall**	**Precision**
2174299MK	122.00	121.00	0.98	0.92
3522295MK	100.00	87.00	0.96	0.75
3640864MK	205.00	204.00	0.94	0.83
4099524MK	100.00	99.00	0.91	0.84
3847091MK	51.00	50.00	0.91	0.68
Average/Article	**115.60**	**112.20**	**0.94**	**0.80**

Figure 17: Summary tables of results of entity localization study

The columns we call **Localization Recall** and **Precision** refer to the calculations comparing R3's localization to the model choices with those of our annotator, Dr. Kotecki (MK), for spans that both identified and localized. We do not consider the annotation quality overall to be good enough to judge this as a "gold standard" or ground truth, but for the purpose of describing those quantities, we use standard recall and precision calculations. As discussed above, we made denominator adjustments here to eliminate some localization references from the reference precision and recall when the marked text spans referred to a protein that was not mentioned in the portion of Reactome we were working with, was therefore not correctly localizable, and so was considered an annotation error.

For entity localization with respect to matching spans, (still Figure 17), the results (with the adjustments described above) indicated a high average **Localization Recall** (.94) and moderately high average **Localization Precision** (.80). Our results with and without knowledge of activation were similar as there were relatively few proteins mentioned as active in these articles relative to the overall number of entities.

Results of RELATION Localization Analysis

	Relations Found		Matched		Span "Recall"		
Article ID	**R3**	**MK**	**Total**	**Exact**	**Total**	**Adjusted**	**Exact**
2174299MK	188	160	85	36	0.53	0.61	0.23
3522295MK	128	118	27	11	0.23	0.23	0.09
3640864MK	221	128	52	28	0.41	0.44	0.22
4099524MK	105	115	33	12	0.29	0.41	0.10
3847091MK	94	99	33	15	0.33	0.38	0.15
Total	**736**	**620**	**230**	**102**			
Average/Article	**147**	**124**	**46**	**20**	**0.37**	**0.42**	**0.16**
			% of MK				
			37%	16%			

	# Matched & Localized			Reference Recall		Reference Precision	
Article ID	**R3 w. Active**	**R3 w/o**	**MK**	**w. Actives**	**w/o. Actives**	**w. Actives**	**w/o. Actives**
2174299MK	27	30	14	0.16	0.37	0.05	0.05
3522295MK	9	9	7	0.30	0.31	0.28	0.07
3640864MK	12	12	49	0.20	0.22	0.48	0.46
4099524MK	15	15	26	0.33	0.39	0.68	0.44
3847091MK	0	2	0	nan	nan	nan	0.00
Average	**12.60**	**13.60**	**19.20**	**0.25**	**0.32**	**0.37**	**0.20**
Std. Dev.				**0.07**	**0.06**	**0.24**	**0.20**

Figure 18: Summary tables of results of relation localization study

With respect to our analysis of relation localization (Figure 18), we had some difficulties matching to MK's annotations, and of course there are far fewer of them. As he sometimes used non-entities (processes or conjunctions) as though they were individual entities playing roles in his relations/events, of the approximately 736 (147 per article) marked by R3 and 620 marked by MK, only 230 matched at least partially, and only 102 matched exactly. Our **Span Recall** (regions marked by R3 as a fraction of those marked by MK) was only .37, or .42 after some invalid annotations by MK were removed from the denominator.

For Relation Localization against the model, we found (on average per article) only 12.6 of the 147 spans marking relations identified by R3 had both spans matching MK also had marked and identified references to the model. If we turned off use of activation information for the R3 localization process, that number increased to 13.6 of the 147 (because the localization matching was thus less restrictive, increasing recall). By contrast, of the span-matching relations MK found, 19.2 had localizations on average.

For calculating precision and recall, we looked at how R3 performed with and without activation knowledge in the model localization process. As suggested, this knowledge both increased precision and decreased recall, as more specific information in the localization matching process leads to fewer matching entities in the model. For **Reference Recall**, the average per article was .25 with activation knowledge and .32 without. For **Reference Precision** we had a score of .37 using activation knowledge and .20 without.

Our results were somewhat impacted by the fact that both R3 and MK had no localizations for the last article, so we basically could not score that article on recall and precision. The reason was that the article was about ASPP2 and the Reactome model fragment we were using unfortunately did not include that protein. If we had more time,

we would definitely redo the scoring of that article (and several others that were less severely impacted) with a larger reference model.

Discussion and future directions

Both improvements to the "gold standard" and improvements to R3 are possible that would greatly improve the baseline numbers above. However, even with these incomplete results we see that the trends suggested by the earlier experiments hold. Namely, the use of information about the functional states of proteins such as the active/inactive state, together with knowledge about which complexes in the model contained those proteins in different states has a measurable impact on our ability to localize correctly (increased Precision).

It was also clear from the way that Dr. Kotecki annotated relations that many of the relations were indirect relations or influences, which were not localizable to a single point in the Reactome model, as they would require search through reaction chains to determine if those influences were present in the model. While we did some work on that issue during the program, it was not ready to be incorporated into the localization process used by R3 for this study. This is an important direction for future work.

The different ways conjunctions were treated by R3 and Dr. Kotecki was another big issue in the study, and is another area for future work. Some of this should be addressed by better annotation methods, but further work is also needed to use context to determine whether conjunctions refer to, for example, co-agents of a single process or alternative agents capable of performing the same process. This is one area where localization into the model could be seen as integral to improving the interpretation process (reading with the model).

5. Activities in support of the CURE consortium system

SIFT provided several forms of support to CURE, as shown in Figure 17 below. First, the information that R3 discovers about the relationship between active states of protein complexes and their structural forms is critical to CURE during its assembly process as a means of relating statements from different readers that talk either about functional states of proteins or the complexes that they are in, but not both. Second, R3 SPARSER became the second reader to read virtually all articles processed by CURE, and provided substantial contrasting information to the REACH reader.

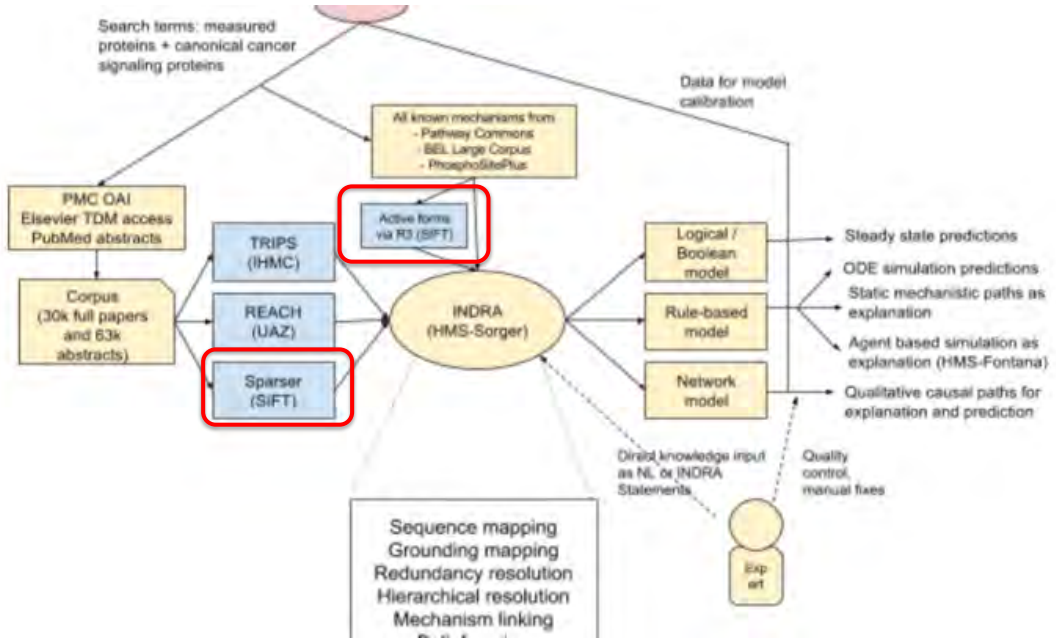


Figure 17: Roles of R3 products in the CURE assembly pipeline (highlighted in red).

Extracting and encoding additional active forms for CURE

One of our contributions to CURE is supplying the set of associations between active forms of proteins and their molecular structures, and the set of activating/deactivating events identified in the Reactome data. We collected these associations when analyzing the English language descriptions of Reactome's reactions for their functional information about active forms. CURE uses the data (i.e., structural configurations of molecules that afford their function as catalysts or key reactants in pathways) to improve their assembly and explanation efforts, since they were now able to identify and cluster descriptions for which they previously could not align the protein references. As we have already noted, knowledge of which forms are considered active (which is how they are often mentioned in texts) is not encoded in the native Reactome XML data, making model assembly from text much more difficult. Yet they are some of the most common kinds of references found in the literature that we have looked at.

HMS presented a slide at the April 2017 PI meeting that showed just how prevalent these expressions are among the statements that they extracted from readers. We reproduce it here as Figure 18, as it shows on a log scale that statements of activation and inhibition together are many times more prevalent than complex and phosphorylation, the next most frequent statement types.

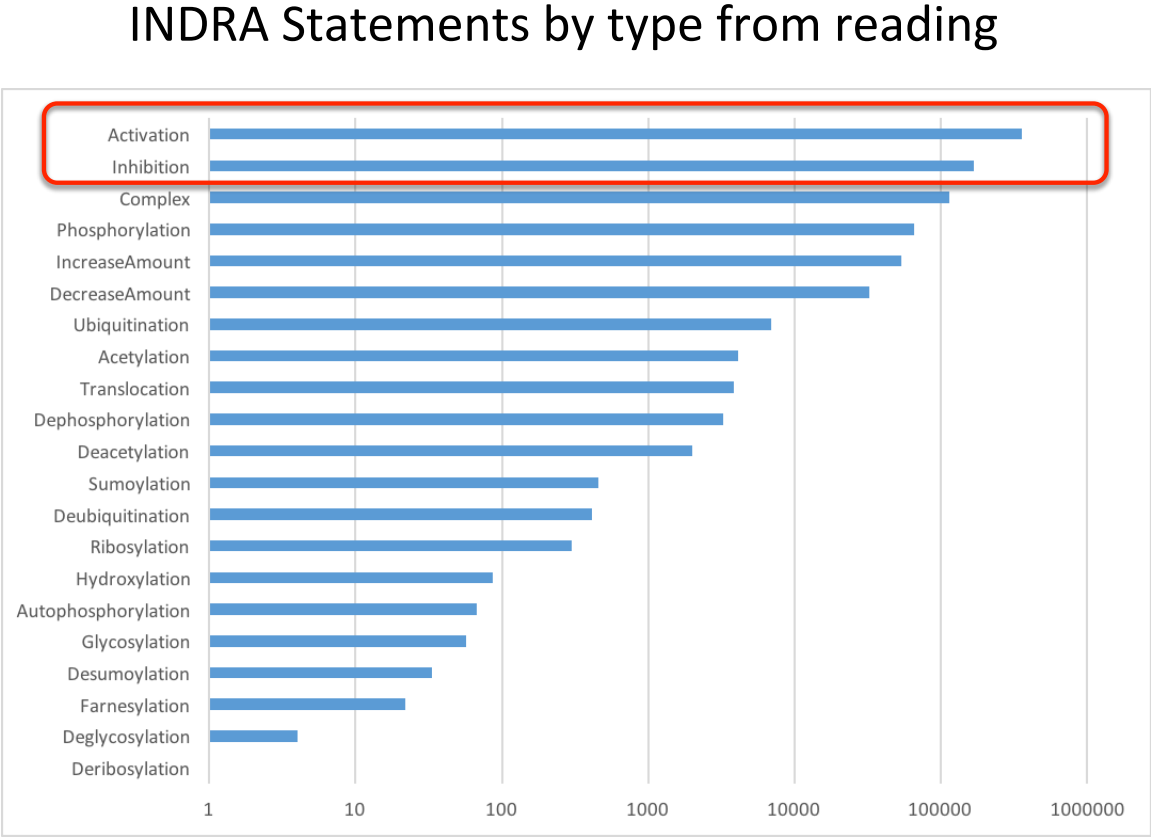


Figure 18: Induced Indra statements by type.

As noted, R3 extracts its active form information by (1) parsing textual comments that describe activity and activation, (2) extending its representation of the referenced complexes to explicitly encode active states, (3) generalizing these active forms by combining them with others to identify minimal active forms, (4) propagating these generalized active forms across cellular locations and reactions, and then (5) converting active forms into INDRA statements to send to CURE.

In October 2016, we sent HMS our first batch of active forms inferred by the first technique for the "EGFR Signaling" subset of Reactome. However, the completeness of the comment-parsing approach was circumscribed by the completeness of the textual comments in Reactome, and we have found many genes of interest to CURE whose active forms are *not* identified in Reactome comments. Consequently, we began to work with the HMS CURE team to identify other sources (textual or otherwise) and strategies for inferring active forms, and to extend our current method to encompass all of the human cell signaling knowledge in Reactome.

Scaling up R3 to Homo Sapiens Reactome

We scaled up our production of active form knowledge by improving SPIRE's ability to handle larger Reactome models. Our "EGFR Signaling" portion of Reactome covered most of the important information about the Ras cascade, but other articles went beyond that, so it was worthwhile to process the entire "Homo Sapiens" portion of Reactome. Compared to the EGFR Signaling portion of Reactome, in this larger portion there are roughly 75

times more reactions (total of 9,643 reactions), and 45 times more molecules (total of 10,416 complexes and 31,248 proteins).

Scaling up presented challenges to our SPIRE reasoner because we use a densely indexed in-memory data store for rapid graph-matching. We developed two new approaches to optimize memory storage and inference machinery and a number of internal elements of the SPIRE reasoner. To more quickly support the export of specific information about active forms for CURE, we established a pipeline to work through Reactome's data models in a piecemeal fashion.

We made several kinds of improvements to the underlying SPIRE inference engine to enable it to scale up to handle larger models, such as moving from simple atomic simple horn clauses to more frame-like terms, (which is also more consistent with Reactome, DRUM and SPARSER). This most notable enhancement allowed SPIRE to use a native frame-based representation (in addition to the native fact-based triple representation) to store all properties and property-values of an entity in a single in-memory object, which is especially convenient for Reactome itself. This also better supports native compatibility with the Logical Form notation used in TRIPS and SPARSER.

Since our SPIRE inference engine relies heavily on different logical contexts—and inheritance relations over them—to compartmentalize its knowledge, SPIRE dynamically unifies frames when an object's properties are defined across multiple inheriting contexts. To correctly answer queries in these situations, we have SPIRE construct these coherent frames across contexts dynamically, reflecting the inferential differences between entity variants in their respective contexts. Handling frames directly as assertions in a context potentially corresponds to a great many triples or predicate terms, which would have made scaling difficult.

The result of all of these improvements to the underlying system produced a 50% increase in performance for query times and in the initialization process when loading a BioPAX model. Additional internal performance tuning changes sped this up further.

SPIRE extensions enabled functional extraction from Human Reactome

By enhancing our SPIRE inference engine to support efficient scale-up from our "EGFR Signaling" portion of Reactome to the entire "Homo Sapiens" portion we were able to load the approximately 10,000 complexes and 10,000 reactions, of the full model. HMS was interested in a specific list of 289 gene names associated with those complexes and reactions. Of the 289, 29 weren't mentioned in the Reactome model. Another 46 were not controllers. For the remainder, R3 inferred 2,125 active forms, given the multiplicative effects of their use as controls in multiple contexts, and when bound in different complexes.

The corresponding active forms (i.e., structural or mutation or modification or binding conditions for gene activity) were extracted by the process detailed earlier for those with identified labels that mentioned "*active,*" and then we used R3 to read the labels and ascribe active states to those structural descriptions in the model.

R3 successfully parsed 80 protein labels that contained the substring "*activ*," and a few others, e.g., "*Activated STAT1/3 homo and heterodimers*" (one of the simpler cases), labeled the entities as active, and then propagated this across all cellular locations and to all subsets of active-labeled sets of molecules. Since many of the labeled proteins were sets

(complexes), this produced multiple active forms for many labels. Ultimately, R3 produced 209 active forms of proteins.

One such active form, from HMS's genes of interest, is shown below in Indra JSON format. The JSON contains information about protein modification sites (STAT3 is active when Tyrosine-phosphorylated at site 705), binding requirements (STAT3 is active when bound to Tyrosine-phosphorylated STAT1), and evidence/provenance.

Unfortunately, the overlap of the 289 genes sought by HMS with the 209 active forms R3 found by reading was quite low (only 20 were within HMS's 289). Since this gap comes from the limitations of what the Reactome curators happened to explicitly label, linguistic coverage (like searching and reading "*activated*" labels) can only get us this far, and we began researching additional strategies for inferring active forms. This work is discussed in the next section.

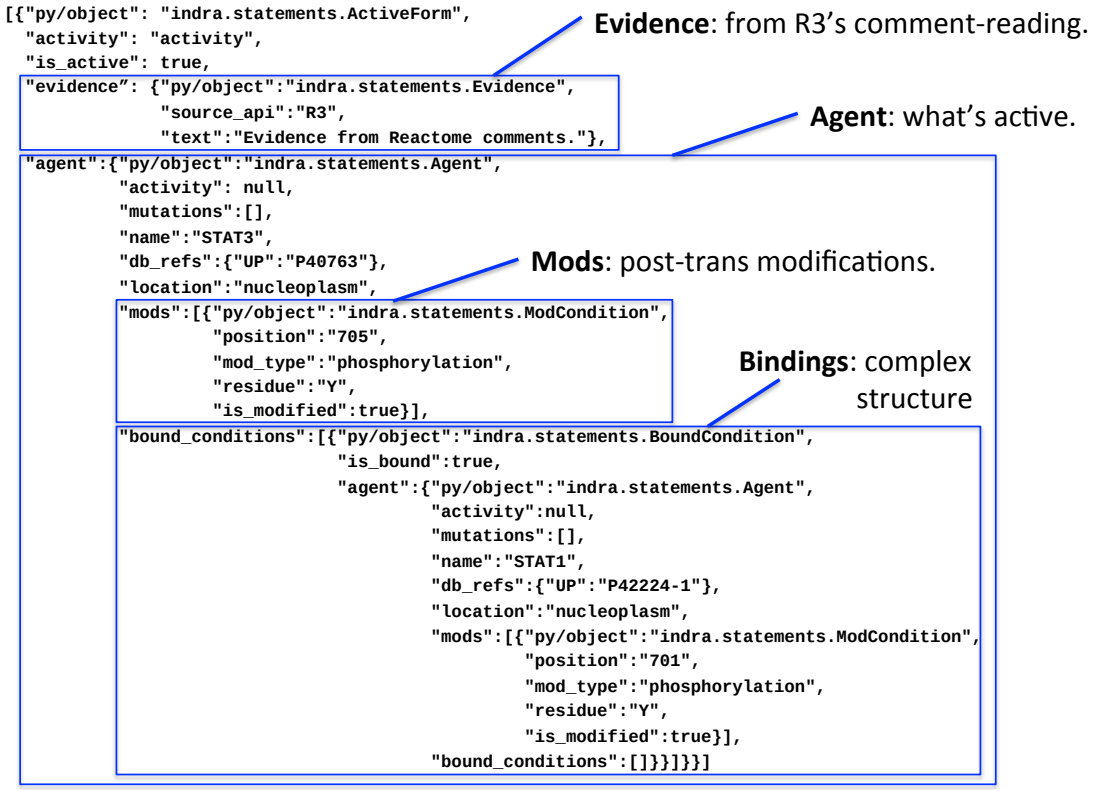


Figure 18: JSON format of complex active form, as supplied to CURE

Gathering active forms by structural reasoning

As discussed, HMS sent us a list of 289 relevant gene names for the materials that they were assembling in CURE. The active forms that we were able to discover by reading comments on the full human Reactome netted 80 proteins that were called active in some cases, and we were able to provide to CURE the exact complexes and state information associated with those forms and related forms (a total of 209). Unfortunately, these forms did not overlap strongly with the genes of interest to CURE. Though, as should be clear from the method we employed, R3 would only find such protein states if they were called that in the associated text labels and descriptions. In many cases this misses proteins that also have active states but were not called out.

To counter this problem, we developed a new method for identifying active forms of proteins and kinases by looking at the situations in which forms of proteins (primarily catalysts) are used as controllers in reactions in the model, and then identifying the specific protein states that appear in those complexes. We then tested using a heuristic definition of "active" to do a structural search of the Reactome BioPAX model, both to identify which variants of the protein regulate reactions that involve other proteins, and to identify the reactions that produce those variants.

During our final year, we developed these capabilities for inferring active forms from non-textual data in the BioPAX Homo Sapiens Reactome model. The non-textual data includes (1) *structural data* about molecule sets, molecule subcomponents, cellular location, and post-translational modification and (2) *behavioral data* about catalysis, positive and negative regulation, and reactions.

Our approach is not guaranteed to be sound and complete in its inference of active forms, since "*active"* is a contextual term that refers to *some* configuration of an entity that underlies its activity (which is also contextual). That said, we aimed to develop multiple heuristics that together produced high precision and recall for inferring pathway activities and the active forms of the entities that support them.

We based our work on the following central assumptions:

- **Activity is circumscribed by positive and negative regulation behavior**. This means that a protein configuration can only be active if it is a direct regulator of a reaction in the model, or it is a subcomponent of a direct regulator.
- **Activity affects *other* things**. Activity is generally thought of as a cascade of entities affecting other things. This means that the reaction "*Protein A regulates Protein B's phosphorylation*" is a better indicator of Protein A's activity than "*Protein A regulates its own phosphorylation.*"
- **Unmodified kinases aren't active if another, modified form of the kinase is potentially active**. This is a heuristic from our biologist collaborators. We integrated HMS's kinase list as a lookup resource for R3 to identify kinases in the model by name.

These assumptions informed R3's heuristics for inferring active forms from structure and behavior. The two figures below were generated by R3; we labeled them afterwards for clarity. In the left figure, R3 starts at right (with the circled MAP2K1 protein reference) and then traverses the model to locate forms of that protein (e.g., "MEK1" and "p-T286,292-MEK1" and others) and onward to protein super-components (e.g., MEK1:ERK1), and ultimately to controlled reactions (e.g., "MAP2K phosphorylates MAPK3"). R3 then uses two heuristics to constrain its search for active MEK: (1) unmodified kinases (e.g., "MEK1") are not active, so the unmodified protein form MEK1 is not considered (and colored red); and (2) regulating a self-modifying reaction, such as "RAF phosphorylates MAP2K dimer," is not sufficient to infer an active form, so that reaction is colored red to exclude it from consideration.

In the right figure, R3 traverses rightward from the remaining reactions and protein forms, identifying flow dominators (e.g., "RAF activating kinase" and "p-2T-MAP2K1"), junctions in the flow graph (e.g., "p-S218,222 MEK1"), and BioPAX molecule sets (e.g., "p-2S MAP2K homo/heterodimers"). These are boxed to indicate minimal active forms of the protein.

Using the new method, R3 found 228 of the 289 proteins mentioned in the full human Reactome, and inferred 453 distinct active forms (some with multiple configurations / locations). These results were provided to the CURE team and incorporated into their assembly process.

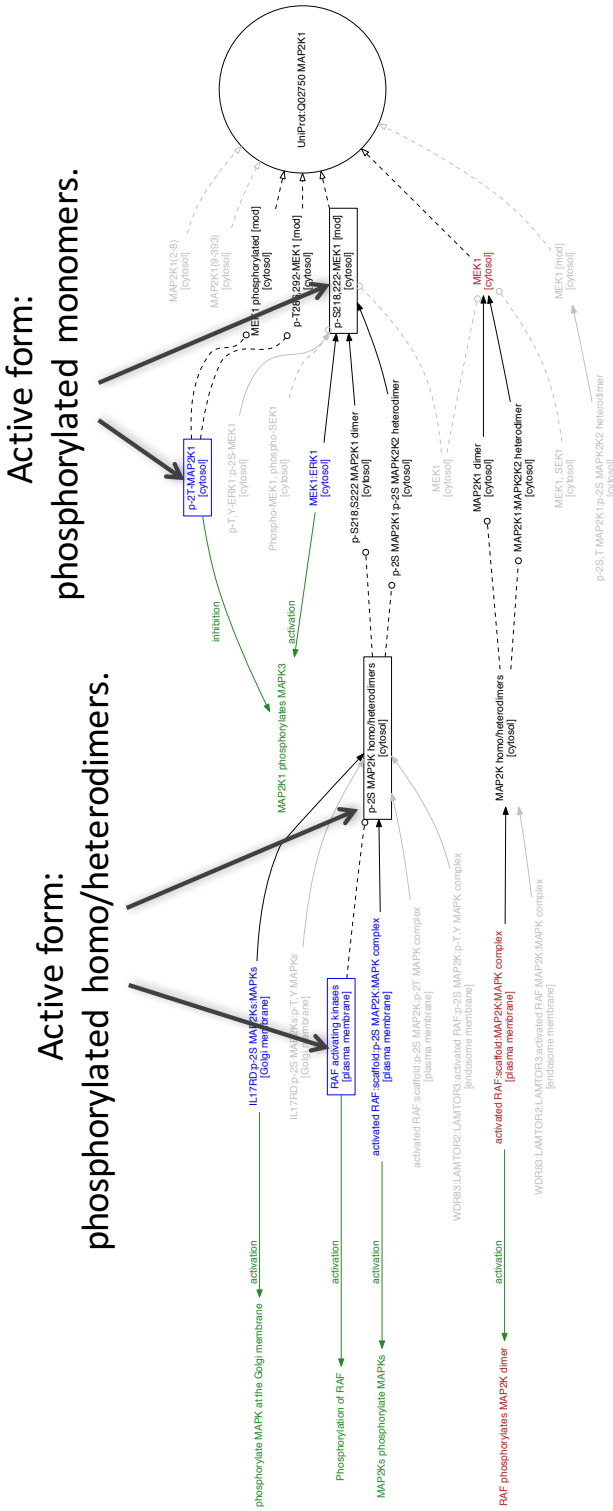


Figure 19: Graph-based heuristics to identify active forms

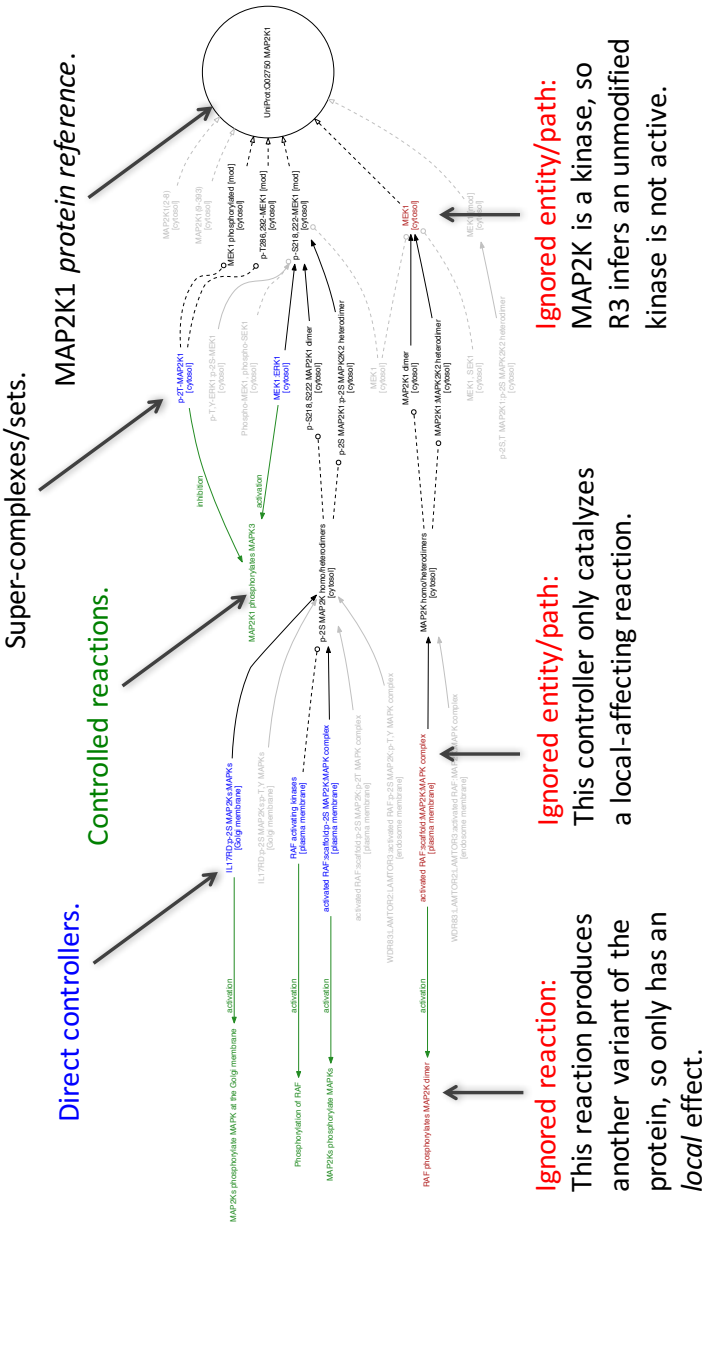


Figure 20: Heuristics for ignoring certain reactions

6. Publications

"Extending Biology Models with Deep NLP over Scientific Articles" by David McDonald, Scott Friedman, Amandalynne Paullada, Rusty Bobrow and Mark Burstein was presented at the 2016 AAAI Workshop on Knowledge Extraction from Text.

"Reconciling Function and Structure in Scientific Models" by Scott Friedman, Mark Burstein, David McDonald, Rusty Bobrow, Brent Cochran, James Pustejovsky and Peter Anick was presented at the IJCAI 2016 workshop on Qualitative Reasoning in July 2016.

"Learning by Reading: Extending & Localizing Against a Model" by Scott Friedman, Mark Burstein, David McDonald, Amandalynne Paullada, Alex Plotnick, Rusty Bobrow, Brent Cochran, James Pustejovsky and Peter Anick. Originally presented at the 2016 Cognitive Systems Conference (April, 2016), extended and published in the Cognitive Systems Journal (December 2017) as Advanced in Cognitive Systems 5:77-96.

References

Bobrow, R. and Webber, B. (1980) Knowledge Representation for Syntactic/Semantic Processing. In Proceedings of AAAI-80, The First National Conference on Artificial Intelligence, Stanford, CA, August 1980 (pp. 316-323)

Burstein, M. H. (1988). Combining analogies in mental models. In Analogical Reasoning, (pp. 179–203). Springer.

Danos, V., Feret, J., Fontana, W., Harmer, R., & Krivine, J. (2009). Rule-based modelling and model perturbation. In Transactions on Computational Systems Biology XI, (pp. 116–137). Springer.

de Kleer, J., and Brown, J. S. (1981) Mental models of physical mechanisms and their acquisition. Cognitive skills and their acquisition 285–309.

Demir, E., et al. (2010). The biopax community standard for pathway data sharing. Nature biotechnology, 28, 935–942.

Falkenhainer, B., and Forbus, K. D. (1991) Compositional modeling: finding the right model for the job. Artificial intelligence, 51(1):95–143.

Friedman, S. E., Barbella, D. M., & Forbus, K. D. (2012). Revising domain knowledge with cross-domain analogy. Advances in Cognitive Systems, 2, 13–24.

McDonald, D. D. (1996). The interplay of syntactic and semantic node labels in partial parsing. In H. Bunt & M. Tomita (Eds.), Recent Advances in Parsing Technology, (p. 295-323). Kluwer Academic Publishers.

McDonald, D. D. (2000). Issues in the representation of real texts: The design of Krisp. In L. M. Iwanska & S. C. Shapiro (Eds.), Natural Language Processing and Knowledge Representation, (pp. 77–110). MIT Press.

Pustejovsky, J. (1991a) The generative lexicon. Computational linguistics 17(4):409–441.

Pustejovsky, J. (1991b) The syntax of event structure. Cognition, 1(41):47–81.

Pustejovsky, J. (2013) Dynamic event structure and habitat theory. Proceedings of GL2013 1–20.

Rickel, J., and Porter, B. (1997) Automated modeling of complex systems to answer prediction questions. Artificial Intelligence 93(1):201–260.

Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldbert, L.J., Elibeck, K, Ireland, A., Mungall, C.J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S. (2007) The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology. 25(11):1251-1255. PMCID 17989687.