REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
02-04-2019	Final Technical Report	3 July 2014 - 2 April 2019

4. TITLE AND SUBTITLE

Pathway Logic Extended with Information Assembled from Data Extracted Selectively (PLEIADES)

5a. CONTRACT NUMBER

W911NF-14-C-0108

5b. GRANT NUMBER

Not Applicable

5c. PROGRAM ELEMENT NUMBER

Unknown

6. AUTHOR(S)

Dayne Freitag, Richard Rohwer, Mark Craven, Carolyn Talcott, Andrew Silberfarb

5d. PROJECT NUMBER

P22648

5e. TASK NUMBER

Not Applicable

5f. WORK UNIT NUMBER

Not Applicable

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SRI International
9988 Hibert Street, Suite 203
San Diego CA 92131

8. PERFORMING ORGANIZATION REPORT NUMBER

None

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

Army Research Office
ARO-RTP
4300 S. Miami Blvd.
Durham NC 27703

10. SPONSOR/MONITOR'S ACRONYM(S)

ARO-RTP

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

None

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Report developed under DARPA BAA 14-14 "Big Mechanism", contract W911NF-14-C-0108. This report details our work under DARPA's Big Mechanism program, which pursued automated methods for the creation and extension of sophisticated mechanistic models through the ingestion of textual information. Under Big Mechanism, this problem1 was factored into three topic areas: reading, or the automated conversion of human-language text to structured semantic form; assembly, or the incorporation of reading outputs into working mechanistic models; and explanation, or the use of these enhanced models to solve problems or provide insights to humans.

As detailed in this report, SRI's PLEIADES effort addressed all three topic areas. We developed novel methods for recognizing and representing experiments described in the biological literature, including approaches to elicitation of feedback from human experts to compensate for deficiencies in available training data. The "experiment frames" produced by such reading were used as a source

15. SUBJECT TERMS

Big Mechanism Report, Machine reading, information extraction, computational biology, knowledge representation, mechanistic models

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UL		

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

RPPR

as of 03-Apr-2019

Agency Code:

Proposal Number: **Agreement Number:**

Organization:
Address: , ,
Country:
DUNS Number: EIN:
Report Date: Date Received:
for Period Beginning and Ending
Title:
Begin Performance Period: **End Performance Period:**
Report Term: -
Submitted By: Email:
Phone:

Distribution Statement: -

STEM Degrees: **STEM Participants:**

Major Goals:

Accomplishments:

Training Opportunities:

Results Dissemination:

Plans Next Period:

Honors and Awards:

Protocol Activity Status:

Technology Transfer:

SRI International

2 April 2019

PLEIADES: Pathway Logic Extended with Information Assembled from Data Extracted Selectively

Final Project Report

SRI Project P22648 & P25195
Contract No.: W911NF-14-C-0108
Proposal No: ECU 14-611(R2)
Contract Performance Period (extended): 07-03-2014 – 04-02-2019

Contractor's Name:

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Subcontractors:

University of Wisconsin-Madison

1. PROJECT OBJECTIVES

This report details our work under DARPA's Big Mechanism program, which pursued automated methods for the creation and extension of sophisticated mechanistic models through the ingestion of textual information. Under Big Mechanism, this problem was factored into three topic areas: *reading*, or the automated conversion of human-language text to structured semantic form; *assembly*, or the incorporation of reading outputs into working mechanistic models; and *explanation*, or the use of these enhanced models to solve problems or provide insights to humans. The program nominated a challenging problem domain, the cellular signaling pathways associated with a critical protein family called Ras, the malfunction of which is implicated in a number of types of cancer. This problem domain has a number of characteristics that made it useful to the program: the relevant mechanisms are poorly understood, the pathways in question are large, textual material treating these pathways are voluminous, and any successes in meeting the program's challenge would potentially yield immediate medical benefit.

Together with its partner, the University of Wisconsin, SRI addressed all three topic areas, seeking to assemble an end-to-end system that extracts pertinent information from the biological literature, updates a high-fidelity executable model of the relevant pathways, and uses this model to explain clinical outcomes, such as the observed effect of drugs on cellular system (e.g., an increase in the production of a particular protein). Our work was predicated on *Pathway Logic* (PL), an approach to modeling cellular pathways under development at SRI for the past two decades. Among other things, the PL project had already produced large, high-fidelity mechanistic models of the pathways of interest to Big Mechanism. Because human knowledge of these pathways is incomplete and human capacity for model curation finite, the relevant PL models are necessarily imperfect, affording an almost ideal starting point for a program pursuing automated model improvement.

PL implements a rewriting logic with temporal semantics. The elements of a PL model are "rules," each rule pairing one or more preconditions with one or more postconditions. Informally, a rule is a statement saying that if a particular configuration of factors holds at some time, and effect is observed that produces a new configuration of factors. In models of cellular signaling, the factors involved in a rule typically express information about particular proteins in the cellular environment, their location, their state of modification, their conformation with other nearby proteins, etc. The "firing" of a rule may effect a change of state that satisfies the preconditions of another rule, which may fire in its turn, etc. Through these rule execution cascades, PL simulates the series of reactions that constitute cellular signals. PL models are ideal for answering the "what if" questions that underlie the design of modern cancer treatment drugs, enabling the designer to experiment with interventions considerably upstream of the change in cell state most directly tied to the development of cancer.

In addition to curating the rules that implement models, the PL project has systematically formalized the biological experiments that provide evidence for these rules. Each record of these experiments, called datums[1], records detailed information about a specific experiment described

[1] We use the non-standard plural in referring to these records to distinguish them from generic data.

somewhere in the biological literature. The database of these records easily constitutes the largest data resource of the PL project, containing some 70,000 records by the end of our Big Mechanism effort. This database provided critical to our project on several fronts, enabling us to substantiate a key insight: *that automated model enhancement requires attention to hard evidence if the resulting systems are expected to be predictive of real-world phenomena*. The datum database essentially provides evidential guardrails that benefit the model optimization process, which otherwise might suffer significant degradation through the incorporation of erroneous or underspecified information.

This key insight became an organizing principle for our effort. Our research on *reading* focused primarily on the automated acquisition of evidence through the automated extraction of simplified datums. The algorithms we proposed for *assembly* employed objective functions based on level of agreement with the datum database. And *explanation* in the PLEIADES project almost always meant accounting for a specific evidential outcome and the identification of the narrow pathway neighborhoods most responsible for a particular observation.

As we drew near the end of the project, we were ready to show that these principles are more general than biology, that they should apply in other domains having features similar to the Big Mechanism challenge problem—abundant technical literature, a source of structured evidence, and need for explanations in the form of series of discrete events. We therefore proposed to demonstrate the application of these techniques to a problem domain of more immediate interest to DARPA's sponsors, accounting for geopolitical interactions in "gray zones." We were granted a modest amount of funding to pursue this research. This report also summarizes the lessons learned in this exercise in domain transfer.

2. TECHNICAL WORK PERFORMED

In what follows, we have organized our reporting into four topics: *Reading for Evidence*, *Amplifying Human Effort*, *High-fidelity Biological Models*, and *Automated Model Extension*. This organization combines closely related tasks from our statement of work (which were slightly more numerous) for the sake of clarity.

This topical organization of material is orthogonal to the report structure required by our contact, which calls for a presentation of the technical work, followed by a summary of the results obtained. We therefore adopt a parallel topical structure in the section on results that follows this section.

2.1. READING FOR EVIDENCE

Our work on reading in this project had a specific end use: We sought to construct a machine representation of the experiments described in a given paper on cell biology, hypothesizing that the experimental evidence this affords will provide important constraints on the assembly of biologically realistic mechanistic models. This focus posed challenges not confronted by other machine reading performers in the program. Whereas the objective of most relation extraction is the faithful capture of certain kinds of statements from individual sentences, we observe that an experiment's key details must often be assembled from multiple sentences. And whereas the conventional reader passes the responsibility of synthesizing the information it extracts to

downstream processes, we have no such luxury. The experimental details are not of interest in isolation, and can only confirm or refute mechanistic assertions when considered in combination.

Experiment Extraction

In pursuit of "experiment extraction" we had the advantage of a large amount of inadvertent data annotation. The Pathway Logic project, which provided our team with the modeling infrastructure upon which our efforts are centered, had created thousands of machine-readable "datums," each describing some experiment culled from the literature, and each retaining a pointer to a paper figure. A core hypothesis we sought to validate is that this alignment between datums and the literature provides a form of distant supervision we can exploit to realize our experiment extraction goals.

A first critical step in validating this hypothesis was to simplify the reading problem to a form that is both tractable and biologically informative. Biological assays involve many details of interest to the human scientist, and the structure of a typical datum reflects this abundance of detail. However, our primary interest within the Big Mechanism program was to provide evidence for or against mechanistic assertions. To this end, we narrowed our focus to four key fields that most datums provide:

- **Assay**. A categorical field that records the type of experiment conducted. For example, the type "phos" indicates a phosphorylation assay, one seeking to determine whether one biological entity causes or is instrumental in the phosphorylation of another.
- **Change**. A categorical field that ranges over a small number of possible experimental outcomes, such as "increased" and "detected". There are perhaps half a dozen such values, which partly depend on the type of assay.
- **Subject**. The biological entity (typically a protein) whose change is being measured. Note that, in contrast with *assay* and *change*, *subject* belongs to a class that is open in practice, as we cannot assume that we have seen all possible assay subjects.
- **Treatment**. The biological entity that is introduced into the cellular environment in order to assess its effect on the subject. Like *subject*, *treatment* is open-class.

As this list implies, our reading challenge in its initial formulation was to accept a paper and output a set of quadruples, each capturing the key aspects of some experiment described in the paper. The objective of this task (Task 2) is to assemble the raw ingredients out of which such quadruples can be constructed, and to note pairwise field correspondences that might inform such construction.

Corroborative Reading

In an initial attack on this challenge, we authored an extensive set of extraction rules for the four properties listed above, using University of Arizona's REACH extraction engine as the rule interpreter. The effort to produce these rules was part of our effort to implement the *Evidence Expert*, a proof-of-concept software component for validating mechanistic assertions with extracted experiments that was part of the FRIES consortium's offering at the first program evaluation. FRIES was a consortium of several performers, including SRI, CMU, University of Arizona, ISI, Leidos, and Elsevier, formed to combine complementary technical strengths in pursuit of a complete offering.

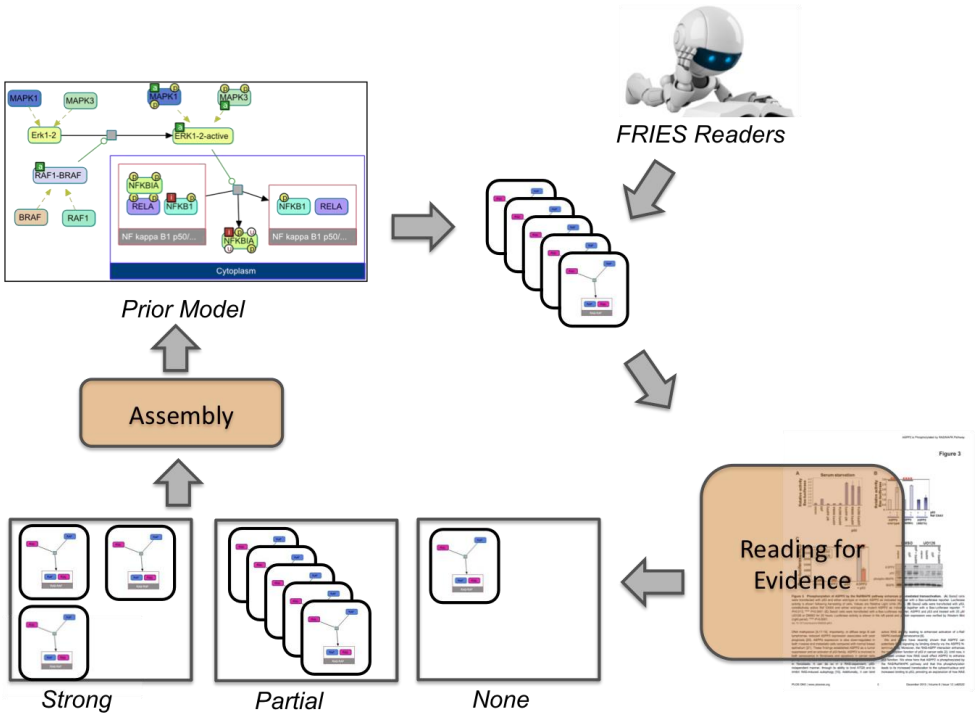

Figure 1: Evidence Expert concept of operations.

The role of our reading system in the overall FRIES architecture is illustrated in Figure 1, specifically the box labeled "Reading for Evidence." As the figure shows, the *Evidence Expert* accepts biological "facts" (typically low-cardinality relations or events) extracted from sentence-level expressions by other consortium groups focusing on reading. It then compares these facts (called "index cards" in the parlance of the Phase 1 evaluation) with experiments extracted from the same paper, assigning each card on of four labels reflecting the card's level of substantiation:

- **Not supported**. Support for the type of relation asserted by the index card has not been added to the Evidence Expert yet.
- **Fully substantiated**. A datum was constructed that represents the right kind of assay to substantiate the asserted interaction.
- **Partially substantiated**. A datum was constructed that agrees in certain key features with the asserted interaction (i.e., shares some entities or represents the right kind of experimental assay), but not all.
- **Not substantiated**. As far as the Evidence Expert can tell, no appropriate assay was conducted in the paper to substantiate the asserted interaction.

As the figure indicates, this level of corroboration is then used to inform decisions about which new facts warrant inclusion in a pathway model.

Finding Experiment Elements

As mentioned, the Evidence Expert relies on hand-authored rules in the syntax used by UA's REACH system. It is very difficult when creating such rules to assess their accuracy or coverage. We compensated for this difficulty by creating a dashboard that showed the number of matches for a given rule, measured its accuracy against known datum elements (e.g., the

"subject" field of one or more datums that point to a paper), and provided a sample of matching sentences. Even with such a facility, creating extraction rules like this and applying them to a large corpus is a hit-or-miss exercise. One never knows whether a given rule is good enough, or whether further tweaks will yield improved performance.

Consequently, we devised an algorithm to *learn* such rules or patterns directly from the alignment between datums and papers. We treated the problem of learning extraction patterns as one of sentence classification, where the representation of a sentence is its dependency parse tree. Let us suppose that we wish to learn extraction patterns for the *treatment* used in an assay. The fact that datums refer to the subfigures in a paper in which an experiment is presented, combined with a simple ability to recognize subfigure references in the text, enable us to harvest treatment-expressing sentences (more generally, sentences expressing any of the above four properties) in sufficient volumes for pattern learning. We simply posit that any sentence that (1) refers to the same subfigure as a datum, and (2) mentions the biological entity in the *treatment* role in that datum, is a member of the class "expresses experiment treatment." Sentences outside this class are sampled randomly from among all other corpus sentences and used as negative examples.

Given this binary partition of sentences, we seek to learn a set of patterns that accounts for the set of positive sentences against the backdrop of negative sentences. As documented in Freitag & Niekrasz (2016), the learning procedure is an instance of top-down set covering, a well-tested paradigm in symbolic machine learning, but the pattern representation and search procedure contains some novel elements. Like a parse, a pattern is a tree, the elements of which must be aligned one-to-one with a subset of parse elements in order for the pattern to be said to "match" the parse. However, pattern nodes and arcs admit several dimensions of generalization that allow for an interesting and powerful mixture of matching constraints. For example, a node may require that a particular word be present, that it have a particular part of speech, that it be tagged as a protein, etc. In addition, the pattern language provides what we call "kleene" edges, any one of which matches zero or more parse edges in sequence up or down the tree. Such edges enable concise capture of long-distance relationships, at some increased expense in search during pattern growth. The growth of an individual pattern always starts with a single wildcard node that matches any word in a sentence. We then specialize this pattern step by step, adding edges and imposing node and edge constraints to decrease the number of negative examples matched by the pattern while retaining as many positive examples as possible.

By default, this matching procedure only identifies sentences that may contain target information. For example, it can indicate that a given sentence expresses the subject of an experiment, but provides no information about which phrase corresponds to the subject. With a small enhancement to the procedure, we were able to implement this missing extractive functionality. By constraining the root node to match the target phrase during training, we grow patterns with known focus on the key information. Extraction then involves returning the phrases aligned to the root node in any matches.

Datum Assembly

When applied to any given paper, rules for the four canonical datum fields derived from the datum database yield a wealth of fragmentary information about biological experiments described in the paper. It remains to assemble this information into the simplified experiment representations described above, and to use these representations to assess statements about the

mechanistic relations involved in signaling pathways. This is a task that is more difficult than might be initially apparent. Recall that all four elements of a datum need not be expressed in a single sentence. One sentence may say that Protein A was applied to a cellular environment (i.e., was the "treatment"). The next sentence may list several variants of Protein B that were studied (each the "subject" of a separate datum). The following sentences may discuss observed outcomes, etc. Thus, the initial single protein mention is a legitimate field value for multiple datums. Its candidacy as a field value must somehow be retained by an extraction process that continues to read further sentences until it has enough to populate one or more datums.

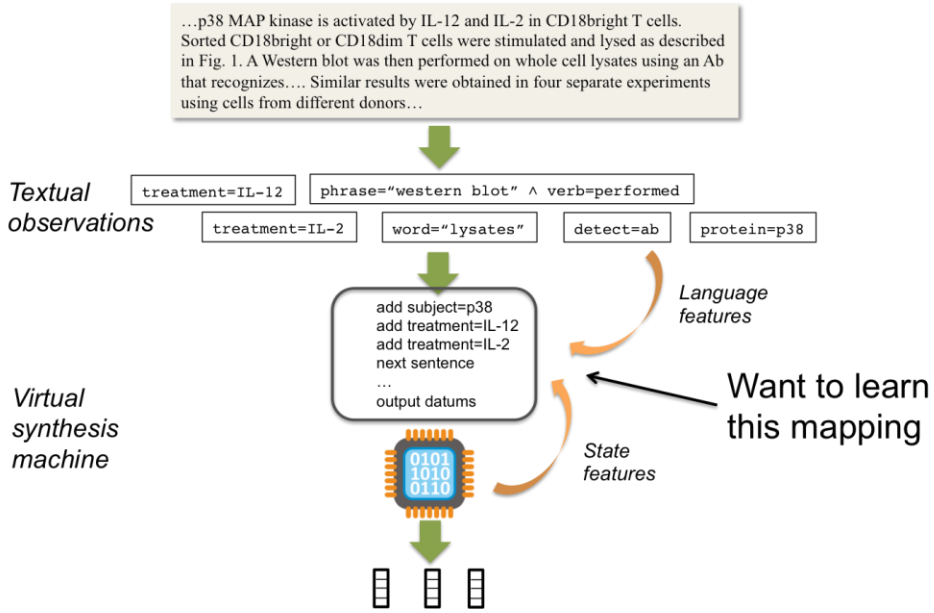

Figure 2: Overview of the empirical datum synthesis machine.

In implementing the Evidence Expert we implemented a procedure that stitched together element-level extractions presenting in this way, but subsequent tests shows that its accuracy was quite low. And it is at least as hard to optimize an assembly procedure like this as to improve manually written rules. Consequently, as with the rules for individual elements, we sought to learn the best assembly procedure from annotations. We approached this problem as one of structured classification, through an instantiation of the SEARN paradigm that has proven effective for comparable problems (Freitag, et al, 2017). Note that the problem and our approach to it have some distinctly novel features. Whereas most structured classification problems motivated by natural language processing adopt a word-level representation, here we range over sentences and the wealth of potentially important information they contain. Thus, a critical challenge is to manage a potentially huge feature space while maintaining efficiency during training.

As shown in the figure, we are modeling the problem of datum assembly as one of structured classification over biological articles. We want to train a system to accept an article and emit a sequence of instructions to a virtual datum assembly machine. We imagine that the machine iterates over the input document, making decisions at each sentence about what to do with any information present. Thus, the machine learning model maps from machine state (a combination of internal registers and location in the document) to next instruction.

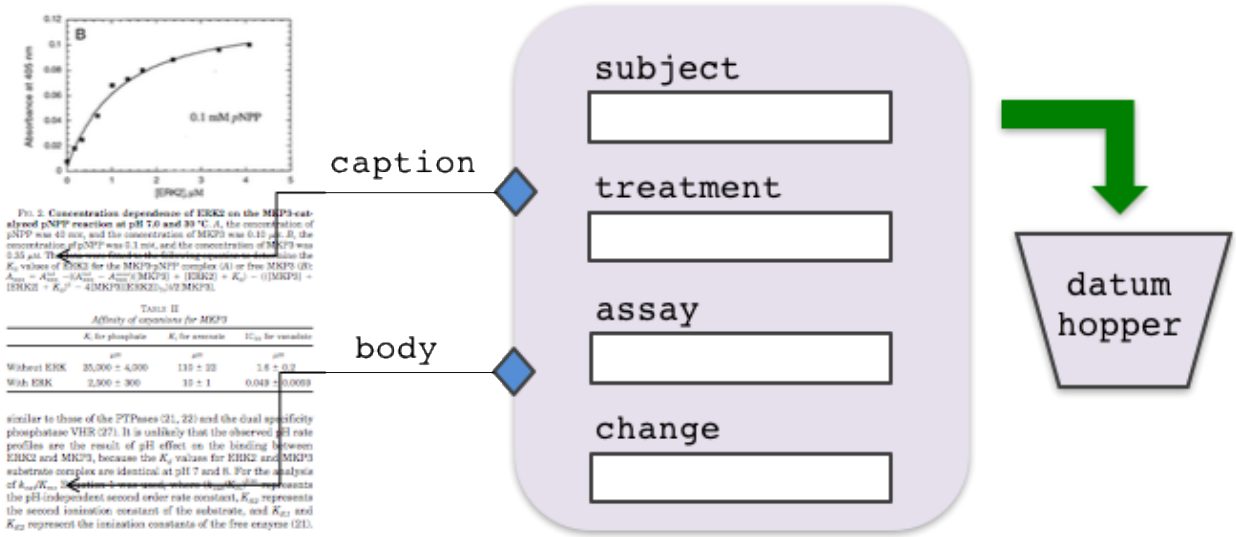

Figure 3: The virtual datum assembly machine.

Figure 3 provides a little bit more detail about the assembly machine. We imagine a machine capable of performing three kinds of actions: movement across an input article (represented as two distinct sentence sequences, corresponding to the contents of captions and the body proper); changes to its internal registers, one for each of the four key fields we seek to extract; and datum extraction, a process that reads the contents of registers and produces simplified datums.

Table 1: Features used in training the assembly machine.

Type	Feature	Description
Cursor	`atPosition(curs, pos)`	True if the cursor `curs` (body or caption) is at the indicated `pos` in its section (beginning, internal, end)
Register	`populated(reg)`	True if the indicated `reg` (subject, treatment, assay, or change) is populated.
	`cregContains(reg, val)`	True if a closed-class register `reg` (assay or change) contains a particular value `val` legal for that type (e.g. the assay register contains "phos").
	`oregContains(curs, reg)`	True if the open-class register `reg` contains the entity under the cursor `curs`.
	`allPopulated`	True if all four registers are populated.
Lexical	`sentContains(curs, word)`	True if the sentence under `curs` contains `word`.
	`wordAtOffset(curs, offs, word)`	True if `word` is observed at offset `offs`, ranging over $[-2, +2]$, from `curs`.
Pattern	`activeAtSent(pat, curs)`	True if the detection pattern `pat` matches the sentence under `curs`.
	`activeAtEnt(pat, curs)`	True if the pattern `pat` matches the entity under `curs`.
Other	`producedDatums`	True immediately after a produceDatums instruction has been executed.
	`bias`	Always true.

At any given point in document traversal, the assembly machine has potential access to a wealth of contextual information, which, in order to apply machine learning, must be represented as features. Thus, success in datum assembly essentially hinges on feature engineering. In order for the empirical datum assembler to perform well, relevant aspects of its state must be made visible to the empirical model. Table 1 summarizes the features used in our experiments to train this model.

We also implemented a conceptually straightforward alternative approach, which we call the *frame classifier*. This approach two of the four fields in a simplified datum are extracted from the text (subject and treatment), while the other two involve classification into a small fixed vocabulary (assay and change). In other words, the subject of a datum is almost always a protein referenced by name in the text (e.g., "K-Ras"). In contrast, there is a fixed number of assay types recognized in Pathway Logic (with labels such as "phos" and "cooptby") and an even smaller number of possible outcomes ("increased", "decreased", "unchanged").

2.2. AMPLIFYING HUMAN EFFORT

One major focus of our research has been to investigate the question of how limited user intervention can be coupled with a distant supervision approach to learn more accurate information-extraction models than distant supervision alone can provide. Our project sought to exploit the Pathway Logic Knowledge Base as a rich resource for training information-extraction models. Given that the Pathway Logic Knowledge Base contains structured information characterizing thousands of experimental results, we have employed it as a source of distant supervision to learn models to extract predicates that describe various aspects of protein signaling experiments. The specific predicates that we have extracted specify such conditions as whether a protein or small molecule was suppressed via RNA interference, a knockout, or an omission, and whether a given protein was found to be required or not required for each experimental result.

The principal advantage of the distant supervision approach is that it enables existing knowledge-base resources to be used to automatically label a text corpus, thus obviating expensive manual labeling. The key limitation of distant supervision, however, is that it may provide noisy labels for training. An underlying hypothesis of much of our work is that limited user intervention can be coupled with a distant supervision approach in order to learn more accurate information-extraction models than distant supervision alone.

We investigated several ways in which user interaction can be interleaved into a distant supervision setting. One approach we devised is based on using a human expert to identify words and phrases, which we refer to as *trigger words*, that are associated with actual positive instances of a target predicate. Our approach involves (1) ranking words and phrases according to their statistical association with instances which have been labeled as positive by distant supervision, (2) presenting these words to a user and asking them to identify which are truly semantically linked to positive instances, (3) and using the expert-identified words either as additional features or to change the labels of training instances.

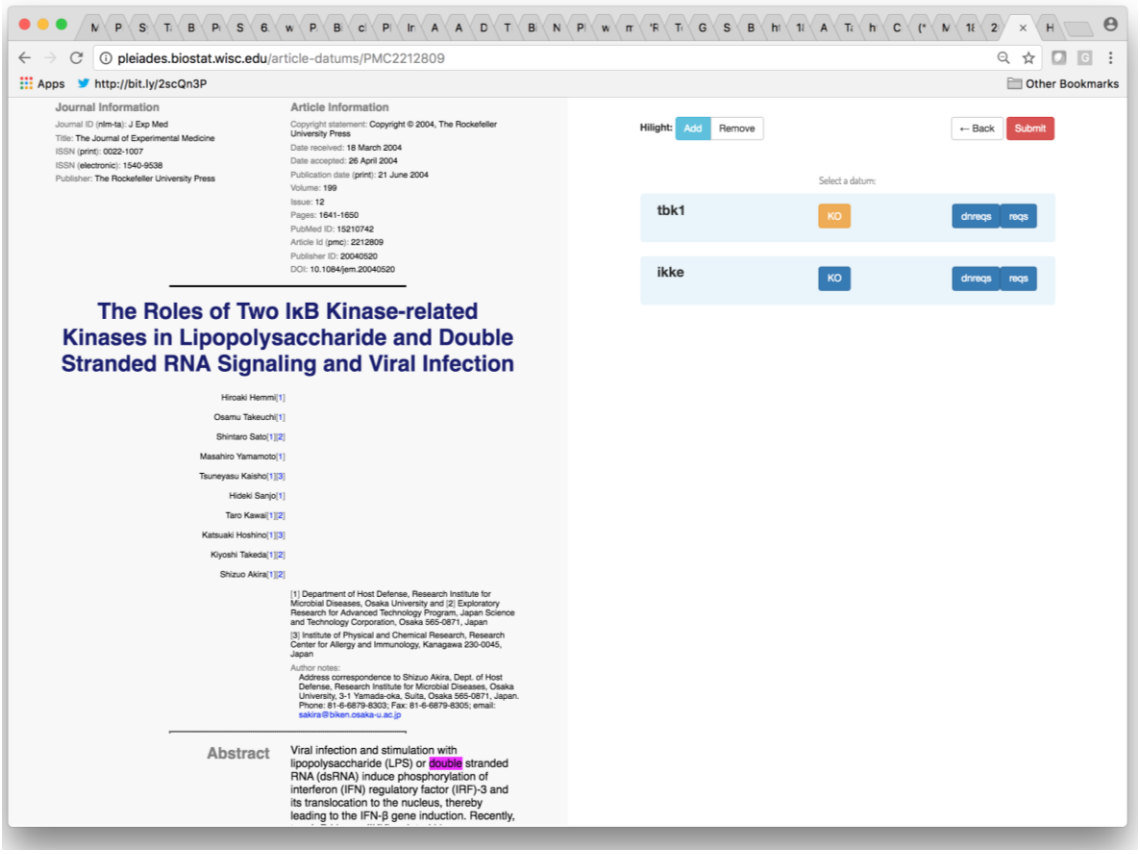

Figure 4: A screenshot of the interface we developed to enable a user to selective revise the labelings provided by distant supervision.

A second approach we explored involved developing a user interface that allows an expert user to selectively relabel some articles that have been labeled via distant supervision. The interface enables a user to (1) browse a distantly labeled corpus seeing the instances of each predicate that have been labeled in each article, (2) browse an article seeing the highlighted mentions of proteins of interest along with occurrences of trigger words, (3) mark the passages of text that are relevant to the predicate of interest. Figure 4 shows one mode of the interface in which a specific article has been selected for user annotation. The annotations mark relevant passages for the predicate in selected articles and are intended to provide more informative training data when learning via distant supervision. Our experiments have shown that most articles can be annotated in five minutes or less, but that the incorporation these articles into the training set does not have a consistently large effect on predictive accuracy across predicates.

Another line of investigation we pursued was to couple multiple-instance learning approaches with distantly labeled training data. The rationale for this approach is as follows: Because we are learning via distant supervision, our training data is not labeled at the right level of granularity. For a given article in the training set, we know which predicates should be extracted from it, but we don't know which passages in the article support each extraction. By using a multiple-instance approach, we can train a model using the (high-accuracy) article-level labels instead of having to use them to derive (low-accuracy) passage-level labels. We investigated various neural-net based multiple-instance architectures and found that, for most predicates, we are able to learn more accurate models when using a multiple-instance approach.

We also conducted a study in which we explored how literature-extracted information can be used to more accurately infer the subnetworks involved in specific biological processes of interest. Many biological studies involve either (i) manipulating some aspect of a cell or its environment and then simultaneously measuring the effect on thousands of genes, or (ii) systematically manipulating each gene and then measuring the effect on some response of interest. A common challenge that arises in these studies is to explain how genes identified as relevant in the given experiment are organized into a subnetwork that accounts for the response of interest. The task of inferring a subnetwork is typically dependent on the information available in publicly available, structured databases, which suffer from incompleteness. However, a wealth of potentially relevant information resides in the scientific literature, such as information about genes associated with certain concepts of interest, as well as interactions that occur among various biological entities. We contend that by exploiting this information, we can improve the explanatory power and accuracy of subnetwork inference in multiple applications. We proposed and investigated several ways in which information extracted from the scientific literature can be used to augment subnetwork inference.

2.3. HIGH-FIDELITY BIOLOGICAL MODELS

The objective of this focus area was to formalize the meaning of experimental findings as they relate to signaling reactions (represented as rewrite rules) and provide insights about how cells work. Specifically, we want to infer the most detailed description of a signaling event supported by given experimental evidence. Furthermore, we want to assemble models for specific questions from a collection of rules. This will require sometimes abstracting from details.

Our starting point is a formal representation of the experiments that yield the evidence upon which biological insights are predicated, which our Pathway Logic (PL) project calls a "datum." Datums provide a computable representation of experimental results using a controlled vocabulary.[2] To formalize what datums mean, we first map a set of datums to a set of logical assertions formalizing what was done and what was observed. Logical axioms relate these ground facts to constraints on rule schemas. Constraint solving then gives possible models of the facts and axioms (values of the schema variables) from which rules can be extracted.

As a result of a discussion with Paul Cohen about how PL can make more contact and be more relevant to the BigMechanism program this task was expanded to include developing a BioCommonSense component, including background knowledge and rules for use. The target application is to be able to check reader output (index cards) for errors or unlikely statements to be further checked. It is synergystic with the PL assembly work since both efforts need to have more background knowledge formalized along with rules for using that knowledge. This task was further adapted to include some support for evaluation exercises.

PL models, including the rules that formalize the changes in cellular state that constitute signaling and the datums from which they were derived, were at the heart of the SRI team's contributions to program evaluations. These models were not only the most detailed accounts of the specific mechanisms associated with Ras-driven cancer in humans, they were *executable*, a feature that some of the other models proposed by performers in the program lacked. One could prime the model with a particular state representing some cellular environment, emulate the

[2] We use the unconventional plural "datums" to distinguish these formal objects from generic data.

intervention applied in some assay, and ask the model to show why some downstream effect was observed. Thus, PL models directly address on of the cardinal objectives of the Big Mechanism program: explanation of large, partially opaque real-world mechanisms. In cases where a model was insufficient to reproduce an experimental result, we developed methods to extend it automatically to increase its coverage of biological phenomena, as described below in Section 2.4.

Assembly.

We investigated the use of abductive reasoning to infer pathways explaining effects of drug treatment. A mapping from PL models (dishnets) to input for the DLV reasoner was defined. This included a small number of model-independent assertions reflecting the underlying semantics. Also, several query patterns were defined. Preliminary experiments indicate that this scales and produces plausible explanations. What remains is to automate the extraction of useful explanations from the models generated by DLV.

Towards automated assembly of rules from datums, assertions have been redesigned to capture more information, such as details of mutations and modifications of proteins. The assertion generator that maps datum collections to input for an answer set reasoner was extended to extract information from extras and to reason about the meaning of mutations. For example, if mutating a serine site on the subject of a phosphorylation assay decreases the measured phosphorylation (inhibits the reaction), then hypothesize that the mutated site is one of the sites modified. Dually, if mutating a phosphorylation site of a treatment protein decreases the response to a treatment then hypothesize that the treatment protein needs to be phosphorylated to enable the response.

We refactored the inference of PL rules from datums to avoid producing lists of *inhibitedby* extras when they can be directly reduced to requirements, since reasoning about the lists caused combinatorial explosion in the logical reasoner. This required modifying the assertion generator to implement reasoning about inhibition. We introduced the concept of weak rule -- one with only partial evidence. The set of fuzzy matching rules for assembling connected networks was expanded to include lifting modification rules to complexes, inclusion of weak rules by need, and abstraction to families and composites.

We advanced the experiment to use abductive reasoning to explain cellular responses. The core logic was modified to generate accurate pathways. We verified that abductive reasoning scales, at least to the level of the SKMEL133 model. We began defining query templates to facilitate analysis, for example: If B and C are both reachable, how can you reach a state where B holds and not C.

This task involved advancing Pathway Logic in support of Big Mechanism efforts, as well as providing resources including data and models, and biological expertise. The Pathway Logic Assistant was improved to support visualization and interaction with the much larger models in the current STM knowledge base. This involved interaction with the maintainers of the Graphviz dot graph layout algorithm resulting in reducing a 3.5 hr layout time to about 5 minutes for our benchmark. The handling of graphics in PLA was also improved to make the system responsive to user gestures when working with large models.

The SKMEL 33 model (see Task 5.) has been published as part of the PL collection of models. In addition to the model itself, documentation has been prepared that describes the model and provides a guided tour -- ways to explore and analyze the model. In particular, the user can

reproduce our explanation of the RPPA data. The SKMELL133 model is part of the PLA Online suite (available at pl.csl.sri.com/online.html) and also directly downloadable (available at pl.csl.sri.com/download.html) for those who have installed PLA locally. The guided tour is available directly at pl.csl.sri.com/skmel133-guide.html or as a link from the PLA Online Launcher.

The PL datum collection and rule KB were extended to include information for additional phosphorylation events and rules explaining changes in protein expression to expand the coverage of response to drug treatment. Specifically, 6 new kinases were added to the model and 9 new rules, based on curated experimental evidence. The datum knowledge base was correspondingly extended.

We expanded the PL controlled vocabulary to allow for representing non-human proteins and their interactions with human cells. This allows modeling response to pathogenic infections as well as to drugs.

A new datum parser was completed, with the following features:

- An order of magnitude faster.
- Easier to install (only depends on Java, not Ruby, which required dependencies that could break at any time).
- More in-depth sanity checks (biological type checks).
- Sanity checking is extensible, readily accommodating new experimental information and constraints.
- More useful error messages and output.
- Easier to update and maintain, thanks to simpler semantics and a consistent internal representation of datums.
- Lays groundwork for building other tools that use it, such as a datum query language.

The parser development version is on github. The packaged jar will be available as part of the PL distribution in the future, being self-contained and easy to use. This opens the opportunity for new forms of reasoning within the PL system by combining information from both the datum and rules knowledge bases.

We developed a query language and web interface to make the information contained in the datum knowledge base (which currently contains more than 70,000 datums) accessible to a broader community. Documentation is available online at pl.csl.sri.com/datumkb.html, and the query interface is available at datum.csl.sri.com.

We developed a tool to mine the Phosphosite database for pubmed ids of papers containing evidence of phosphorylation reactions, and added new kinase rules to STM Common Rules based on these papers.

2.4. AUTOMATED MODEL EXTENSION

From the outset, we have approached “explanation” not only as an end in itself, but also as a provider of high-level “sense-making” feedback to assembly and reading. We believe that this interplay between bottom-up and top-down reasoning is essential to the scientific process, and makes reading onto a mechanistic model far more promising than either machine reading or modeling in isolation.

We began our work by formalizing this concept mathematically in terms of Bayesian machine learning concepts. This was set out in our working paper *"Sketch of a meta-model for co-evaluation of theories and data credibility"* (explModel.pdf), later updated to *"Design of a reasoning engine and associated credibility model"* (explModel2.pdf). In this framework, the role of a trainable model is played by a knowledge base containing rules that describe reaction steps in cellular signaling processes, and the role of training data is played by the results of laboratory experiments. A Bayesian prior is introduced to express a preference for concise rule sets that explain relatively many experiments (which can only be accomplished by introducing underlying mechanisms) over verbose rule sets that offer little mechanistic insight beyond restatement of the data. Aside from the rather unusual treatment of a knowledge base as a model, our framework allows the data, as well as the knowledge base, to be treated as uncertain. In this way we can capture some of the tension that exists in the scientific process, as practiced, between revising a heretofore successful theory and accepting new experimental data that contradicts it. We believe this capability will also be important for robust performance in the face of machine reading errors. This feature can also be used to express relative prior certainty about a rule by introducing an artificial data point that asserts the rule itself.

The treatment of a knowledge base as a model was motivated in part by the availability of the Pathway Logic knowledgebase and other bio-chemical knowledge bases as a starting point, and in part by the preponderance of machine reading output in the form of assertions that are more amenable to interpretation as discrete rules than as components of process models expressed over real-valued time and concentration variables. Our initial plan was to use the Pathway Logic reasoning engine as the executable model, but found that for our particular purposes we obtained much faster execution with less software integration overhead by writing our own reasoning engine using sparse matrix routines in Python. (We checked for agreement with the Pathway Logic reasoner.) To keep it fast and simple, we restricted the functionality to propositional reasoning rather than write a first-order reasoner. In practical terms, this means that one must decide in advance the set of variables to be used in the simulations, and the rules that will operate over them. This information is "compiled" into a sparse matrix representation of a dynamical system operating over a "state space" consisting of a set of Boolean variables, each of which asserts the presence or absence of a particular chemical entity. Rules can be turned on and off (or to some extent, modified) without re-compiling, but re-compilation is necessary when new variables are introduced. The reasoning engine supports negation, so it is possible to assert that a reactant is removed, or that a result requires the absence of a reactant.

Capabilities were added to the reasoning engine during the course of the project. The first implementation had no support for negation, making it impossible to assert that a result requires the absence of an entity, or that an entity becomes depleted. Such assertions are important for inferring causality, so one of our first improvements was to support negation. This was easily accommodated with minor modifications to the sparse matrix algebra. We also wanted to be able to control the level of specificity or generality with which rules and data points were asserted, partly because assertions in the literature vary widely in their specification of, for example, specific proteins as opposed to protein families, or the details of protein modifications. We also wanted to be able to adjust expression of generality in order to support searching for the most general rules that support all the given data; more specific rules being unjustifiably complex, given the experimental evidence. (As mentioned above, prior knowledge that is not implied by the experimental evidence can be supplied using artificial data that directly asserts the validity of particular rules. We have not had an occasion to exercise capability, however.) We

therefore introduced support for *"ontological rules"* stating that instances of a specific category are also instances of a more general category. We then started calling the ordinary rules that describe reaction steps *"dynamical rules"*. The inference engine treats the ontological rules and the *"dynamical rules"* identically, except that the ontological rules are iterated to convergence, so that all generalizations are inferred, between each application of the dynamical rules.

As discussed in the section on Task 8, we eventually deemed the ontological rules to be an inadequate solution. The entities involved in the dynamical rules have hierarchical structure, some parts of which vary in ontological generality. Realizing that trying to describe this situation in terms of an ontology at the level of complex structures is a seriously flawed approach, we migrated from a representation with one Boolean variable per entity to using multiple variables per entity so that both structural and ontological variants could be readily represented.

An initial state must be supplied in order to run a simulation. Pathway Logic provides a selection of initial conditions appropriate to various classes of experiments, from which we have chosen a common one for experiments involving treatment with Egf, called *EgfDish*. This a temporary expedient. Experiments are described in terms of entities such as growth medium, cell line and treatment. Therefore the inference chain should begin with these types of entities, and rules asserting that particular growth media and certain cell lines contain certain reactants. We created a rule of this form, calling it an *"initialization rule"*, that produces the contents of *EgfDish* from any medium. This is merely a placeholder. Our intention is to develop more realistic initialization rules for various cell lines, and perhaps to subject them to Bayesian optimization, just like the other dynamical rules, thereby using experimental evidence to infer how one cell line differs from another.

Similarly, we must introduce rules that connect the final state of the dynamical system to the type of assertions made about experimental outcomes. (Here we assume convergence; we discuss non-trivial limit cycles in the section on Task 8.) Experiments are described in terms of the measurement procedure, or assay that is performed, a particular reactant, the *subject*, of the assay, and an *outcome* such as whether the subject increased or decreased in quantity, or neither. We therefore introduced another class of dynamical rules we call *"observation rules"* to describe how the state, assay and subject are related to outcomes. To determine whether or not a simulation agrees with a data point, we compare the simulated and reported outcomes. We have written observation rules covering several common types of assay, but still better coverage would be desirable.

We have described the use of Bayesian principles to provide a score function for sets of rules. To find highly explanatory sets of rules, it is also necessary to generate candidate rule sets that are likely to score well. The candidates can be derived by varying the contents of existing knowledge bases such as Pathway Logic, from automation of the manual process of inferring rules from data such as described in the section on Task 5, and from machine reading. As described above, and under Task 8, we have created an infrastructure that enables us to explore structural and ontological variants of the entities appearing in rules in an existing knowledge base. In order to make use of the output of machine reading or the output of automated rule inference algorithms, it is necessary to introduce processing that transforms their fragmentary and incomplete output into plausible, well-formed rules. We engaged in two lines of effort to support this objective. In one, we carried out clustering experiments on STM7 to find groups of similar rules that can be used to define plausibility metrics describing whether a nominated rule

"looks like" a rule that would not be out of place in STM7. In the other, we developed various heuristics to assemble multiple incomplete rules into single well-formed rules.

In order to actually generate explanations, it is necessary to examine the execution traces of the inference engine. We created visualization software to assist with this, and also to help with debugging and checking that the system works as expected.

We restructured the code to improve management of adding new data sources (readers or knowledge bases) to the sense-making layer, to define and manage experiments involving combinations of data sources, and to archive the pre-processing and the results reproducibly. Having moved from a local to a distributed representation of the interacting reactants, graph matching became involved in aligning rules onto the state, which increased our computation requirements. Therefore we parallelized the stochastic search for optimal rule combinations over a cluster of 7 multi-core machines, with 8 processes per machine, giving us a speed-up of about 50X.

Several experiments were carried out aiming to demonstrate that readers could improve the curated STM7 knowledgebase. It was found that adding rules produced by readers did enable the knowledgebase to explain more experiments, but for the spurious reason that the reader rules are overly simplistic, which makes them fire to easily, leading to overproduction of reactants, often including the one the assay tested for. Control experiments would penalize the overproduction, but our data has too few of these. Therefore we realized that other methods would need to be devised to eliminate overly simplistic rules. This led us to develop two methods that incorporate common-sense knowledge into rules obtained from readers.

Upon examination of FRIES reader failures in a Mitre evaluation, we noticed that in many cases, the entities and structural components required to produce a correct reading were successfully found by REACH, but the relationships between these entities were either missed or incorrectly labeled. We therefore ignored the assembly attempts made by the reader and took an exhaustive combinatorial approach to assembly, starting directly from the entity extractions. Sentence by sentence, we collected all the extracted actions, proteins, modifications, sites, locations, etc., and assembled them into one or more molecular structures in all possible ways (typically hundreds or a few thousand). These structures were then assembled into rules in all possible ways. Every such assembled rule was added to the knowledgebase in turn, and evaluated against experimental data to determine which assemblies were more plausible than which. To limit the combinatorial search, we codified expert prior knowledge into a *bio-common-sense* module and supplemented this with a simple statistical model of the likely structure of reactants based on those appearing in STM7. The model computes a score as weighted mean of log probabilities of occurrence and co-occurrence probabilities between proteins, modifications, sites and locations, and structure size. These scores were used to define rule scores by averaging over the reactants appearing in the antecedents and consequents of the rules that were generated combinatorically for the Mitre evaluation post analysis. This, finally, led to obtaining useful results from reading.

We studied linking up CMU's DySE model with Pathway Logic (PL). The DySE model is broadly similar to our SAMM model, both being Boolean dynamical simulations. The DySE model starts from a manually constructed core of about 300 rules, and adds read rules to this core in various ways. It operates at a lower level of chemical detail than PL, but has broader coverage, including inter-cellular and phenomenological behavior. DySE is trained on about 20

desired properties of the execution traces, whereas SAMM is normally trained on larger numbers of experiment descriptions. While DySE operates at a coarser level of both mechanistic and representational detail, there is a large enough of overlap between the base elements used in both frameworks to draw alignments between them. Out of 309 distinct rules in DySE, we found 66 that could be accounted for by PL. 82 additional rules have one element aligned with an element in PL, but have their other element remaining unmapped. For each combination of the initial and final entities in an alignable DySE rule, we can obtain the sequence of PL rules and proteins activated in a traversal between them.

During the final year of the project, including the no-cost extension, we include several novel innovations on top of the previously derived explanation platforms. The key improvements included a closed-loop reading model, a Markov Chain Monte Carlo (MCMC) rules testing system, and the ability to work with multiple cell lines simultaneously (extending the previous STM7 focused work).

Closed-loop reading shifted the paradigm from the readers feeding the explanation platform a slew of poorly curated results, to the explanation platform querying readers for specifically relevant explanation. The explanation platform would identify gaps within the current knowdelge base, based on identifying specific protein modifications that were missing explanations. For example, if an assay said Raptor-phos!S792 was observed but no current system rules were available to produce it this would generate a query specifically looking for suggestions from automated readers that could fill this gap. This shift in approach allowed a more principled review of the literature, and more closely matched the human process used in extending the Pathway Logic model.

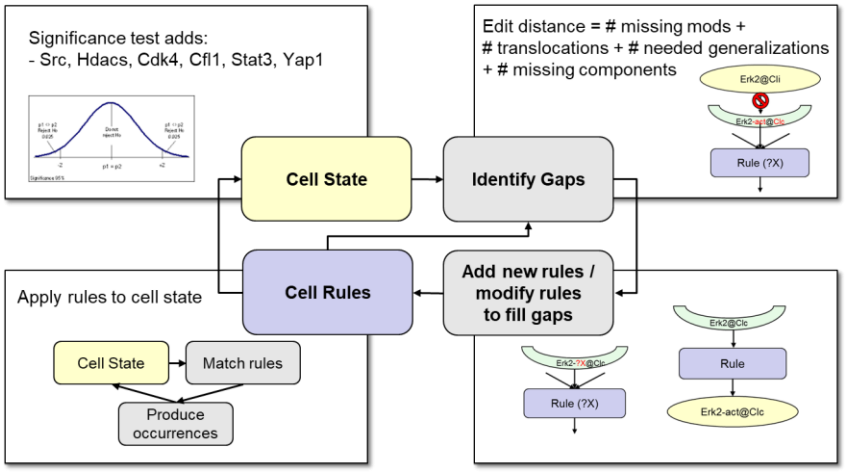

Figure 5. Closed loop reading procedure, that identifies gaps in the explanation model, uses machine reading to propose new rules, checks rules against the experimental evidence, and updates the cell state to based on statistical tests.

The queries generated by the explanation platform would be used to either search through a database of read results or directly fed to a reader through a web based interface, allowing the reader to use the context to search for all relevant extracts. Once the response was received the system would search through the set of responses, and gather evidence for the proposed rule (or rules). This would entail construction of a set of candidate rules that could fill the gap, including specification of optional modifications such as specific phosphorylation sites and compounds. Each candidate rule would then be scored based on a set of criteria including the number of reading extracts that supported the rule, the specificity with which those extracts supported the rule, and the agreement with biological common sense. The first two used straightforward metrics (number of mentions in different papers, number of mentions of the specific phosphorylation site, etc.) while biological common sense was evaluated against a statistical common sense module that was trained through machine learning.

The statistical common sense module was trained based on the existing pathway logic rules. Each protein (and modification) was characterized by a set of keywords curated from Uniprot. The types of reactions that each protein participated in were also characterized (including whether it was a control in the interaction or if it was itself modified). The machine learning model then evaluated the likelihood of a rule given the keywords associated with its constituents. For example, the system learned that phosphorylation reactions were likely to be performed by proteins with the keyword "kinase".

The closed loop reading results were fed into the existing explanation platform, and compared to observed experimental results (either using the STM7 model, or later using the MCMC based multi-cell line model). Accepted rules would be used to extend the model, providing a better explanation of previously unexplained observations. The process would then be repeated with the new model generating additional queries to fill newly revealed gaps until convergence.

Markov Chain Monte Carlo (MCMC) sampling was used to extend the previous stochastic search algorithms used in the STM7 model to support a more targeted search of the rule space, and to eventually support more complex multi-cell line rule sampling. The MCMC system considered a set of rules simultaneously and asked the question, "Which subset of the proposed rules best explains experimental observations." This compares to the previous STM7 model which focused on single rule extensions of the system, or stochastic search over multiple extensions.

The MCMC model extended evaluation to include all experimental results, including increase, decrease and unchanged. The last result was particularly critical in removing over-explanations that would add rules that would result in incorrect reactions. A regularization term was used in the MCMC method to ensure parsimony, penalizing the overall agreement score based on the number of additional rules and additional proteins added to the model. Together these mechanisms restrained the complexity of the model extensions, allowing search for single rules that explained the most experiments. This procedure also combatted the tendency of the system to create long jump rules that explained novel biological observations through a single rule connecting the inputs to the outputs. In practice these rules are not useful, as they short circuit the explanation system. The combination of curating rules for biological common sense and of selecting for rules that provide partial explanations for multiple results disfavors such long jumps and improves the quality of the overall explanation system.

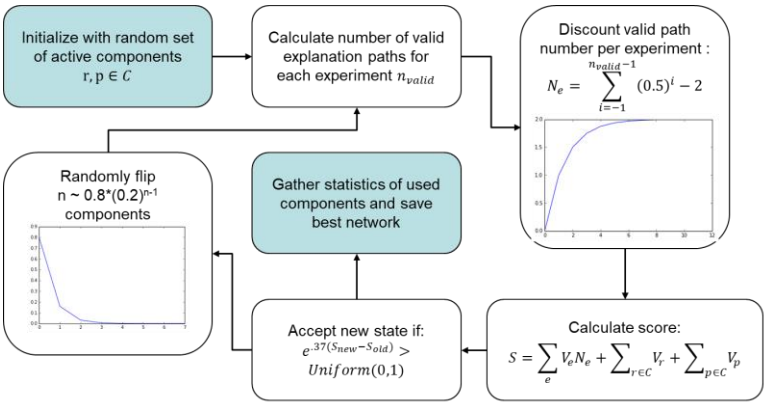

Figure 6. MCMC sampling procedure for evaluating rule utility based on experimental agreement, biological plausibility, reader agreement and parsimony.

The MCMC system was integrated with the closed-loop reading system providing evaluation of candidate rules. Initially only high quality rule candidates were fed to the MCMC evaluation system, however, we realized that some useful rules were being discarded early due to not having much support in the literature. We then shifted to a late decision paradigm, where the closed-loop reading rule evaluation resulted in a score that was incorporated into the MCMC

procedure as a prior for each proposed rule. The prior was combined with a likelihood based on experimental agreement to produce a posterior that was used to drive MCMC sampling as shown in Figure 2. The MCMC system allowed us to both evaluate a best coherent rule system, as well as evaluating the probability that a given proposed rule was used in any of the MCMC sampled explanation systems.

Multi-cell line explanation was the final major change introduced to the explanation system. Where previously we focused on a single (STM7) cell line, we extended the platform to allow for multi-cell line testing in support of the later evaluations and to make use of additional Pathway Logic curated data for experimental verification. To derive the STM7 model human curators selected a set of proteins present in the initial system state, and customized general rules to connect to these proteins and reproduce the experiments. Previously, we had started from this post-curation model, and used stochastic search with the explanation platform to vet new rules and to measure recovery of ablated rules. The multi-cell line extension pushed us further backwards, automating the manual model customization process. Here the system would automatically propose an initial protein state for each cell line, while simultaneously testing new rules and connecting existing generic rules together to create a cell line specific model. The explanation system would then suggest new generic rules that worked across cell lines using the MCMC rule selection procedure described above.

To extend the MCMC system to multiple cell lines we simultaneously searched over generally applicable rules and cell line specific initial states. A single MCMC sample consisted of proposed initial state for each tested cell line and a subset of the rule candidates. The overall explanatory power of this set was evaluated, including the prior based on biological common sense and machine reading and the likelihood of all experimental observations curation in the Pathway Logic Database for all tested cell lines. After completion we would have a best ruleset that applied across all cell lines, the likely initial cell line states, and the rules that are active for any given cell line. In particular, the MCMC approach allowed us to identify rules critical for explanation across multiple experiments, and how likely particular proteins were to be present in different cell lines based on the MCMC sampling across explanations.

3. RESULTS

In this section, we briefly summarize the salient experimental results we obtained under the three top-level tasks our project comprised: *Reading for Evidence*, *Amplifying Human Effort*, *High-fidelity Biological Models,* and *Automated Model Extension*. Please refer to Section 2 for a description of the technical approaches underlying these results.

3.1. READING FOR EVIDENCE

The Value Proposition

When considering the prospect of extracting information such as experiment frames from biological articles, as opposed to the more general factual information that is typically the focus of biological extraction, it is important to ask whether it is worth the trouble. Extracting these frames is much more difficult than generic extraction for three reasons: they have more fields than typical factoids, which are binary relations; the information out of which they are

constituted is, in general, distributed over multiple sentences; and we lack phrase-level annotations for these pieces of information. What justifies the effort of extracting these frames?

We analyzed the results of the first program evaluation and were able to show that experiment frame extraction provides an important corroborative signal, enabling us to filter generic extracted facts. The analysis looked at the "index cards" produced by the system submitted by the FRIES consortium, of which we were part. Index cards were essentially the program's formalization of individual extractions output by program readers. The key insight is that some of these extractions cover types of interactions directly related to the assays we attempted to recognize. If, for example, a paper states that Protein A phosphorylates Protein B, it is reasonable to look for an assay that looks for phosphorylations and in which Protein A and Protein B are the treatment and subject, respectively. Extraction of such an assay can be taken as corroboration that the relation in question actually holds. We deployed an "Evidence Expert" that processed facts extracted by other readers and attempted to align them to extracted experiment frames in this fashion. Of the approximately 21K index cards FRIES readers produced, 47% involved interaction types that the component supported. Of these, 36% were deemed to be fully substantiated by extracted datums, 29% partially substantiated, and 33% were discarded as having no experimental support.

Table 2: Index card precision as a function of corroboration level.

Degree of substantiation	Fraction of FRIES index cards	Strict Precision (Info) over MITRE-graded cards
All cards	100%	50%
Unsupported	53%	N/A
None	17%	43%
Partial	13%	60%
Full	17%	80%

The key question is whether corroborated index cards are actually more likely to be correct, according to human experts. Table 2 presents the result of this analysis, showing the value of "strict information precision" of subsets of these cards grouped by corroboration level. As the table indicates, the baseline precision of FRIES readers was about 50%. It also appears to be the case that the stricter the level of experimental corroboration we required of a card, the higher its precision, with full corroboration corresponding to roughly 80% precision. Because the number of cards graded in this way was small, caution is required in the use of these results. However, we assessed the statistical significance of some of the observed improvements and found, for example, that the improvement of "Full" over "None" is significant at the 95% level.

These results gave us confidence that the task of extracting experiment frames was both feasible and relevant to program objectives, but the work was far from complete. The coverage of the

Evidence Expert was modest in terms of both the number of reader assertions it could assess and the number of salient assay types it supported. Only five types were implemented for this analysis, out of perhaps a dozen types that are frequent in the datum knowledge base.

Table 3: Accuracy of the Evidence Expert in detecting manually created datums.

# Matching Elements	Precision	Recall	F1
1	0.189	0.901	0.223
2	0.095	0.604	0.128
3	0.018	0.151	0.026
4	0.001	0.014	0.002

More critically, there was evidence that the implementation of the Evidence Expert, which involved the heuristic assembly of frames (4-tuples) from elements matched by hand-authored extraction rules, left much room for improvement. As summarized in Table 3, we evaluated the Evidence Expert's ability to recover the key details of experiments (specifically, the four datum fields it targets) by comparing its extractions from papers for which datums have been created. As the table suggests, while it was modestly successful at recovering individual properties, complete agreement with respect to all four required fields posed a stiff challenge. Note that care must be taken in interpreting these numbers, as there is no guarantee that a given datum is described in the text (it may be communicated only in a figure), nor that all experiments described in a paper have been captured in a datum. Nevertheless, it appeared safe to conclude that more research was warranted.

Finding Experiment Elements

The validation we received from the analysis described above motivated us to replace the laboriously created extraction rules of the Evidence Expert with rules learned directly from the data, in the hope of obtaining better scalability, recall, and precision. As detailed in Freitag and Niekrasz, 2016, we developed an approach that automatically produced extraction rules against dependency parses, based on a simple binary separation of sample sentences into those that express a certain kind of information and those that do not.

Table 4: Performance of automatically learned extraction patterns.

Assay	Role	Learned	Written	Baseline
phos	subject	0.48/0.62/**0.54**	0.32/0.45/0.37	0.17/1.0/0.29
	treatment	0.46/0.51/**0.48**	0.41/0.32/0.35	0.17/1.0/0.29

ubiq	subject	0.57/0.50/**0.53**	0.38/0.43/0.41	0.01/1.0/0.02
	treatment	0.26/0.32/**0.29**	0.50/0.11/0.17	0.004/1.0/0.01
GTP-association	subject	0.56/0.65/**0.60**	0.17/0.25/0.20	0.005/1.0/0.01
	treatment	0.21/0.67/**0.32**	0.08/0.03/0.05	0.005/1.0/0.01
Any	subject	0.38/0.83/**0.52**	0.12/0.36/0.18	0.17/1.0/0.29
	treatment	0.48/0.64/**0.55**	0.18/0.38/0.24	0.17/1.0/0.29

The Value Proposition

When considering the prospect of extracting information such as experiment frames from biological articles, as opposed to the more general factual information that is typically the focus of biological extraction, it is important to ask whether it is worth the trouble. Extracting these frames is much more difficult than generic extraction for three reasons: they have more fields than typical factoids, which are binary relations; the information out of which they are constituted is, in general, distributed over multiple sentences; and we lack phrase-level annotations for these pieces of information. What justifies the effort of extracting these frames?

We analyzed the results of the first program evaluation and were able to show that experiment frame extraction provides an important corroborative signal, enabling us to filter generic extracted facts. The analysis looked at the "index cards" produced by the system submitted by the FRIES consortium, of which we were part. Index cards were essentially the program's formalization of individual extractions output by program readers. The key insight is that some of these extractions cover types of interactions directly related to the assays we attempted to recognize. If, for example, a paper states that Protein A phosphorylates Protein B, it is reasonable to look for an assay that looks for phosphorylations and in which Protein A and Protein B are the treatment and subject, respectively. Extraction of such an assay can be taken as corroboration that the relation in question actually holds. We deployed an "Evidence Expert" that processed facts extracted by other readers and attempted to align them to extracted experiment frames in this fashion. Of the approximately 21K index cards FRIES readers produced, 47% involved interaction types that the component supported. Of these, 36% were deemed to be fully substantiated by extracted datums, 29% partially substantiated, and 33% were discarded as having no experimental support.

Table 2: Index card precision as a function of corroboration level.

Degree of substantiation	Fraction of FRIES index cards	Strict Precision (Info) over MITRE-graded cards
All cards	100%	50%

Unsupported	53%	N/A
None	17%	43%
Partial	13%	60%
Full	17%	80%

The key question is whether corroborated index cards are actually more likely to be correct, according to human experts. Table 2 presents the result of this analysis, showing the value of "strict information precision" of subsets of these cards grouped by corroboration level. As the table indicates, the baseline precision of FRIES readers was about 50%. It also appears to be the case that the stricter the level of experimental corroboration we required of a card, the higher its precision, with full corroboration corresponding to roughly 80% precision. Because the number of cards graded in this way was small, caution is required in the use of these results. However, we assessed the statistical significance of some of the observed improvements and found, for example, that the improvement of "Full" over "None" is significant at the 95% level.

These results gave us confidence that the task of extracting experiment frames was both feasible and relevant to program objectives, but the work was far from complete. The coverage of the Evidence Expert was modest in terms of both the number of reader assertions it could assess and the number of salient assay types it supported. Only five types were implemented for this analysis, out of perhaps a dozen types that are frequent in the datum knowledge base.

Table 3: Accuracy of the Evidence Expert in detecting manually created datums.

# Matching Elements	Precision	Recall	F1
1	0.189	0.901	0.223
2	0.095	0.604	0.128
3	0.018	0.151	0.026
4	0.001	0.014	0.002

More critically, there was evidence that the implementation of the Evidence Expert, which involved the heuristic assembly of frames (4-tuples) from elements matched by hand-authored extraction rules, left much room for improvement. As summarized in Table 3, we evaluated the Evidence Expert's ability to recover the key details of experiments (specifically, the four datum fields it targets) by comparing its extractions from papers for which datums have been created. As the table suggests, while it was modestly successful at recovering individual properties, complete agreement with respect to all four required fields posed a stiff challenge. Note that care must be taken in interpreting these numbers, as there is no guarantee that a given datum is

described in the text (it may be communicated only in a figure), nor that all experiments described in a paper have been captured in a datum. Nevertheless, it appeared safe to conclude that more research was warranted.

Finding Experiment Elements

The validation we received from the analysis described above motivated us to replace the laboriously created extraction rules of the Evidence Expert with rules learned directly from the data, in the hope of obtaining better scalability, recall, and precision. As detailed in Freitag and Niekrasz, 2016, we developed an approach that automatically produced extraction rules against dependency parses, based on a simple binary separation of sample sentences into those that express a certain kind of information and those that do not.

Table 4 compares these automatically learned patterns with the rules we created for the Evidence Expert. Performance is shown for rules targeting *subject* and *treatment* across three frequent and important assay types, as well as rules that are agnostic to assay type. Each entry lists the precision, recall, and F1 of the corresponding classifier (which is typically a collection of rules or patterns), with the bolded values showing the highest F1 score achieved. The *Learned* column presents the performance of the patterns derived as described above, *Written* the performance of the rules making up the Evidence Expert, and *Baseline* the performance of a simple default rule that says all sentences are a member of the target class.

It will quickly be observed that the learned patterns dominate the hand-written ones in terms of F1 (and almost every other measurement)—despite the fact that the hand-written rules were not intended as a straw man, and were implemented with full access to our experimental corpus (whereas the learned rules were subjected to a strict separation among training, validation, and test sentences).

This approach to extraction is quite general. To show this, we applied the same approach to extracting information from the "BEL corpus," a dataset that pairs sentences pulled from the literature with formal statements representing key information content in those sentences. We were able to exploit this pairing to train rule sets that recognize various abstractions of the formal statements.

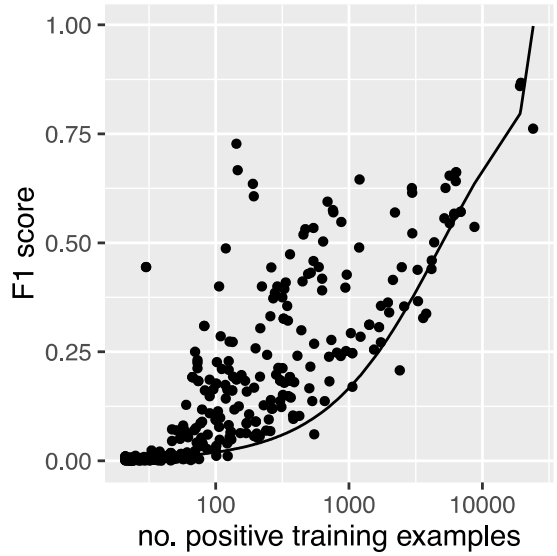

Figure 1: Sentence identification F1 as a function of statement type frequency.

As shown in Figure 1, the rule sets learned by our system are able to outperform, on average, a simple baseline classifier that marks all sentences as an instance of the target class. Here, the class represents some aspect of biological meaning, such as "the sentence expresses a phosphorylation interaction." The strong performance of the baseline with increasing positive training examples is due to dataset exhaustion. Essentially, as we move toward the right, we encounter statements for which it is increasingly true, in the BEL corpus, that every sentence expresses the statement. Each point in the plot represents performance on a different type of extraction problem. The extent to which a point falls above the baseline curve corresponds to the "lift" we experience with the induced rules. We observe this lift on problems with various degrees of representation in the data, but particularly encouraging is the significant lift we see on fairly rare types of information (e.g., on the order of 200 training exemplars). Note also that the correspondence between text samples and BEL statements and the corresponds fact formalizations is very loose. Often, an example comprises a paragraph of text and several BEL statements. Our method successfully localizes the particular expressions that correspond to each type of statement.

Datum Assembly

Due to the difficulties of assembling multi-part extractions, which are illustrated starkly by Table 3, we did not expect the performance numbers we observed in extracting individual frame elements to extend to the extraction of full frames. Note that our metric of performance is the same for both individual elements and frames. Any deviation from the ground truth is recorded as an error. Consequently, frame extraction can never be more accurate than the extraction of a frame's most difficult slot, and each slot that must be populated represents another opportunity to make an error. Errors tend to compound under this strict correctness criterion.

Table 5: Performance of two approaches to datum extraction.

		Frame Classifier			Register Machine		
		Prec.	Recall	F1	Prec.	Recall	F1
phos	+	0.15	0.58	0.23	0.18	0.21	0.19
	-	0.03	0.52	0.05	0.12	0.03	0.04
copptby	+	0.03	0.55	0.06	0.32	0.20	0.25
	-	0.01	0.5	0.03	0.17	0.04	0.07
ivka	+	0.04	0.46	0.07	0.16	0.07	0.10
	-	0.01	0.41	0.02	0.00	0.00	0.00
Summary		**0.05**	**0.54**	**0.09**	**0.16**	**0.14**	**0.15**

Nevertheless, over the course of our efforts in Big Mechanism, we gradually improved on performance levels with single-digit accuracies. As shown in Table 5, varies approximately with the representation of a particular type of frame in the data. Frames with assay type "phos" (representing experiments designed to detect phosphorylation, an important chemical reaction in cellular signaling pathways) with a positive outcome are the most frequent type in our data. Consequently, performance of both our approaches to datum extraction is highest on such frames. However, it is clear that even our best performance is below what would be reasonably be required for general utility.

The comparison between the two approaches yields no clear winner, but interestingly different behavioral profiles. The Frame Classifier is clearly biased in favor of recall, preferring to see many juxtapositions of proteins as underlying experiment frames. The Register Machine is much more selective, and is therefore unable to match the Frame Classifier's recall on any task. This complementarity seems to promise that some simple approach to ensembling the two methods would yield performance superior to either. Such an experiment remains as future work.

However, we did investigate an approach to incorporating the multi-instance learning (MIL) models for predicate detection described in Section 2.2. Consider the problem of determining whether a particular protein mention (e.g., of "ATF-2") should be taken as evidence that it's the subject of a "phos" assay. The MIL approach considers this question at the document level, combining evidence from all mentions of ATF-2, whereas the Frame Classifier makes this judgment based on evidence local to the mention. We showed that by incorporating the MIL assessment as an additional feature when training the Frame Classifier, we were able to boost F1 by 5-10 points over the assay types presented in Table 5.

3.2. AMPLIFYING HUMAN EFFORT

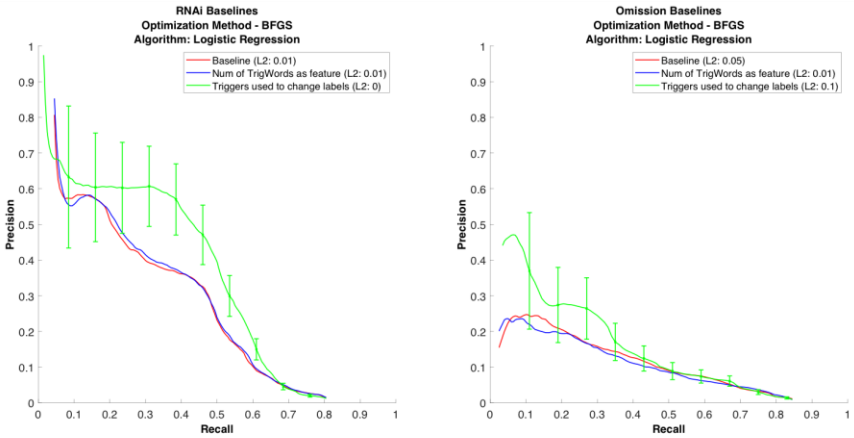

Figure 7. Precision-recall curves for two predicates when trigger words are used as features (blue) or to refine the labeling of instances (green), versus the baseline learner (red).

Above, we presented a simple method for eliciting human review and correction of "trigger" words, words deemed indicative of the presence of particular predicates in a text sample, and derived through statistical association with labels assigned through distant supervision. This procedure is extremely lightweight. The expert reader spends perhaps 5-10 minutes reviewing and selecting the words automatically determined to be indicative of a particular predicate. As shown in Figure 7, when these human-vetted indicators are used to change the training label of sentences, the resulting models recognize predicate mentions with increased precision.

We were similarly successful in our attempts to infer subnetworks associated with specific processes by augmenting existing structured databases with information automatically extracted from the biological literature. We showed that we can use literature-extracted information to (i) augment the set of entities identified as being relevant in a subnetwork inference task, (ii) augment the set of interactions used in the process, and (iii) support targeted browsing of a large inferred subnetwork by identifying entities and interactions that are closely related to concepts of interest. We used this approach to uncover the pathways involved in interactions between the HIV-1 virus and a host cell, and the pathways that are regulated by a transcription factor associated with breast cancer. A paper on this work is in press at *PLoS Computational Biology*.

3.3. HIGH-FIDELITY BIOLOGICAL MODELS

Explaining Drug Response.

We provided a human derived model and explanation for the Dry Run phase of the Phase 3 evaluation. 42 out of 86 measurements with sufficient grounding were explained by the PL model. The model was derived from the existing PL rule knowledge base using what is known about SKMEL133 cells to specify an initial state, and using the Pathway Logic Assistant (PLA) tool, a visualization interface to PL models, to collect the reachable rules. This required some iteration to reformulate some rules to obtain a connected network. This process is currently being automated. Briefly, down regulation of a phosphorylation state was explained by showing that the drug in question blocked the known pathways to that state. Up regulation of protein expression was explained by showing that the degradation pathway was blocked. Down regulation of a modified state was explained by showing that an alternative pathway was blocked, thus likely increasing the flow through the pathway leading to the modification. As a

consequence of the model derivation and analysis two lists of questions to be used for targeted reading by the automated FRIES pipeline were prepared.

For the evaluation second round: 1. We analyzed the Raw Protein Data from the Korkut paper, and came to the conclusion that because the variance across techical replicates was greater than across biological replicates and the number of technical replicates used was not consistent, the raw data didn't give more information than was in the original fold change summary. 2. The human derived Phase III model was revised to meet the latest MITRE specifications. 3. The FRIES reading results were searched by hand to look for information related to the Phase III data that was grounded correctly, and reaction properly typed. One relevant assertion was found, the Phase III model was augmented with a corresponding rule, and analyzed to see what changed (or did not).

We prepared a presentation of the SKMEL133 model and the PL method for explaining RPPA drug response data. This included annotated versions of visual representations produced by the Pathway Logic Assistant. A reduced version was presented at the April, 2017 Big Mechanism PI meeting.

A paper titled "Explaining response to drugs using Pathway Logic" was presented at Computational Methods in Systems Biology 2017 (Talcott & Knapp, 2017). The paper extends the drug explanations with a section on generating hypotheses concerning the action of unknown drugs.

Assembly.

We extensively investigated the use of PL models to explain the “Fallahi dataset,” a set of drug treatment outcomes published by Fallahi et al, and used by Big Mechanism in its evaluation of methods for automated assembly. We explained two of the findings using the human-derived STM model. The explanation included a model of the observed part of the cellular behavior (called the FallahiDish) and subnetworks of this model giving possible mechanistic explanations of the stated findings.

The explanation of Finding 1 (decrease in Rps6 phosphorylation on S235/S236 in 5 cell lines for all drugs) was based the PL STM model of BrafV600E signaling, including the rules involved and a diagram of the subnet involved. For Finding 2 (a decrease in Histone H3 phosphorylation in LOXIMVI cells when treated with AZ628), we hypothesized an explanation based on AZ628 inhibition of VegfR2, using the knowledge that the growth medium contains a VegfR2 ligand, Vegfa. The STM model of the Fallahi data was extended to include VegfR2 rules. This allowed us to explain the decrease in histone phosphorylation, but left other observations unexplained.

The STM derived model was provided to the machine learning team those working on *automated model extension* in our team. We provided performers working on automated machine reading (“readers”) with information needed to fill the gaps in the STM Vegf network. We also provided the readers with search terms for the drugs used in the Fallahi dataset. We checked statements and papers found by the readers for validity and relevance. We also provided some explanation of how to interpret the data taking into account the different seeding and growth conditions, different drug sensitivities, and different methods for determining significant change.

To help define the last evaluation, we provided statistics on the Pathway Logic datum and STM rule knowledge bases:

- number of datums using a given cell line, with/without mutations

- identified datums that have not been used in STM rules
- provided information on putative targets (i.e. what the authors said they were using the chemicals for) for chemicals used as treatments in 431 datums.

During January and February 2017 we hosted Beatriz Santos Buitrago, a Spanish student working on her Masters in Bioinformatics in Seoul So. Korea (Korean schools have a long winter break). For her project, she mapped a data table from Boersema, 2010, onto the PL Egf model. The paper, found by Leora, reports phosphoproteomics response of HELA cells treated with Egf at 10 and 30 minutes.

The mapping process (reading to a model by human) involved 4 steps.

1. The table entries were converted to the PL controlled vocabulary (a) associating Uniprot identifiers to each protein (the table used IPI accession numbers, which are no longer readily available), and looking up the PL name; and (b) converting phosphorylation positions to use the PL numbering system which is based on the Uniprot canonical sequence (splice variant 1) for the protein. This was facilitated by the fact that the table included the peptide containing the site.
2. For each measure phosphoprotein, look for rules in the Egf model that produce it (or a more general form). This required mapping proteins to families in some cases, such as Erk2.
3. For phosphoprotein's with no associated rule, look in the datum kb and the common rules for evidence that could be used to make a rule.
4. Make datums from the table, using a pattern based on the material and methods.

The data set mentioned 54 proteins. 44 appear in the PL Egf model in some form and there are rules for 16 of proteins. 2 more proteins are covered in the common rules, but do not connect to the Egf model. Our knowledge base of datums contained evidence for 9 more rules. A little network was made of the 16 rules and used to check consistency, in that a control for a rule mapping to an increase in the data set that is also in the data set should also increase. The missing rules suggest directions to grow our Egf model as data becomes available.

3.4. AUTOMATED MODEL EXTENSION

We carried out several experiments to validate the approach. These all used the Pathway Logic *Signal Transduction Model 7* (STM7) to supply an initial set of 1581 rules involving 1251 entities, as well as a set of lab-bench experiment descriptors called “datums”. A single Pathway Logic datum typically describes multiple related experiments that demonstrate the dependence of a response on various conditions, so we wrote pre-processing code to break each datum into individual experimental data points. Although Pathway Logic has about 40,000 datums, we restricted attention to a subset of about 450 datums that pertain to pathways involving Hras. These expand to about 2300 individual data points. Each data point asserts that if a simulation is started with a particular initial condition, then it will reach a final state that does or does not contain particular entities.

In early experiments, we demonstrated that the machine learning principles we wished to employ can be applied as intended. The likelihood function is essentially the number of experiments “proved”; i.e., explained by a simulation generated by the inference engine. We experimented with various priors, starting with a simple expedient based on counting the number of rules. To

account for the varying complexity of different individual rules, we generalized this prior to depend on the total number of antecedents and consequents. We found that results did not depend very strongly on the details of the prior. As long as it mildly penalizes rules in general, redundant or irrelevant rules that do not contribute to explaining experiments get weeded out.

The simplest type of experiment to do is greedy iterative rule deletion. We demonstrated that this is an effective way to post-process rules generated automatically from datums using an Answer Set Programming (ASP) algorithm. The ASP algorithm tends to over-generate. We carried out an experiment in which a set of 16 related rules related to Hras activation were deleted from STM7, the ASP algorithm was applied to auto-generate rules from the datums supporting the deleted rules, these auto-generated rules were added back to STM7, and greedy iterative rule deletion was applied until a local minimum in the regularized likelihood was reached. Our biologists judged the results to be broadly sensible. We were also able to control the severity of rule trimming by varying a parameter controlling the strength of the prior.

We also verified that placing priors on data points works as expected. The simulations with STM7 were normally run from an initial state containing 127 of its 1251 entities. By assigning high confidence to data points that say other entities are produced eventually, we found that rule sets score more highly if they result in production of more final products.

One of our earliest experiments involving input from machine reading used output from the University of Arizona's REACH system. REACH extracted entities and relations (automatically) that were translated (manually) into 50 Pathway Logic entities and 35 rules so that they could be augmented onto STM7. Two things were done with the augmented rules: (a) they were assigned prior probabilities; and (b) the posterior probabilities of the augmented models created using subsets of the extracted rules were computed. The priors (a) were obtained by checking whether they "looked like" rules already in the heavily curated STM7. To define "looked like", the entities in the rules of STM7 were backed off to more generic ontological categories; e.g., proteinName-modification@location became proteinGeneric-modification@locationAny. Statistics were gathered from STM7 at this backed-off level, the extracted rules were similarly backed off and matched to the backed-off STM7 rules, and were then assigned probabilities based on the number of occurrences in STM7. This gave results that roughly agreed with curator opinion. The first step in computing the theory posteriors (b) was to make an extended STM7 model by inserting the 35 extracted rules. We then checked how this impacted proofs of plausible data that could be made with the augmented model. Variants of the augmented model were generated by deleting augmented rules one by one, and these were evaluated with a complexity prior that favored smaller sets of rules. The outcome was somewhat disappointing: The rules imported from machine reading did not enable any more data points to be explained than could be explained with STM7 alone. This was because, of the 50 entities in the augmented rules, only 5 were in STM7 and these did not impact any of the relevant inference chains. So while successful as a demonstration of ingestion of input from machine reading, the main lessons were that more effort needs to be applied to ontological alignment between data sources, and that it is not trivial to improve on an already highly-curated knowledgebase.

In our first significant exercise with the re-organized sense-making software, we combined the 1581 rules from the Pathway Logic STM7 knowledge base with 5159 rules from the UA REACH reader and 322 rules from the Leidos table reader. We also added a few manually modified Pathway Logic rules. We then applied the sense-making layer to score the ability of the combined rules to explain 339 datums from Pathway Logic that pertain to Hras. Many

datums describe multiple experiments, some descriptions have multiple interpretations, and some interpretations can be mapped onto the simulation state space in multiple ways, so the 339 datums expand into 548 experiment descriptions with 1414 interpretations, 705 of which could be mapped onto the simulation state space forming 2325 formalized experimental data points. The maximum possible score (experiments explained) was therefore 2325, and the actual score was 1146.

We evaluated the importance of every (interpreted) rule by deleting one rule in all possible ways and re-computing the score. Only 29 rules had any impact, ranging from -204 for deleting the rule that binds Egf to its receptor, the initial step of a major pathway, to +4 for one of the manually altered rules. All but one of the impactful rules came from Pathway Logic, the remaining rule coming from REACH. However, the REACH rule was deemed to be a misreading of the text that luckily contributed to some explanations, but in an implausible way. At this point, reading was not improving the curated model.

We then took another approach, asking whether reader output could be used to recover knowledge ablated from an STM7. We used a collection of 1320 of the 1581 STM7 signaling rules, 3831 relevant datums (each of which describes one or more related experiments), and UAZ REACH extractions from papers indicated as supporting the rules. The difficulties with the 261 omitted rules were mostly due to PDF extraction problems. The unaltered knowledgebase of 1320 rules explains 846 experimental results. Each rule was deleted, one at a time, noting the change in the number of experiments explained. For 1246 of the rules, there was no effect. There were 63 rules whose ablation resulted in a loss of roughly 70 explained experiments, and 11 that resulted, unexpectedly, in a *gain* of about 100 explained experiments. After ablating a rule, new rules were assembled from extracts read from the relevant papers and added to the knowledgebase, and the change in the number of experiments explained was noted again. This almost always had a substantial and seemingly beneficial effect. There were only 78 rules for which reading produced no improvement after ablation, 58 for which reading resulted in a loss of roughly 25 explained experiments, but 1184 that resulted in a gain of around 350 explained experiments. This appears to show that reading produces great value.

However, a deeper assessment found that most of the beneficial reader rules went directly from the starting condition to the outcome conditions for the data points, i.e., essentially stated a direct connection between the treatment and subject of an experiment, and therefore provided little explanatory value. We carried out a *hitting time analysis*, in which we kept track of the number of time steps between the initial setup of a simulation and its production of the result predicted by the experiment description. With the unperturbed knowledgebase and the ablated knowledgebases, this almost always takes at least 2 time steps, but with rules added by reading, 1 time step often becomes sufficient. This corroborates the thesis that the read rules fire too readily.

Inspecting some of the rules led to a fairly clear explanation of this behavior. Rules obtained from reading tend to be much less detailed than the ablated rules they are hoped to recover, lacking location information and many structural details about the interacting entities. The omission of details makes the antecedent conditions of the rules easier to meet, so they are more likely to fire, resulting in the creation of more products than are produced in reality. With a more balanced suite of experiment descriptions, rich in control experiments providing evidence ***against*** the creation of most of these products, the overly permissive rules would contradict

many of the controls, more than compensating for their simplistic and most likely spurious explanation of extra positive results.

This led us to develop the combinatorial approach to entity and rule assembly described above, generating all possible rules from the entities reported by a reader, filtering them through a common sense evaluation module, and checking agreement with data. The experimental data was extracted manually by our curator from the papers used in a Mitre evaluation, following her routine methodology. They should eventually be extracted automatically by our Evidence Expert. The data set was small, but the results were very encouraging, with recall improving from 35% to 48%, and when restricted to the interactions we have had time to support, 65%.

We first tested the closed loop reading using the previously existing platform. This procedure started with the pathway logic model and attempted to extend the model using closed loop reading to explain the results observed in Korkut et al, 2015. This data looked at drug treatment and corresponding measured changes in phosphorylation, across multiple drugs and assays in a single cell line. The initial human curated pathway logic system could explain 3/66 of the rules out of the box (using manual definition of the initial stat and rule customization to the cell line). The automated system processed 22k citations and proposed 7k rules, of which 330 passed the common sense filter, and 8 were ultimately accepted by the system. The 8 new rules allowed the system to explain 44/66 of the observed experimental data points. Note that this did not use the subsequently developed MCMC system so that only parsimony was used to control rule growth, rather than the more effective measures introduced later.

The system was extended to apply to multiple cell line for the phase 3 evaluation Fallahi dataset. For the phase 3 evaluation we implemented the following procedure:

1. Extract experimental data, and convert to probability of change
2. Attempt automated extraction of drug targets (not integrated in further processing yet)
3. Automatically work backwards from measurements to determine rules relevant to producing observed outcomes (backwards collection)
4. Create candidates for the initial dish based on required proteins for rules identified in step 3
5. Construct all paths through rules from the initial dish that can produce a measured outcome
6. Identify paths that intersect drug targets, as paths relevant for predicting the observed experimental results
7. Create 3 matrices needed to simulate the experiments:
    a. Pathway matrix indicating rules and initial proteins needed for each path
    b. Measurement matrix indicating which experiments are predicted by each path
    c. Score matrix corresponding to probabilistic score for each measurement base on the observed experimental results

8. Using these three matrices we rapidly simulate many initial states and rulesets for each cell line, currently selecting the networks that maximize agreement with experiment

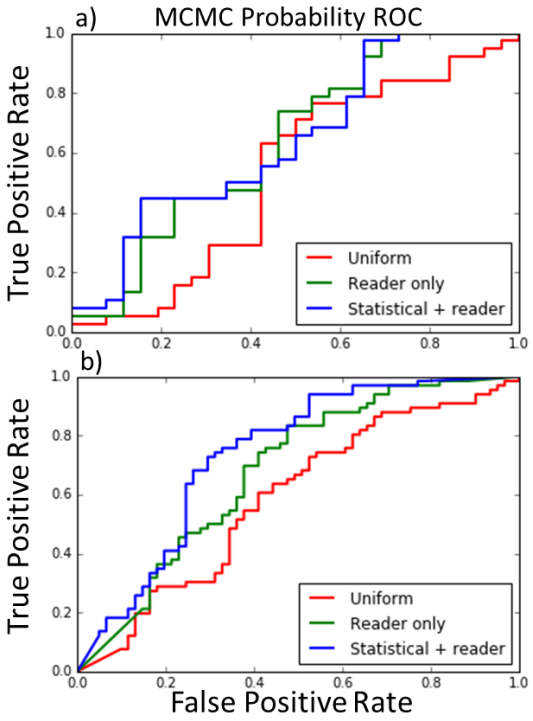

Figure 8. Receive Operating Characteristic (ROC) for automatically generated rules against expert human curation. a) trained using only the Hek293 data b) trained using data from 4 independent cell lines (mEFs, Hek293, Hek293T, HELA). Different colors indicate use of different prior information in selecting rules.

9. Use the optimal networks to explain or refute the observed findings

After the phase 3 evaluation we switched to the MCMC approach for multi-cell line simultaneous processing on the general pathway logic rules. This updated system was run against 4 cell lines from the Pathway logic database (mEFs, Hek293, HELA, Hek293T) with the largest number of data points. We compared the predicted results to post-hoc biologist evaluations of all proposed new rules. The results are shown in Figure 3.

In the figure we see that use of multiple cell lines increases the ability of the system to distinguish which rules are most useful for explanation, based on expert annotation. Additionally, we see that inclusion of reader information from the closed-loop reader process substantially enhances performance, and further inclusion of statistical common sense information improves performance yet again.

The system attempted explained 48 previously unexplained experiments, using 77 new rules. Of these new rules 48 were evaluated as correct by an expert, while 19 were evaluated as incorrect.

We explored the application of automated model extension to Gray Zone problems. Specifically, we attempted to automatically create a ruleset that applied to the Ukraine Crisis over the past eight years. We started by curating potential rules from the relevant literature including contemporary news articles, retrospective reports, and transcripts of round tables discussing the crisis. A typical example of a rule generating statement is “Without meaningful Western military aid, Ukrainian President Petro Poroshenko acceded to unreasonable demands from Russian President Vladimir Putin.”

A statement would be manually decomposed into three rule components: context, control and assay. This structure parallels the structure used in PL biological rules, where the “consumed”, “control” and “produced” protein occurrences played similar roles. For the above text extract, we identified the rule:

- Context: Demand by aggressor
- Control: Military aid provided by USA
- Assay: Reduced probability of acceding to demands

We explored matching each rule to data from the Global Database of Events, Language and Tone (GDELT). We downloaded all GDELT events from GDELT 1.0 from 2013 until the present. We

selected events with at least one relevant location in the Ukraine. Each article for the identified events was downloaded and parsed to extract relevant keywords. Each event was then represented by the GDELT event type as well as the set of keywords extracted from the article.

We then created a neural network representation of each rule. The network was built using SRI's Deep Adaptive Semantic Logic (DASL) python package. Events were binned into two-week slices, and each component of a rule (Context, Control, Assay) was implemented as a neural network (multi-layer perceptron). The components were temporally tied together with Context and Control being measured at each time step, and the Assay being measured for the subsequent time step. We vectorized the event data to form the network input.

Both GDELT event types and keywords were placed within a vector embedding. For the GDELT event types we defined a similarity score between events types based on the CAMEO code hierarchy, along with the overall four-bin CAMEO code. A singular value decomposition was then used to create 30 dimensional vectors that accurately reproduced the distance metric (30 selected based on a principal component analysis). The pre-trained (300 dimensional) word2vec word embedding, trained on the google news corpus, was used to encode keywords as vectors.

Each two-week interval was represented as a bag of events, with each event represented as a paired Cameo event code vector and key word vector. The full set of events was sub-selected to those relevant to each rule component (Context, Control or Assay), based on distance from the event code and keyword manually identified for each rule component. For example, the above rule for the context "demand by aggressor" used the CAMEO code for "Demand" along with the keyword "russia" (all keywords were lowercase to align with word2vec standards). After sub-selection DASL was used to construct a theory that tested for the presence of the purported event in each time interval.

Each multi-layer perceptron was trained using stochastic gradient descent to identify rule components that maximized a rule's predictive accuracy based on a training set consisting of the first few years of the Ukraine conflict. The rules were then tested on more recent data (last 2 years). The metric chosen was the number of predicted counter-factual events, i.e., the number of observed time periods for which the rule correctly predicts the outcome when the control is true versus minus the expected number if the control was false. For example, in the above this corresponds to the number of additional times that the Ukraine acceded to Russian demands when the United States provided aid versus a baseline when no aid was provided, conditioned on a Russian demand being made. This metric provides an operationally sound way of comparing probabilistic rules, as it puts highly predictive rules that trigger with low probability on the same footing as less accurate rules that can be applied often.

The result is a system that can rapidly customize rules to specific situations. We manually extracted 50 candidate rules from the literature, of which 29 could be effectively implemented. Of these 6 were detected as significant and consistent across time. The verified rules (with the predicted probability of the rule influencing the assayed event) are:

- 16% Threats to escalate conflict decrease adversary activity
- 11% Building local support decreases future conflict
- 13% Support for border infrastructure decreases adversary activity
- 7% Support for accountability decreases adversary activity

- 8% If aggressor sharply expands aggression during a ceasefire sanctions will not be increased
- 5% Threats to use force during negotiations make them more likely to succeed

Overall the system performed at about the level we expect from our previous Big Mechanism results, as most of the extracted rules tend not to be correct. The main thing is that automated reading allowed us to test thousands of rule candidates versus the few we obtained from purely manual extraction for the Gray Zone study.

The largest challenge was the limited amount of data available in this study to verify each rule. In trying to interpret the rules, the system was clearly over-training, as the training set would predict a much higher number of counterfactual examples than were observed in the test set. We used regularization to limit the overtraining but the ultimate issue was the limited amount of training data obtained by focusing on a single conflict. The best methods to address this would be to train rules across multiple conflicts and to break conflicts down into smaller pieces (e.g. by geographical area / theater) allowing more precise training over more examples.

Several of the rules looked like their behavior was inconsistent over time. Specifically, the rule "Ending politicization of multi-national trade groups boosts trade" was true in the initial part of the conflict, but later was incorrect (i.e. the rule reversed). This likely has to do with unmodeled temporal dynamics related to oil production and sanctions.

Assembling a complex mechanism in the same way as Big Mechanism biological data was infeasible due to limited rule coverage and lack of sufficient data to test the long causal chains. The most critical piece needed to assemble an effective mechanism is to have the rules overlap such that the Control component of rule A connects to Assay component of rule B. At the purely textual level this connection works, as many rules mention the same entities (trade, force, demands, etc.). Unfortunately, the neural network customization gives differing interpretations to these rules, in that the specifically detected events differ due to the different neural network representations. DASL allows us to constrain the interpretations to overlap; however, with limited data we were unable to construct a sufficiently complicated connected ruleset due to the poor statistical evidence and overtraining issues.

4. TECHNICAL FEASIBILITY

Targeted machine reading for factual information, by which we mean the extraction of entities or relations fulfilling specific roles (e.g., the subject of an assay), is indeed feasible in some cases, but certain forms of the problem remain beyond the practical reach of the state of the art. It is possible to detect such roles based on the evidence in a single sentence, and we developed new methods to detect relevant mentions based on noisy distant supervision, methods capable of localizing the particular sub-sentence expressions used to express a particular role.

The feasibility of this problem decreases in proportion to two complicating factors. First, as the arity of a target relation increases, the opportunities for "getting it wrong" by incorrectly populating a "slot" also increase. In our experiments, we sought to populate 4-tuples, and were only beginning to see marginally practical accuracies by the end of our effort. The difficulty of this task was increased by the presence of the second complicating factor: noise in the training data. In contrast with much of the work on information extraction, our annotations were at the

sentence level and we arrived at them through a heuristic alignment between figure references in datum records and sentences. There are several points at which this alignment can fail, and any failure yields an incorrect label, with corresponding degradation in the resulting models.

In a sense, this is the problem that our attempts to *amplify human effort* are intended to address. We derived ample evidence that modest human intervention increases the accuracy of models subject to noisy training data. The trick is in how information is presented to human experts and how feedback is captured. We showed, for example, that just having the expert filter a set of words automatically determined to correlate with predicate mentions—a task that typically takes only minutes to perform—yields statistically significant improvements in detection models. Our efforts to provide for finer-grained control by the expert, through the highlighting of particular phrases in the input, remain somewhat more speculative. It is important that the expert not be reduced to reading large sections of text, as this impedes scaling and is not consistent with the goals of automated information extraction. We continue to believe that an efficient communication of estimated salience, e.g., through the automated highlighting of key phrases by the system, is critical. Moreover, we believe that it would be promising to explore user interfaces that enable the expert to provide multiple forms of feedback in a given session, such as adding trigger words, highlighting key passages, and indicating links between predicates that pertain to the same frame.

We established that our basic concept for *automated model extension* is workable. We can use a layer of explanatory reasoning to merge machine learning output into a knowledgebase in a sensible way. We can blend fragmentary rules from readers or automated rule-generating algorithms into well-formed hypothetical rules and evaluate those rules in the context of existing knowledge.

We have demonstrated the utility of our closed loop reading and MCMC approaches in supporting automated extension of an existing biological cell signaling model (Pathway Logic). We have demonstrated the ability to combine information extracted from automated reading with machine learning for statistical common sense and experimental verification using an MCMC search procedure.

We have tested the applicability of our model extension system to a novel gray zone context, where experimental evidence is replaced by evidence curated from historical examples (quasi-experiments). In this case the rules are less precise, requiring context-specific interpretation (e.g. a rule that states “Trade decreased” requires interpretation in the specific context of the local conflict to identify the party, and to detect events indicative of trade decrease). We have demonstrated the ability of neural networks to perform this interpretation automatically by optimizing a rule quality function over automated text extracts curated from GDELT. The key challenge is collecting a wide enough set of data to allow confident rule interpretation without overtraining. For our demonstration we manually extracted the rules rather than use machine reading, meaning that a human was required to identify the passages that corresponded to potential rules, and to extract the key components of the potential rules (context, control, assay). The rule structure paralleled the biological rules in our Big Mechanism work, implying that we can apply similar techniques for automated reading to extract potential rules from text, assuming one were to customize the Big Mechanism machine readers to the gray zone context. Automated model assembly proved more challenging in the Gray Zone context than in the biological context as a consistent grounding similar to biological protein occurrences was not available, so rules were even more challenging to connect together. Additionally, the primary evidence for Gray

Zone data was directly applicable to single rules, while the probabilistic nature of the data made testing a rule chain impractical (and statistically unsupportable). Together these effects meant that the overall rule compilation engine was under constrained. Overall the clear lesson is that we need to simultaneously interpret rules across a large number of Gray Zone conflicts to both get a large enough training dataset and to ensure that rules will generalize across locations and times. We did not produce sufficient evidence to indicate that construction of large mechanisms that explain complex chains of events is possible due to the rarity of observing complex chains (versus in the biological context where single experiments often observe such chains).

We made progress in inferring mechanistic cell signaling rules directly from datums. We were able to replicate the human-curated Hras network. A key challenge is assembling rules into executable PL models where rule outputs exactly match inputs to rules for subsequent steps. Inferred rules capture the level of detail provided by experimental assays, which differ in specificity. The matching problem was solved by adding less precise instances of each rule. This does not scale well. In an ongoing collaboration, we are exploring fuzzy matching to solve the scaling problem and extending the reasoning tools to a "soft logic."

PL executable models succeeded in providing mechanistic explanations of observations from experimental data where cells were treated with drugs inhibiting steps in different signaling pathways. We also succeeded in one exercise in extending a model with information requested from readers to fill a gap in the model. The extended model was then able to explain additional observations. Although some human effort is needed to bridge the reading-to-model gap, this indicates promising potential synergy between human and automated curation. The explanatory potential of PL knowledge bases and models is more general than explaining the effects of drugs. We have been extending the scope to modeling host pathogen interactions and protective immune response. One application is using the resulting models to identify novel pathogen attack points/surfaces.

5. BIBLIOGRAPHY

P. J. Boersema, L. Y. Foong, V. M. Y. Ding, S. Lemeer, B. van Breukelen, R. Philp, J. Boekhorst, B. Snel, J. den Hertog, A. B. H. Choo, and A. J. R. Heck, "In-depth qualitative and quantitative profiling of tyrosine phosphorylation using a combination of phosphopeptide immunoaffinity purification and stable isotope dimethyl labeling," Molecular & cellular proteomics : MCP, vol. 9, no. 1, 2010. 19770167.

Deborah Chasman, Yi-Hsuan Ho, David B Berry, Corey M Nemec, Matthew E MacGilvray, James Hose, Anna E Merrill, M Violet Lee, Jessica L Will, Joshua J Coon, Aseem Z Ansari, Mark Craven, Audrey P Gasch. 2014. "Pathway connectivity and signaling coordination in the yeast stress-activated signaling network." Molecular Systems Biology 10:759.

Dayne Freitag and John Niekrasz. 2016. "Feature derivation for exploitation of distant annotation via pattern induction against dependency parses." Proceedings of the 15th Workshop on Biomedical Natural Language Processing.

Dayne Freitag, Paul Kalmar, and Eric Yeh. 2017. “Discourse-wide Extraction of Frames Representing Biological Assays.” Workshop on Biomedical Natural Language Processing.

Sid Kiblawi, Deborah Chasman, Amanda Henning, Eunju Park, Hoifung Poon, Michael Gould, Paul Ahlquist, and Mark Craven. 2019. “Augmenting Subnetwork Inference with Information Extracted from the Scientific Literature.” *PLoS Computational Biology*, in press.

Korkut A, Wang W, Demir E, Aksoy BA, Jing X, Molinelli EJ, Babur O, Bemis DL, Onur Sumer S, Solit DB, Pratilas CA, Sander C. Perturbation biology nominates upstream-downstream drug combinations in RAF inhibitor resistant melanoma cells. Elife. 2015 Aug 18;4. PMID:26284497.

Vivek Nigam, Robin Donaldson, Merrill Knapp, Tim McCarthy, and Carolyn Talcott. 2015. "Inferring executable models from formalized experimental evidence." In Roux, Olivier, and Jérémie Bourdon, eds. *Computational Methods in Systems Biology: 13th International Conference, CMSB 2015, Nantes, France, September 16-18, 2015, Proceedings*. Vol. 9308. Springer, 2015.

Gustavo Santos-Garcia, Carolyn Talcott, Adrian Riesco, Beatriz Santos-Buitrago, and Javiar de las Rivas. 2016. “Role of nerve growth factor signaling in cancer cell proliferation and survival using a reachability analysis approach.” Proceedings of the 10th International Conference on Practical Applications of Computational Biology and Bioinformatics.

Carolyn Talcott. 2016. “The Pathway Logic Formal Modeling System: Diverse views of a formal representation of signal transduction.” In the Workshop on Formal Methods in BioInformatics and BioMedicine (invited paper), December 18, 2016.

Carolyn Talcott. 2018. "Pathway Logic: Symbolic Executable Models for Reasoning about Cellular Processes" Invited talk at the Logic for Systems Biology workshop at FLoC 2018 in Oxford, UK. The slides are available at pl.csl.sri.com/publications.html

Carolyn Talcott and Merrill Knapp. 2017. "Explaining response to drugs using Pathway Logic" Computational Methods in Systems Biology.