Qualitative causal analyses of biosimulation models

Maxwell L. Neal[1], John H. Gennari[1], Daniel L. Cook[1,2]
[1] Department of Biomedical Informatics and Medical Education
[2] Department of Physiology and Biophysics
University of Washington
Seattle, WA, USA
mneal@uw.edu

***Abstract*—We describe an approach for performing qualitative, systems-level causal analyses on biosimulation models that leverages semantics-based modeling formats, formal ontology, and automated inference. The approach allows users to quickly investigate how a qualitative perturbation to an element within a model's network (an increment or decrement) propagates throughout the modeled system. To support such analyses, we must interpret and annotate the semantics of the models, including both the physical properties modeled and the dependencies that relate them. We build from prior work understanding the semantics of biological properties, but here, we focus on the semantics for dependencies, which provide the critical knowledge necessary for causal analysis of biosimulation models. We describe augmentations to the Ontology of Physics for Biology, via OWL axioms and SWRL rules, and demonstrate that a reasoner can then infer how an annotated model's physical properties influence each other in a qualitative sense. Our goal is to provide researchers with a tool that helps bring the systems-level network dynamics of biosimulation models into perspective, thus facilitating model development, testing, and application.**

Keywords—automated inference; biosimulation; biological modeling; network analysis

I. Introduction & Background

Biomedical researchers develop computational models to understand and predict the dynamics of biological processes. With increasing access to computational power and systems-level biological data, these models continue to grow in size and complexity. There is therefore a growing need for informatics tools that will help researchers quickly obtain system-level perspectives on these models so they can readily ascertain a model's architecture, its underlying assumptions, and how its components interact. Additionally, as models become more complex, the time required to generate simulation results can impede research. Motivated by these needs, we have developed an approach for performing qualitative causal analyses over biosimulation models that leverages the formal semantic structure of the Ontology of Physics for Biology (OPB [1], [2]), the Semantic Simulation (SemSim) framework [3], [4], and automated ontological inference. Our aim is to develop software that will allow users to quickly investigate the qualitative effects of perturbation experiments on biosimulation models and to identify perturbations that would influence a model element of interest. With the former, users can investigate the impact of a shift in the value of a model variable and ascertain the system-wide consequences. For example, a user investigating a metabolism dynamics might ask, "What are the consequences of increasing glucose concentration in the model?" With the latter, a user can identify which model variables, if perturbed, would shift a variable of interest in a given direction. In this case a user might ask, "Which perturbations will cause glucose concentration to increase?" Such thought experiments are a routine and valuable tool researchers use for understanding and validating hypotheses derived from their empirical studies. We anticipate that giving modelers this systems-level perspective will help guide systems biology experiments and drug target identification, as well as model-based clinical decision making.

To perform our qualitative causal analyses, we used automated inference to programmatically identify the dependencies (e.g., chemical reaction rate laws, conservation laws) used in a set of test models and the qualitative relations between the physical properties (e.g., chemical concentrations, fluid pressures) that participate in those dependencies. Using this information, we then identify how an increment or decrement in a physical property propagates to other properties in the model. For example, by semantically identifying the instances of first-order mass-action rate laws used in a chemical network model, a reasoner can infer how the physical properties involved in those laws interact, qualitatively. We can then determine how a reaction rate will, say, respond to an increase in the amount of one of its reactants. While running iterative simulations on isolated physical dependencies might also be used to accomplish this task, it is difficult to determine appropriate parameter ranges for such simulations so that the approach generalizes across physical scales and modeling domains. We therefore developed an ontology-based approach for automatically classifying dependencies and for characterizing the roles that the properties in those dependencies play.

Here we describe our qualitative reasoning approach and demonstrate it with a published chemical network model. Our approach includes two steps. First, we classify physical dependencies within the OPB and use reasoning to characterize the roles played by the properties in such dependencies. Second, we apply a simple algorithm that propagates perturbations in physical property values (increments or decrements) through the dependency network. To implement the first step, we augmented the OPB with a set of Web Ontology Language (OWL) axioms and Semantic Web Rule Language (SWRL) rules necessary for characterizing how the physical properties in a dependency influence each other, qualitatively. Thus, when we annotate a biosimulation model against the OPB, we can apply a standard OWL reasoner to infer the information needed to trace perturbations throughout a model's math dependency network.

This work was funded by National Institutes of Health (http://www.nih.gov/) grants R01LM011969, P50GM094503, and U01HL122199

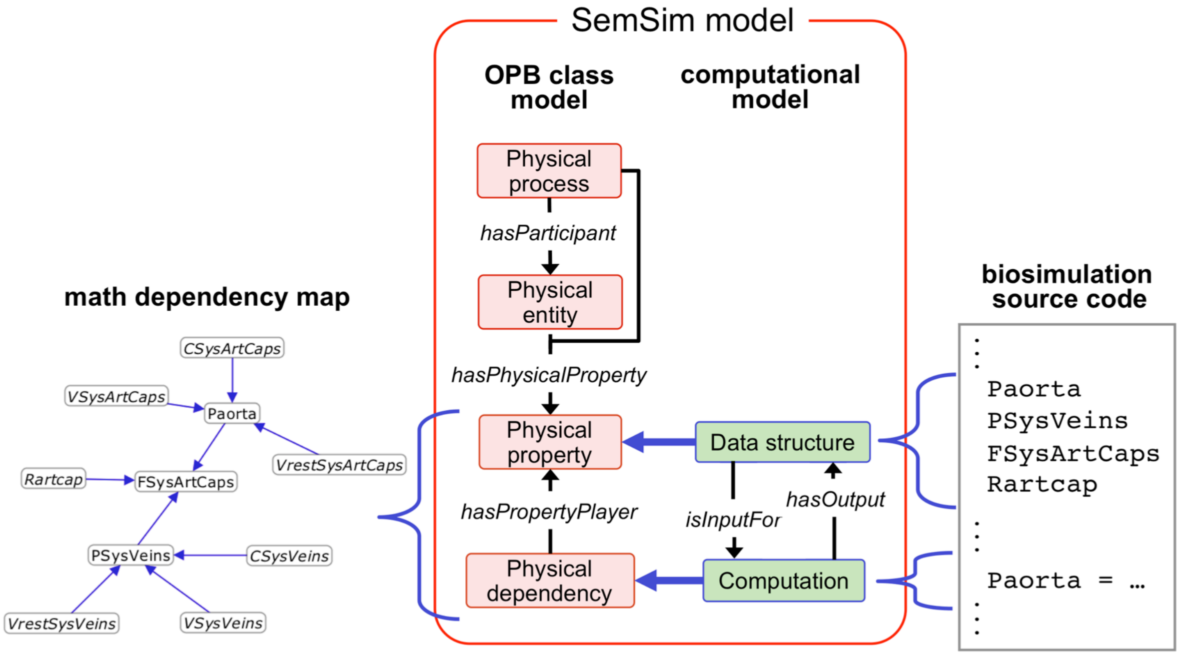


Fig. 1. Workflow by which SemGen imports and parses biosimulation model source code (right) and abstracts a SemSim model (in OWL) which is annotated against instances of OPB:*Physical property* classes. By combining the knowledge in the OPB and the SemSim model we instantiate a “math dependency map” which we traverse to propagate qualitative perturbations of physical property values.

A. Model semantics and the Ontology of Physics for Biology

Biosimulation models are becoming more readily available as public repositories such as BioModels [5] and the Physiome Model Repository [6] grow in size, and as standardized modeling formats such as the Systems Biology Markup Language (SBML) [7] and CellML [8] gain wider adoption. To make repositories more useful and amenable to automated inference, a first step is to annotate model elements to provide explicit semantics for the code-level representations of these models. Although standard modeling formats often support biological semantic annotation, little research has been carried out to describe the semantics of the physical dependencies (the mathematical laws used to relate values of physical properties to each other) being modeled. We must make these associations explicit because our qualitative causal analyses require an explicit representation of how physical dependencies determine the influence that physical properties have on each other. We therefore annotate the semantics of a model’s mathematical relationships using the OPB, which includes a taxonomy of physical dependencies and properties used in biosimulation models across biological scales [1], [2].

The OPB is a reference ontology that aims to provide a comprehensive, formal representation of the concepts used in modeling biophysical processes. It includes, for example, an OPB:*Physical property* taxonomy for annotating model variables (e.g., OPB:*Charge amount* for electrical charges) and an OPB:*Physical dependency* taxonomy (e.g., OPB:*Electrical resistive dependency* for Ohm’s Law) for representing the biophysical meaning of a model’s mathematical dependencies. Previously, we have leveraged the OPB for multi-scale model integration tasks [4], [9]. As shown in the middle panel of Figure 1, we use OPB:*Physical property* classes to annotate computational data structures that represent the quantitative values of model variables. OPB:*Physical property* is a dual inheritance hierarchy that distinguishes dynamical property type (e.g., forces vs. flow rates vs. amounts) and dynamical domain (e.g., fluid domain vs. chemical domain vs. electrical domain). Thus, a variable representing blood flow rate in the aorta, say, is semantically distinct and non-conformable with a variable for cytosolic glucose concentration.

The OPB:*Physical dependency* hierarchy represents mathematical dependencies between physical property values. This hierarchy is also dual-inheritance; the OPB dependencies are sub-classed according to dynamical domain and by dynamical dependency type. For example, an instance of OPB:*Resistive dependency* represents an occurrence of a process governed by an analog of Ohm’s Law. Thus, in the electrical domain an electrical current depends on an electrical voltage and the electrical resistance in the conducting pathway: $I=V/R$, classically. Likewise, in the fluid domain, a fluid flow rate depends on a pressure difference and a fluid flow resistance (an analog of electrical resistance). In the OPB schema, each OPB:*Physical property* instance has two key relations. First, it is linked by a *hasPhysicalProperty* relation to an instance of the entity or process that bears the property, and by a *hasPropertyPlayer* to the OPB:*Physical dependency* instance according to the model code. In Section II we describe extensions to the OPB:*hasPropertyPlayer* relations required for tracking property value perturbations through a modeled system.

B. The SemSim framework

Based on the biophysical semantics of the OPB, we established the SemSim model framework for annotating and representing the mathematics of simulation models (Figure 1). Each SemSim model represents the computational structure of a model and provides a logical framework for annotating its variables and computations in terms of OPB classes. Applications of the SemSim framework are limited in our work to the domain of discrete causal models written and archived using ordinary differential equations. In prior work, we have created SemSim models to annotate, decompose, and recompose models in any of several modeling languages: SBML, CellML, and JSim's Mathematical Modeling Language [10]. To accomplish this we have developed SemGen [4], [9], a Java-based semantic modeling application that automatically creates SemSim models from models coded in these languages. The SemSim framework relies on composite annotations [3] to formally describe the biological meaning of model elements, and SemGen provides tools for creating these annotations *de novo* and for auto-generating them from annotations present in curated models such as the those in BioModels. Each composite annotation consists of an instance of an OPB:*Physical property* class that identifies the physical property type of the variable and is classifiable within the OPB:*Physical property* hierarchy. As we describe below, this critical feature allows us to automatically identify the mathematical dependencies as pathways by which perturbations propagate throughout a set of modeled properties.

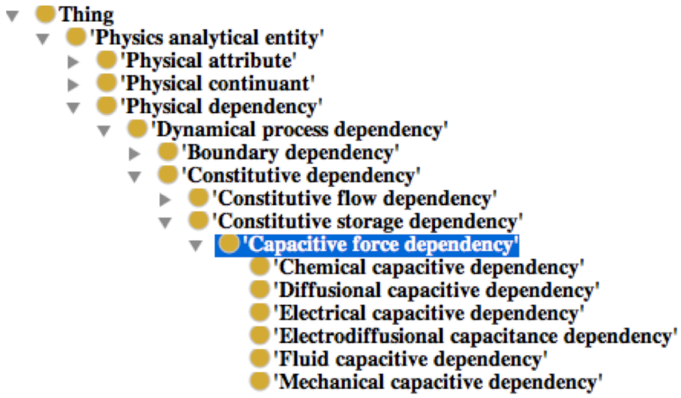
II. Methods

To support causal qualitative reasoning, we must augment the OPB, apply a standard OWL reasoner to produce the math dependency map, and then apply an algorithm to determine how qualitative perturbations propagate through a model's mathematical network. There are two types of semantic augmentations we add to the OPB: axioms that define dependencies based their property players, and axioms and rules that determine which properties are positive or negative players in a dependency.

A. Classifying physical dependencies in the OPB

The first steps in performing our qualitative analyses are to convert a model into the SemSim format using SemGen and then annotate its physical properties against the OPB. As we will describe below, this step allows us to then classify the physical dependencies in the model and automatically infer the qualitative influence that physical properties have on each other. Many of the models we tested for this study were curated SBML models from BioModels. For these we were able to use SemGen to automatically annotate the model's physical properties against OPB classes, the physical entities they are properties of, and the processes in which the entities participate. We then leveraged this critical information to identify the qualitative role that the properties play in the model's dependencies.

Since publicly available models do not contain annotations against the OPB that indicate which mathematical dependencies are used in a model, we developed a method for automatically classifying a model's physical dependencies within the OPB hierarchy. To provide an example of a physical dependency and how we classify it, consider the law of fluid compliance, often used in hemodynamic models: $P = V/C$. Here P is the transmural fluid pressure in a compartment, V is the compartment fluid volume, and C is the compartment compliance. (Modelers may choose to use a fluid *elastance* instead of compliance as they are reciprocal.) This law is analogous to the law of capacitance used in electrical dynamics models. Instead of pressure, volume and compliance, the electrical analog relates voltage, charge, and capacitance. As shown in Figure 2, these two physical dependencies are subclasses of OPB:*Capacitive force dependency*, which subsumes analogous versions of this law for each physical domain represented in the OPB.



- Thing
  - 'Physics analytical entity'
    - 'Physical attribute'
    - 'Physical continuant'
    - 'Physical dependency'
      - 'Dynamical process dependency'
        - 'Boundary dependency'
        - 'Constitutive dependency'
          - 'Constitutive flow dependency'
          - 'Constitutive storage dependency'
            - 'Capacitive force dependency'
              - 'Chemical capacitive dependency'
              - 'Diffusional capacitive dependency'
              - 'Electrical capacitive dependency'
              - 'Electrodiffusional capacitance dependency'
              - 'Fluid capacitive dependency'
              - 'Mechanical capacitive dependency'

Fig. 2. Location of *Capacitive force dependency* in the OPB class hierarchy.

Leveraging this hierarchical structure, our approach for automatically identifying the physical dependencies used in our test models was to first create logical definitions in the form of OWL equivalent class expressions for higher-level OPB physical dependency classes. These statements define physical dependencies in terms of the physical properties that participate in them. We term these participating physical properties "property players", language that is reflected in the object property names used to construct our logical class definitions. For example, we used the following equivalent class expression to define OPB:*Capacitive force dependency*, presented here in Manchester style syntax:

```
((hasPropertyPlayer some Capacitance) or
(hasPropertyPlayer some Elastance))
and (hasPropertyPlayer some 'Amount property')
and (hasSolvedPropertyPlayer some 'Force proper-
ty')
```

We then applied simple equivalent class axioms to define the domain-specific subclasses of OPB:*Capacitive force dependency*. For example, the equivalency axiom for OPB:*Fluid capacitive dependency* is

```
'Capacitive force dependency'
and (hasPropertyPlayer only (hasPhysicalDomain
some 'Fluid kinetic domain'))
```

This captures the knowledge that OPB:*Fluid capacitive dependency* is an OPB:*Capacitive force dependency* that only relates physical properties from the fluid kinetic domain.

B. Characterizing the qualitative influence among properties

To represent how the different physical properties in a dependency will react to a perturbation in one of those properties, we first identify the physical property that is the "output" of the dependency, as represented in the original SemSim model. SemSim models currently represent equations as having one output variable on the left hand side (LHS) and the expression that solves it on the right hand side (RHS). We use OWL object property axioms such as *Dependency_A <hasSolvedPropertyPlayer> Property_B* to capture the knowledge that *Property_B* is on the LHS of *Dependency_A*. Physical properties on the RHS that produce the same response in the solved property when increased or decreased are termed "positive property players". Properties that produce a change in the opposite direction are termed "negative property players." This knowledge is captured using object property statements on the physical dependency individuals such as *Dependency_A <hasPosPropertyPlayer> Property_C*. For example, the fluid volume property in the fluid compliance law mentioned above is a positive property player in the dependency; if increased, the solved fluid pressure property on the LHS also increases. Conversely, the fluid compliance property is a negative property player in the dependency: if it is increased, fluid pressure decreases. Because SemSim models do not include these axioms, we use class restriction axioms on OPB dependency classes, a set of SWRL rules, and a reasoning engine to generate them automatically. For example, we added the following class restriction to OPB:*Capacitive force dependency*:

```
(hasPosPropertyPlayer exactly 1 'Amount property')
and (((hasPropertyPlayer exactly 1 Capacitance)
and (hasNegPropertyPlayer exactly 1 Capacitance))
or ((hasPropertyPlayer exactly 1 Elastance)
and (hasPosPropertyPlayer exactly 1 Elastance)))
```

This axiom indicates that if a physical dependency individual is classified as an OPB:*Capacitive force dependency*, it must have exactly one positive property player that is an OPB:*Amount property* (e.g., fluid volume or electrical charge), along with one negative player that is an OPB:*Capacitance* (e.g., fluid compliance or electrical capacitance) or one positive player that is an OPB:*Elastance* (e.g., fluid elastance or electrical elastance). In combination with the equivalent class axioms mentioned above, an automated reasoner can infer from these axioms that if a physical dependency is an OPB:*Capacitive force dependency* and it has one fluid compliance property player, then the compliance is a negative property player. As described in Section III, we can then use this knowledge to determine how a perturbation in the compliance will affect the fluid pressure property in the dependency.

These class restriction axioms are effective for characterizing properties as positive or negative players when the properties differ in type, as in our fluid compliance law example. However, they are less effective when multiple instances of the same property type are on the RHS of a dependency. For example, the mass conservations equations used to compute fluid volume changes in hemodynamic models or the species amounts in chemical network models often have multiple flow rates on the RHS. These rates may be positive or negative players, depending on the model's formulation. For example, the conservation equations that determine chemical species amounts are often formulated by summing the rates of the reactions that produce a species and subtracting those that consume it. The rate of a reaction that consumes a chemical would be a negative player in the dependency, while the rate of a reaction that produces it would be a positive player. To determine the polarity of the property players in these dependencies, we created five SWRL rules that leverage the model's underlying biological semantics. As an example, the following rule identifies negative players on the RHS of mass conservation equations:

```
hasSolvedPropertyPlayer(?dep1, ?prop1),
hasPropertyPlayer(?dep1, ?prop2),
physicalPropertyOf(?prop1, ?ent1),
hasSource(?process, ?ent1),
physicalPropertyOf(?prop2, ?process)
    -> hasNegPropertyPlayer(?dep1, ?prop2)
```

The rule states that if a physical dependency solves for a property of a physical entity, and there is a property of a process that consumes the entity on the RHS, then the latter property is a negative player in the dependency. As a specific example, this rule indicates that for all reactants in a reaction, the reaction rate is a negative player in the conservation equation that determines the amount of the reactant. Thus, if the reaction rate increases, the reactant amount decreases. We created a similar rule for dependencies that solve for the amount of products involved in reactions. In this case, the reaction rate is a positive player in the dependency. We also created three additional SWRL rules to identify the polarity of property players in dependencies that solve for reaction rates:

1. If a process rate is solved using the property of a thermodynamic source of that process, then the source property is a positive player
2. If a process rate is solved using the property of a thermodynamic sink of that process, then the sink property is a negative player
3. If a process rate is solved using the property of a mediator of that process, then the mediator property is a positive player

Rule 1, for example, identifies reactant concentrations on the RHS of first-order reaction rate equations as positive players in those equations; as reactant concentration increases, so does the reaction rate. We achieved our inference goals applying only a very few SWRL rules because the basic dependency relations are declared at a superclass level in the OPB:*Physical dependency* hierarchy. Thus, a single SWRL rule suffices for both fluid and electrical analogs of Ohm's Law (i.e., OPB:*Resistive flow dependency*) or for Hooke's Law (i.e., OPB:*Capacitice force dependency*; see Figure 2) as examples.

C. *Automatically inferring math dependency maps*

To automatically classify the dependencies in a model and identify their positive and negative players, we first automatically instantiated the SemSim model's physical dependencies, properties, entities and processes as OWL individuals in our augmented OPB. Properties were asserted as individuals in whichever OPB:*Physical property* class was used in their composite annotation. Dependencies, entities and processes were asserted as individuals in the top-level OPB classes representing these concepts. Using the SemSim object model underlying SemGen, we automatically identified which dependencies involved which property players. As mentioned above, we asserted this knowledge by applying *hasSolvedPropertyPlayer* and *hasPropertyPlayer* object property axioms on the dependency individuals. We then used HermiT [11] version 1.3.8 to programmatically classify the resulting ontology and infer which properties were positive players in the model's dependencies and which were negative. This single classification step simultaneously reasons over the new OPB OWL class restrictions and the SWRL rules described above to generate a math dependency map for that model. In the next section, we describe how we used the inferred information from the reasoner to traverse this map and determine the outcomes of perturbation experiments.

III. Qualitative perturbation experiments

The inferences produced by the HermiT reasoner include the knowledge necessary to determine how an increment or decrement in a given physical property within a dependency will affect the property that is solved by the dependency. To investigate the system-wide effects of such perturbations, we developed a Java-based algorithm that takes as input a property perturbation (an increment or decrement), and determines how that perturbation propagates throughout the math dependency network of the model. This algorithm, similar to those used in colored petri net analyses, is based on the "PathTracing" function of the Chalkboard software for composing and investigating the dynamics of network models [12]. Our algorithm begins by identifying the dependencies in which the perturbed property participates, determines how the perturbation affects the dependency's solved property based on the perturbed property's positive/negative player status, records how the solved property is perturbed, then continues iteratively by identifying the dependencies in which the solved property participates. Iteration continues until the algorithm returns to the original perturbed property or, in the case of negative feedback loops, if it finds that a physical property is both increased and decreased by a perturbation. The algorithm then lists the physical properties that are increased by the perturbation, those that are decreased, and those that are influenced in both directions (an ambiguous effect).

By simply reversing this algorithm and traversing the math dependency map in the opposite direction, we are also able to investigate which perturbations will cause a property of interest to increase or decrease. In this case, the inputs to the algorithm are a property of interest, and whether it is increased or decreased. The algorithm then lists how to perturb the other properties in the model to shift the property of interest in the specified direction.

A. *Scope and caveats*

Before describing the performance of our qualitative inference methods, we want to properly position our work and tools in the context of biological research and modeling. Our aim is to offer a "thinking tool" of practical use to those who struggle to understand the behavior of complex causal networks. From our own experience, both investigators and students routinely explore the behavior of these systems by tracing event paths through both informal and consensus network representations. Thus, our methods are intended to formalize, illustrate and explore biological hypotheses.

We are well aware of important assumptions and limitations to the qualitative pathway analysis that we have implemented. First, we assume that the system is at a stable operating point far away from dynamical instabilities and that perturbations are infinitesimal. Thus, we assume that perturbations propagate through the system without changing the signs of the partial differentials between dependent variables. Even with this assumption, however, ambiguities can arise when an increment converges with a decrement to produce an ambiguous result resolvable only by quantitative analysis. That being said, an ambiguity resulting from feedback may be interpretable: a positive feedback loop will only reinforce the originating perturbation while one *might* be able to assume that a negative feedback signal is quantitatively less than the originating signal and, hence, can be ignored by the propagation algorithm. However, we would caution that the longer the pathway, the less confidence one might have in the result, especially in highly coupled networks.

B. *Example perturbation experiment*

Figure 3 illustrates the results of our approach. It shows the outcome of a qualitative perturbation experiment on BioModel #313, an SBML-encoded model that simulates the Janus kinase signaling pathway in a primary mediastinal B-cell lymphoma cell line. In the model's source publication [13], the authors used numerical sensitivity analysis to identify the STAT5 protein as a potential drug target that could reduce the amount of *CD274*, an mRNA associated with cell proliferation. Guided by this result, we tested whether our qualitative approach would generate similar results. We converted the SBML model into the SemSim format, used automatic classification to identify the positive and negative players in the model's dependencies, and performed a perturbation experiment where we decreased STAT5. Automatic classification of the model via HermiT finished in four seconds on a MacBook Pro 2.8 GHz Intel Core 2 Duo laptop. Performing the perturbation on the classified ontology's math dependency network is much faster, finishing in 33 milliseconds. As shown in Figure 3, the network-wide results of the perturbation experiment include a decrease in *CD274* mRNA.

IV. Discussion

This study demonstrates how the comprehensive, hierarchical knowledge contained in the OPB can be leveraged to automatically identify physical dependencies in models and then characterize how the physical properties in these models influence each other. This inferred knowledge can form the basis of qualitative perturbation experiments, as reported here, but it could also potentially be used to help guide and/or automate the model annotation process. For example, once a model's physical dependencies are classified within the OPB, this knowledge could be applied to flag inconsistent model annotations and also suggest which OPB terms to use for unannotated physical properties.

Our intent with this work is not to supplant the more precise, quantitative analyses that numerical simulations of biosimulation models provide, but rather develop an additional tool that researchers can use to quickly check model assumptions and explore how qualitative perturbations propagate through networks. The perturbation experiment shown in Figure 3 demonstrates that we can quickly generate systems-level information that could potentially help researchers identify drug targets without the costs associated with iterative numerical simulation. While a concern for the use of rich ontological representations and inference can be the computation cost of general-purpose reasoning engines, our experience shows that

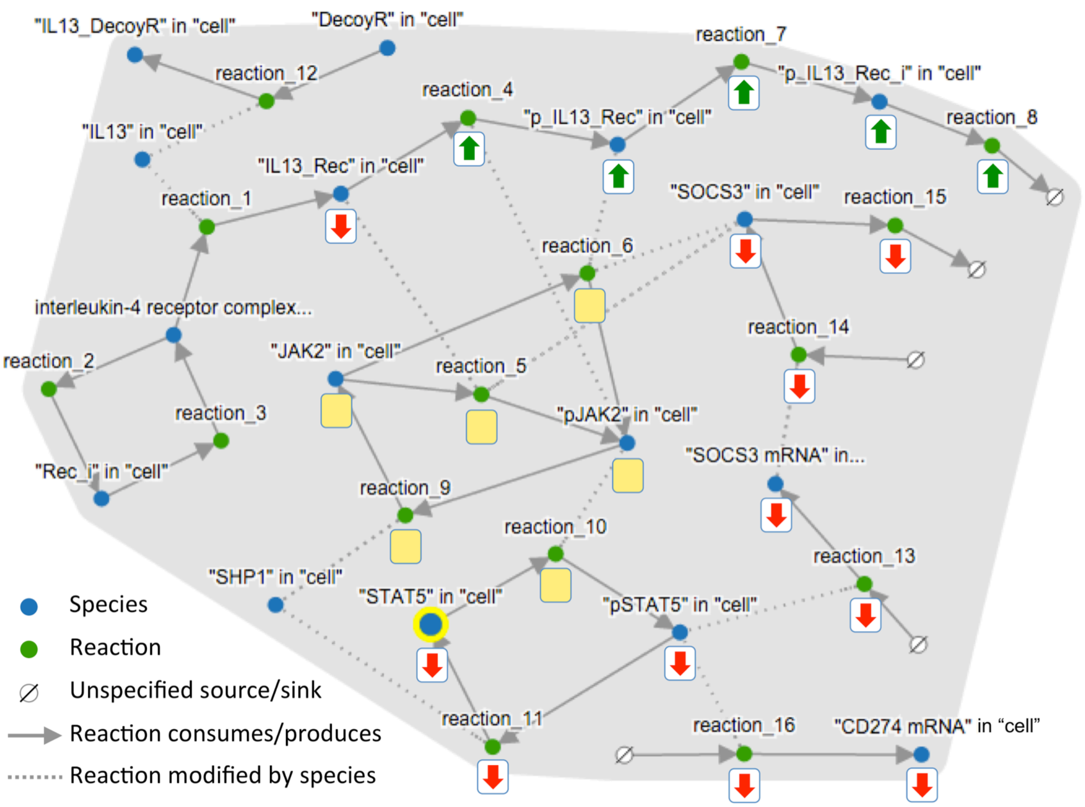


Fig. 3. BioModel #313 reaction network. Overlaid are the results of a qualitative perturbation experiment where species STAT5 is decreased. Qualitative shifts for species and reactions are indicated in rectangles below nodes. Yellow rectangles indicate an ambiguous shift. Rectangles are absent for nodes unaffected by the perturbation.

for models similar in size to that of Figure 3, generating the model's math dependency map using HermiT only required several seconds.

For this study we have combined semantics-based modeling formats, description logics, rules and automated reasoning to represent qualitative causal networks in biosimulation models. We have demonstrated how we can analyze these networks to gain systems-level perspectives on a model's response to qualitative perturbations. Given that the OPB represents physical dependencies and properties across physical scales and domains, our approach extends beyond chemical network modeling; theoretically, it can be applied to models from the molecular to the whole-body level. Additionally, our approach can be applied to any of the several hundred models available through public repositories that are convertible into the SemSim format.

Acknowledgments

We thank Christopher Thompson and Karam Kim for helping create the network diagram in Figure 3 as well as Robert Hoehendorf for helping us select an automated reasoner for our approach.

References

[1] D. L. Cook, M. L. Neal, F. L. Bookstein, and J. H. Gennari, "Ontology of physics for biology: representing physical dependencies as a basis for biological processes.," J. Biomed. Semantics, vol. 4, no. 1, p. 41, 2013.

[2] D. L. Cook, F. L. Bookstein, and J. H. Gennari, "Physical Properties of Biological Entities: An Introduction to the Ontology of Physics for Biology," PLoS One, vol. 6, no. (12), p. e28708, 2011.

[3] J. H. Gennari, M. L. Neal, M. Galdzicki, and D. L. Cook, "Multiple ontologies in action: Composite annotations for biosimulation models," J. Biomed. Inform., vol. 44, no. 1, pp. 146–154, 2011.

[4] D. A. Beard, et al., "Multi-scale modeling and data integration in the Virtual Physiological Rat Project," Ann. Biomed. Eng., vol. 40, no. 11, pp. 2365–2378, 2012.

[5] N. Le Novere, et al., "BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems," Nucleic Acids Res., vol. 34, no. suppl 1, pp. D689–D691, 2006.

[6] T. Yu, et al., "The physiome model repository 2," Bioinformatics, vol. 27, no. 5, pp. 743–744, 2011.

[7] M. Hucka, et al., "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," Bioinformatics, vol. 19, no. 4, pp. 524–531, 2003.

[8] A. A. Cuellar, et al., , "An overview of CellML 1.1, a biological model description language," Simulation, vol. 79, no. 12, pp. 740–747, 2003.

[9] M. L. Neal, et al., "Semantics-based composition of integrated cardiomyocyte models motivated by real-world use cases," PLoS One, vol. 10, no. 12, p. e0145621, Dec. 2016.

[10] E. Butterworth, B. E. Jardine, G. M. Raymond, M. L. Neal, and J. B. Bassingthwaighte, "JSim, an open-source modeling system for data analysis.," F1000Research, vol. 2, p. 288, 2013.

[11] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "HermiT: An OWL 2 Reasoner," J. Autom. Reason., vol. 53, no. 3, pp. 245–269, 2014.

[12] D. L. Cook, J. H. Gennari, and J. C. Wiley, "Chalkboard: Ontology-Based Pathway Modeling And Qualitative Inference Of Disease Mechanisms," Pac Symp Biocomput, vol. 12, pp. 16–27, 2007.

[13] V. Raia, et al., "Dynamic mathematical modeling of IL13-induced signaling in Hodgkin and primary mediastinal B-cell lymphoma allows prediction of therapeutic targets," Cancer Res., vol. 71, no. 3, pp. 693–704, 2011.