REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
22-10-2019	Final Report	18-Jan-2017 - 17-Sep-2019

4. TITLE AND SUBTITLE
Final Report: KaZam: An integrated inference engine for assembly

5a. CONTRACT NUMBER
W911NF-17-1-0073

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Carnegie Mellon University
5000 Forbes Avenue

Pittsburgh, PA 15213 -3589

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

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

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
70837-NS-DRP.3

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

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

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Qinsi Wang
UU	UU	UU	UU		19b. TELEPHONE NUMBER 000-000-0000

RPPR Final Report
as of 06-Nov-2019

Agency Code:

Proposal Number: 70837NSDRP **Agreement Number: W911NF-17-1-0073**
INVESTIGATOR(S):

Name: Qinsi Wang
Email: qinsiw@cs.cmu.edu
Phone Number: 0000000000
Principal: Y

Organization: **Carnegie Mellon University**
Address: 5000 Forbes Avenue, Pittsburgh, PA 152133589
Country: USA
DUNS Number: 052184116 EIN: 250969449
Report Date: 17-Sep-2019 Date Received: 22-Oct-2019
Final Report for Period Beginning 18-Jan-2017 and Ending 17-Sep-2019
Title: KaZam: An integrated inference engine for assembly
Begin Performance Period: 18-Jan-2017 **End Performance Period:** 17-Sep-2019
Report Term: 0-Other
Submitted By: Qinsi Wang Email: qinsiw@cs.cmu.edu
Phone: (000) 000-0000

STEM Degrees: 1 **STEM Participants:** 2

Major Goals: Cancer signaling is an example of a complicated system where interactions have important causal effects. Creating mechanistic models, rather than correlative models, helps with understanding such systems. The goal of the Big Mechanism is to push forward tool development to support 1) the automatic generation of these models from natural language and 2) the analysis of these models to improve understanding. The Big Mechanism project has four phases - reading, assembly, modeling, and analysis, and our efforts have been put into the later three aspects in the following ways.

Task 1: Reasoning about Assembly
An important hurdle for the BM project involves assembly, being able to assemble coherent models from natural language output. How to integrate semantic ontological reasoning about natural language structures with the underlying biochemistry is a key problem for this project.

Task 2: Reasoning about Causality
The causal analysis is another important problem involved in the BM project. The aim of causal analysis is to understand better how pathways emerge from a multitude of potential low level protein-protein interactions. A great number of these interactions is known thanks to the progresses of experimental methods but the way they result in complex signaling behaviors remains to be understood. One difficulty in figuring this out comes from the density of the reaction network involved along with its apparent lack of structure. In particular, there are many cross-talks between signaling processes that seem completely unrelated. As a consequence, small signaling models that focus on a limited number of interactions in order to stay tractable for human reasoning usually bring a limited insight. Thus, the development of automated analysis techniques for large models of protein-protein interaction networks appears to be an important step towards a better understanding of signaling pathways.

Task 3: Querying about Rule-based Models
With a given Kappa model, how can one explore a question like “which complexes contribute to kinetics”? Experimental biologists have been using labeling techniques for decades, but implementing this in a modeling framework requires being able to track individual agents, and query particular events.

Accomplishments: 1 Syndra - a Reasoning Tool for Assembly
Rule-based biological models help researchers investigate systems such as cellular signaling pathways. Although these models are generally programmed by hand, some research efforts aim to program them automatically using biological facts extracted from papers via natural language processing. However, NLP facts cannot always be directly converted into mechanistic reaction rules for a rule-based model. Thus, there is a need for tools that can

convert biological facts into mechanistic rules in a logically sound way. We have constructed such a tool specifically for Kappa - Syndra, by implementing Iota, a logic language over Kappa programs. Our tool can translate biological facts into Iota predicates, check predicates for satisfiability, and find models that satisfy predicates. We have tested our system against realistic use cases, and shown that it can construct rule-based mechanistic models that are sound with respect to the semantics of the biological facts from which they were constructed. In the following, we will offer more details of the problem and our solution.

The Problem: Turning Facts into Rules
The problem is that NLP output can be messy: it contains mechanistic rules, non-mechanistic rules, and domain knowledge, all of which must somehow be woven together in order to create an accurate model composed only of mechanistic rules.

The Solution: Logical Inference over Chemical Semantics
As a solution, we have created a tool that uses logical inference to deduce a set of clear-cut mechanistic rules that are consistent with the messier input facts NLP gives us. A good computational model should contain all available facts pertaining to the system it models, including facts that can’t be immediately converted into the rules of a rule-based model. Our solution allows us to produce models that incorporate all of this available knowledge, by performing logical inference in order to deduce which mechanistic rules are implied by the available facts.

2 Reasoning about Causality for Rule-Based Models
For this task, we study the problem of analyzing the causal structure of rule-based models. More concretely, suppose we are given a fine-grained mechanistic model of a biochemical interactions network, expressed in a language such as Kappa. In Kappa, chemical interactions between proteins are modeled as stochastic graph-rewriting rules. Given a model (a set of such rules) and a phenomenon of interest, it is possible to simulate the model and see whether or not the phenomenon of interest happens and under which conditions. However, execution traces alone (which can be extremely large) do not offer an easy way into understanding why and how the phenomenon of interest emerges from atomic interactions. The aim of causal analysis is to extract causal diagrams from simulation traces, which explain this emergence in terms of primitive causal interactions between atomic events (such as activation and inhibition). These causal diagrams can be seen as formal counterparts of ``signaling pathways''.

* Counterfactual Resimulation for Causal Analysis of Rule-Based Models
The previous work on this topic suffered from a significant limitation: it could only handle activation arrows, whereas inhibition plays a very important role in many biological phenomena. Our main contribution was to extend this framework to produce causal diagrams involving inhibitory influences. This is challenging because reasoning about inhibition requires reasoning about events that did not happen during simulation. Our first contribution in this work was to propose ``counterfactual reasoning'' as a foundational framework to define and reason about inhibitory influences.

Given an observed simulation trace, we looked into counterfactual statements of the kind: ``would event B had happened had event A not happened''. Giving a semantics to such statements is tricky because of the stochastic nature of rule-based models. It is not simply possible to answer such a statement by resimulating the system after blocking event A.

Intuitively, every random contingency that is not remotely affected by A must be replayed the same. In our work, we gave a semantics to counterfactual statements in the setting of rule- based models by generalizing Pearl’s seminal definition of counterfactuals. Then, we proposed an efficient algorithm to evaluate such statements, which we implemented into a signaling pathway.

* KaStat
We have also developed a statistical analyzer KaStat for Kappa models. Currently, it carries out two types of analyses: estimating the probability of a certain type of influence between two given Kappa rules with the help of statistical testing methods, and outputting the causal core that contributes most to the existence of the influence between the given Kappa rules by generating causal cores and carrying out the isomorphism checking.

With KaStat, we can 1) return a finer-grained causal core explaining the underlying causality between two specified events, 2) detect important motifs in the graphical representation of the given rule-based model, 3) remove redundancy existing in the given models, and 4) uncover evolutionary behaviors.

3 Trace Querying Language for Rule-based Models
Rule-based modeling languages such as Kappa and BioNetGen can be used to write mechanistic models of complex reaction systems. Models in these languages consist of stochastic graph-rewriting rules that are equipped with rate constants indicating their propensity to apply. Together with an initial mixture graph, these rules constitute a dynamical system that can be simulated using Gillespie's algorithm. Each run of simulation results in a sequence of transitions that we call a trace.

In practice, simulation traces are often discarded in favor of a limited number of global features, such as the concentration curves of a set of observables. However, a more detailed analysis of their structure and statistical properties can provide useful insights into a system's dynamics. For example, causal analysis methods exist that compress a large trace into a minimal subset of events that are necessary and jointly sufficient to replicate an outcome of interest, and then highlight causal influences between those remaining events (see work on counterfactual resimulation). Queries about the statistical behavior of individual agents can lead to complementary insights. Examples include 1) measuring the average lifespan of a complex under different conditions, 2) computing a probability distribution over the states in which a particular type of agent can be when targeted by a given rule, and 3) estimating how much of a certain kind of substrate getting phosphorylated is due to a particular pathway at different points in time.

In this work, we proposed and implemented a unifying language to express queries of this kind, that are concerned with statistical features of groups of molecular events that are related in specific motifs. Out tool is actively maintained and has already been used in a study of the WNT signaling pathway, for which one of the largest rule-based models in existence has been built.

Training Opportunities: Nothing to Report

Results Dissemination: Through 1) publishing papers, 2) presenting developed methods and tools in conferences and workshops, 3) show the functionalities of our tools to people who are potential users.

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Qinsi Wang
Person Months Worked: 15.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co PD/PI
Participant: Jean Yang
Person Months Worked: 15.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

RPPR Final Report
as of 06-Nov-2019

CONFERENCE PAPERS:

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: Computational Methods in Systems Biology
Date Received: 16-Oct-2018 Conference Date: 12-Sep-2018 Date Published: 12-Sep-2018
Conference Location: Brno, Czech Republic
Paper Title: A Trace Query Language for Rule-Based Models
Authors: Jonathan Laurent, Hector F. Medina-Abarca, Pierre Boutillier, Jean Yang, Walter Fontana
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: Twenty-Seventh International Joint Conference on Artificial Intelligence {IJCAI-18}
Date Received: 28-Oct-2019 Conference Date: 13-Jul-2018 Date Published: 13-Jul-2018
Conference Location: Stockholm, Sweden
Paper Title: Counterfactual Resimulation for Causal Analysis of Rule-Based Models
Authors: Jonathan Laurent, Jean Yang, Walter Fontana
Acknowledged Federal Support: **Y**

Big Mechanism, DARPA Project
Final Report from CMU

Qinsi Wang, Jean Yang, Jonathan Laurent
Computer Science Department
Carnegie Mellon University

1 Introduction

Cancer signaling is an example of a complicated system where interactions have important causal effects. Creating mechanistic models, rather than correlative models, helps with understanding such systems. The goal of the Big Mechanism is to push forward tool development to support 1) the automatic generation of these models from natural language and 2) the analysis of these models to improve understanding. The Big Mechanism project has four phases - reading, assembly, modeling, and analysis, and our efforts have been put into the later three aspects in the following ways.

Task 1: Reasoning about Assembly (Section 3)

An important hurdle for the BM project involves assembly, being able to assemble coherent models from natural language output. How to integrate semantic ontological reasoning about natural language structures with the underlying biochemistry is a key problem for this project. Solving this problem includes reasoning about biochemical properties of proteins and tacit domain-specific assumptions, all of which are important for navigating the space of possible models implied by the literature and databases. The key enabler of this research is the fact that we are expressing models of cellular signaling using a rule-based modeling formalism, rather than as systems of differential equations. The compactness of the rule-based models, as well as the fact that the models have a precise operational semantics that can transparently represent mechanism, allows us to describe these models in a logical form conducive to adapting techniques that have been developed for analyzing programs. We have developed an integrated inference engine *Syndra* that addresses this problem for the rule-based models being able to carry out the assembly with guarantees about the correctness and coherence of the models. The reasoning engine integrates semantic, ontological reasoning with reasoning about the mechanics of the underlying chemistry.

Task 2: Reasoning about Causality (Sections 4)

The causal analysis is another important problem involved in the BM project. The aim of causal analysis is to understand better how pathways emerge from a multitude of potential low level protein-protein interactions. A great number of these interactions is known thanks to the progresses of experimental methods but the way they result in complex signaling behaviors remains to be understood. One difficulty in figuring this out comes from the density of the reaction network involved along with its apparent lack of structure. In particular, there

are many cross-talks between signaling processes that seem completely unrelated. As a consequence, small signaling models that focus on a limited number of interactions in order to stay tractable for human reasoning usually bring a limited insight. Thus, the development of automated analysis techniques for large models of protein-protein interaction networks appears to be an important step towards a better understanding of signaling pathways.

For the Kappa rule-based models that have been studied, we have developed an approach that complements the existing causal analysis of event series generated from rule-based models by using counterfactual reasoning to answer questions of the kind: Had event e1 not occurred, would event e2 have happened? In detail, we have provided a semantics for counterfactual statements in the context of rule-based models, where the standard definition of counterfactuals based on structural equations [1, 2] does not apply. We have shown how such statements can be evaluated by sampling counterfactual traces that are meant to probabilistically ``hug'' a given (factual) trace as much as an external intervention permits them to. To this end, we have developed an algorithm to generate counterfactual traces and pro- vide an efficient implementation for the Kappa language. We have also shown how counterfactual dependencies between events can be systematically explained in terms of enablement and prevention relations that are more in line with biological reasoning.

We have also developed a statistical analyzer *KaStat* for Kappa models. Currently, it carries out two types of analyses: estimating the probability of a certain type of influence between two given Kappa rules, and outputting the causal core that contributes most to the existence of the influence between the given Kappa rules. With KaStat, we can 1) return a finer-grained causal core explaining the underlying causality between two specified events, 2) detect important motifs in the graphical representation of the given rule-based model, 3) remove redundancy existing in the given models, and 4) uncover evolutionary behaviors.

Task 3: Querying about Rule-based Models (Section 5)

With a given Kappa model, how can one explore a question like “which complexes contribute to kinetics”? Experimental biologists have been using labeling techniques for decades, but implementing this in a modeling framework requires being able to track individual agents, and query particular events. Implementing a framework to query events on the trace of an agent and rule simulation seems a natural way of tackling these classes of problems. Moreover, once a sufficiently rich mechanistic model is available, questions on mechanism arise. For a subset of these, a satisfying answer will require a change of vocabulary; the explanations desired use the individual’s lexicon (e.g.it bound, it unbound, it got dephosphorylated 800 times), rather than a whole system lexicon (e.g. the abundance changed from 500 to 50). Thus, rather than tracking the whole model’s behavior (akin to a top-down approach), one needs to focus on agents, and observe their individual experiences (akin to a bottom-up approach). These approaches are complementary, as they explore a model’s intricacies from very different viewpoints. We have introduced a query language contributing to make agent-centric analysis more widespread and accessible.

1.1 Pointers to Phase I Task Deliverables

Task 1: Reasoning about Assembly
Tool: Syndra (https://github.com/csvoss/syndra)

Task 2: Reasoning about Causality
Tools: kappa-counterfactuals (https://github.com/jonathan-laurent/kappa-counterfactuals)
KaStat (https://github.com/rachelwang/KaStat)
Published Paper: Counterfactual Resimulation for Causal Analysis of Rule-Based Models Jonathan Laurent, Jean Yang, Walter Fontana. 27th International Joint Conference on Artificial Intelligence 2018.

Task 3: Querying about Rule-based Models
Tool: Kappa-TQL (https://github.com/jonathan-laurent/Kappa-TQL)
Published Paper: A Trace Query Language for KappaJonathan Laurent, Hector Medina Abarca, Pierre Boutillier, Jean Yang, Walter Fontana. 16th International Conference on Computational Methods in Systems Biology, September 2018, Brno.

2 Kappa Rule-based Modeling

Rule-based modeling of cellular interactions holds two promises: 1) to organize and assess information about protein-protein interactions that govern cellular behavior and 2) useful insights for understanding such behavior, especially new and unforeseen events, and therefore addressing disease.

As a widely used rule-based modeling formalism. The Kappa rule-based language was designed to model interactions between proteins with rewriting rules over site-graphs. In kappa, proteins are modeled by agents. An agent features some sites through which it can bind other agents. Moreover, some sites can hold an internal state, usually using ``u'' when unphosphorylated and ``p'' when phosphorylated. The number and the nature of the sites featured by an agent depend on its type, each type of agent being described in the signature of the kappa model. In detail, a shared exponent denotes the existence of a bond between two sites. Sites without exponents are considered to be free. An important remark is that a rule can feature underspecified agents. When it does, it can be triggered whatever the binding or internal states of the missing sites are. This characteristic of kappa makes it different from most modeling techniques traditionally used by biologists like differential equations systems or Petri nets. Indeed, the latter require that one variable is introduced for each fully specified species, which leads quickly to a combinatorial explosion. For instance, if a protein has 10 distinct phosphorylation sites, it yields at least 1024 different species.This situation in real world signaling networks where many proteins can bind each other and form large complexes is even worst.

In Kappa, a reaction mixture is modeled by a large site-graph. When a pattern matching the

left hand side of a rule r is recognized in it, it can be updated locally according to r. The agents preserved by r are those of the longest prefix featuring agents of the same type and mentioning the same sites in both sides of r. The sites they mention are updated according to their state in the right hand side of r and the others are left unchanged. Agent featured in the left hand side of r and not in its right hand side are removed and agents featured in the the right hand side of r and not in its left hand side are created. For instance, the rule K(d), S(x_p) → K(d), K(d) deletes an instance of S and creates an instance of K. For a rigorous definition of kappa's semantics, see [3].

3 Syndra - a Reasoning Tool for Assembly

Rule-based biological models help researchers investigate systems such as cellular signaling pathways. Although these models are generally programmed by hand, some research efforts aim to program them automatically using biological facts extracted from papers via natural language processing. However, NLP facts cannot always be directly converted into mechanistic reaction rules for a rule-based model. Thus, there is a need for tools that can convert biological facts into mechanistic rules in a logically sound way. We have constructed such a tool specifically for Kappa - Syndra, by implementing Iota, a logic language over Kappa programs. Our tool can translate biological facts into Iota predicates, check predicates for satisfiability, and find models that satisfy predicates. We have tested our system against realistic use cases, and shown that it can construct rule-based mechanistic models that are sound with respect to the semantics of the biological facts from which they were constructed. In the following, we will offer more details of the problem and our solution.

3.1 The Problem: Turning Facts into Rules

One obstacle is that facts extracted by NLP may not be able to feed directly into a rule-based model. Models need to be constructed from mechanistic rules: "Raf phosphorylates MEK at site Ser222," for example, is easy to transform into an executable model simulating levels of Raf and ERK and their states. In contrast, the facts extracted by NLP may take many forms, not all of which are clear-cut mechanistic rules. NLP may produce non-mechanistic rules, for example, such as the following facts about the Ras-Raf-MEK-ERK cancer pathway:

- Active ERK1 phosphorylates RSK. This seems mechanistic at first – phosphorylation reactions are common – but we're missing one key piece: what does it mean for ERK to be active?
- MEK phosphorylates the ERK protein family. This is type error; which members of the ERK protein family does MEK phosphorylate, and by what mechanism?
- Addition of EGF causes activation of ERK1. This tells us that activity in one protein causes activity in another protein, but we don't know how many causal steps take place in between; in fact, EGF only activates ERK1 through a pathway involving several intermediate receptors and signaling proteins.

Some facts produced by the NLP aren't even ``rules'' at all, but are still useful for constructing

a model and disambiguating other facts. We call these domain knowledge. Some examples:
• When ERK1 is phosphorylated, it is active. This is not a rule, but it helps us decode my earlier ``Active ERK1 phosphorylates RSK'' example.
• ERK1 and ERK2 are in the ERK protein family. This is not a rule, but it helps us decode my earlier ``MEK phosphorylates the ERK protein family'' example.
• S151D-mutated ERK1 behaves as if always phosphorylated.This only makes sense as an attachment to existing rules about how phosphorylated ERK1 behaves.

We want our final model to only contain mechanistic rules. For example, the above collection of non-mechanistic rules and domain knowledge is consistent with the following list of mechanistic rules:
• MEK phosphorylates ERK1.
• MEK phosphorylates ERK2.
• Phosphorylated ERK1 phosphorylates RSK.
• Phosphorylated ERK2 phosphorylates RSK.
• S151D-mutated ERK1 phosphorylates RSK.

The problem is that NLP output can be messy: it contains mechanistic rules, non-mechanistic rules, and domain knowledge, all of which must somehow be woven together in order to create an accurate model composed only of mechanistic rules.

3.2 The Solution: Logical Inference over Chemical Semantics

As a solution, we have created a tool that uses logical inference to deduce a set of clear-cut mechanistic rules that are consistent with the messier input facts NLP gives us. A good computational model should contain all available facts pertaining to the system it models, including facts that can’t be immediately converted into the rules of a rule-based model. Our solution allows us to produce models that incorporate all of this available knowledge, by performing logical inference in order to deduce which mechanistic rules are implied by the available facts.

A naïve implementation of this solution would be to come up with a set of deduction rules that can combine facts together in order to, ultimately, come up with the complete list of valid mechanistic rules. To complete the naïve approach, we would develop a set of inference rules describing the deductions that we can make among all of the kinds of facts that we might come across. Then we could add new facts to our list of facts by looking for matching inference rules, until there are no more facts to add. At that point, we could output the mechanistic rules that we had inferred, and be more confident in the completeness of our model.

However, our quest will not be to attempt to enumerate all such possible inference rules as the naïve approach would. For one thing, this approach would not scale: if there are n kinds of fact – that is, sentences with variables, like ``A activates B'' or ``A binds B'' – then there will be

Ω(n^3) different candidate inference rules. And that’s only for two-input inference rules like x ∧ y ⇒ z, not to mention three-input inference rules like x ∧ y ∧ z ⇒ w and so on. For another thing, this approach would be buggy: one could easily include an inference rule that is overly generous in what it permits, and make the entire implication system unsound.

To improve the naïve approach, we can give each kind of fact in these inference rules a definition, specified in a logical language that permits us to describe the interactions between chemical agents. Having done this, we can use this logical language in order to prove the soundness of individual inference rules, or in order to deduce properties of entire collections of facts. That way, the implications we prove will be grounded in a rigorous chemical semantics, and ensured to be sound within those semantics. Additionally, this solves our scaling problem: we only need to formalize the definitions of each of our n kinds of fact, which is better than deliberating over Ω(n^3) inference rules. Syndra is the tool that we have created that executes this improved approach. It implements a logic language for reasoning about the semantics of biological models, and implements definitions for several kinds of facts as predicates in that logic language. From there, it can reason about implications between predicates in order to prove inference rules, and it can compute a rule-based model satisfying a collection of given biological facts.

4 Reasoning about Causality for Rule-Based Models

For this task, we study the problem of analyzing the causal structure of rule-based models. More concretely, suppose we are given a fine-grained mechanistic model of a biochemical interactions network, expressed in a language such as Kappa. In Kappa, chemical interactions between proteins are modeled as stochastic graph-rewriting rules. Given a model (a set of such rules) and a phenomenon of interest, it is possible to simulate the model and see whether or not the phenomenon of interest happens and under which conditions. However, execution traces alone (which can be extremely large) do not offer an easy way into understanding why and how the phenomenon of interest emerges from atomic interactions. The aim of causal analysis is to extract causal diagrams from simulation traces, which explain this emergence in terms of primitive causal interactions between atomic events (such as activation and inhibition). These causal diagrams can be seen as formal counterparts of ``signaling pathways''.

4.1 Counterfactual Resimulation for Causal Analysis of Rule-Based Models

The previous work on this topic suffered from a significant limitation: it could only handle activation arrows, whereas inhibition plays a very important role in many biological phenomena. Our main contribution was to extend this framework to produce causal diagrams involving inhibitory influences. This is challenging because reasoning about inhibition requires reasoning about events that did not happen during simulation. Our first contribution in this

work was to propose ``counterfactual reasoning'' as a foundational framework to define and reason about inhibitory influences.

Given an observed simulation trace, we looked into counterfactual statements of the kind: ``would event B had happened had event A not happened''. Giving a semantics to such statements is tricky because of the stochastic nature of rule-based models. It is not simply possible to answer such a statement by resimulating the system after blocking event A. Intuitively, every random contingency that is not remotely affected by A must be replayed the same. In our work, we gave a semantics to counterfactual statements in the setting of rule-based models by generalizing Pearl's seminal definition of counterfactuals. Then, we proposed an efficient algorithm to evaluate such statements, which we implemented into a signaling pathway.

4.2 KaStat

We have also developed a statistical analyzer *KaStat* for Kappa models. Currently, it carries out two types of analyses: estimating the probability of a certain type of influence between two given Kappa rules with the help of statistical testing methods, and outputting the causal core that contributes most to the existence of the influence between the given Kappa rules by generating causal cores and carrying out the isomorphism checking.

With KaStat, we can 1) return a finer-grained causal core explaining the underlying causality between two specified events, 2) detect important motifs in the graphical representation of the given rule-based model, 3) remove redundancy existing in the given models, and 4) uncover evolutionary behaviors.

5 Trace Querying Language for Rule-based Models

Rule-based modeling languages such as Kappa and BioNetGen can be used to write mechanistic models of complex reaction systems. Models in these languages consist of stochastic graph-rewriting rules that are equipped with rate constants indicating their propensity to apply. Together with an initial mixture graph, these rules constitute a dynamical system that can be simulated using Gillespie's algorithm. Each run of simulation results in a sequence of transitions that we call a trace.

In practice, simulation traces are often discarded in favor of a limited number of global features, such as the concentration curves of a set of observables. However, a more detailed analysis of their structure and statistical properties can provide useful insights into a system's dynamics. For example, causal analysis methods exist that compress a large trace into a minimal subset of events that are necessary and jointly sufficient to replicate an outcome of interest, and then highlight causal influences between those remaining events (see work on counterfactual resimulation). Queries about the statistical behavior of individual agents can lead to complementary insights. Examples include 1) measuring the average lifespan of a

complex under different conditions, 2) computing a probability distribution over the states in which a particular type of agent can be when targeted by a given rule, and 3) estimating how much of a certain kind of substrate getting phosphorylated is due to a particular pathway at different points in time.

In this work, we proposed and implemented a unifying language to express queries of this kind, that are concerned with statistical features of groups of molecular events that are related in specific motifs. Out tool is actively maintained and has already been used in a study of the WNT signaling pathway, for which one of the largest rule-based models in existence has been built.

Reference

[1] Judea Pearl. Causality. Cambridge university press, 2009.
[2] Joseph Y Halpern. Actual causality. MIT Press, 2016.
[3] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Christopher D. Thompson-Walsh, and Glynn Winskel. Graphs, rewriting and pathway reconstruction for rule-based models. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, volume 18 of LIPIcs, pages 276–288. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.
[4] Daniel T Gillespie. Exact stochastic simu- lation of coupled chemical reactions. The journal of phys- ical chemistry, 81(25):2340–2361, 1977.

Personnel

In addition to the PI (Dr. Qinsi Wang) and the former PI (Dr. Jean Yang), the following students were employed as Graduate Research Assistants during the contract period:
Chelsea Voss, Master Student, 2015.9 – 2016.9
Jonathan Laurent, Ph.D. Student, 2016.9 – 2018.9.