A Contradiction-Based Framework for Testing Gene Regulation Hypotheses

Steve Racunas, Nigam Shah, and Nina V. Fedoroff
The Pennsylvania State University
sar147@psu.edu, nigam@psu.edu, nvf1@psu.edu

Abstract

We have developed a mathematical framework for representing and testing hypotheses about gene, protein, and signaling molecule interactions. It takes a hierarchical, contradiction-based approach, and can make use of multiple data sources to assess hypothesis viability and to generate a viability partial order over the space of hypotheses. We have developed an event-based formal language for the expression of such hypotheses. This language seamlessly integrates regulatory diagrams (graphical inputs) and structured English (text input) to maximize flexibility. We have developed a mathematical formalism that allows us to make precise statements about hypothesis similarity and the convergence of iterative refinements of a base hypothesis. To this, we add mathematical machinery that allows us to make precise statements about control and regulation.

1. Introduction

The representation of diverse data and biological processes in a single formalism is a major challenge in bioinformatics [4], because of the variety of available data and the varying levels of detail at which biological processes can be considered [1]. Biologists are in a situation similar to the one engineers faced when they needed to analyze dynamical systems for which existing empirical means proved inappropriate[11]. We extend symbolic dynamics methods that have been applied to engineering problems, in order to develop an event-based representation scheme that can handle the nonlinearity and complexity of biological systems. Our scheme performs data fusion on the logical level, allowing us to incorporate heterogeneous data (a stumbling block for previous biological systems modeling efforts[4]).

We have modified and extended Discrete Event Systems control theory [5] to allow for the discovery of control mechanisms for gene regulatory networks. We have developed an event-based description language for representing biological processes and data from different experimental sources at varying levels of resolution, and use this language to support qualitative reasoning about biological systems. We have constructed a contradiction-based hypothesis testing framework to aid in the discovery of gene regulatory control mechanisms.

2. Hypothesis Representation

To permit computer-aided composition and evaluation of hypotheses, we must be able to specify hypotheses in a well-defined and machine-understandable way. We have designed a formal grammar for hypothesis composition and a formal mathematical structure for the space of all expressible hypotheses. We have developed both graphical and structured text representations for our hypothesis language, and have specified the context-free grammar that generates this language.

Let $\mathcal{G}$ be a context free grammar [10] of the general form:

Clause → Subject . Verb . Object
Clause → Subject . Verb . Object . Context
Subject → (Actor | Context | Clause)
Verb → (PhysicalOperator | BiologicalOperator | LogicalOperator)
Object → (Actor | Context | Clause)
Actor → (Gene | Protein | SignalingMolecule . . .)
Context → (PhysicalContext | TemporalContext | ExperimentalContext)
PhysicalContext → (InsideCell | InsideNucleus . . .)
TemporalContext → (ObservationTime1 . . .)
ExperimentalContext → (Treatment1 | Treatment2 . . .)

. . . where the lists of terminals for Contexts, Genes, Proteins, Signaling Molecules, and Operators are system dependent. These terminals may be taken from any of the large (and growing) number of ontologies currently under development[2, 8, 9] for biological systems.

This sample grammar specifies a structured text representation for a hypothesis composition language. A graphical representation may be specified as follows: Let Actors be represented by points in the plane. Let Operators be represented by directed line segments or arcs. Let Contexts be represented by closed curves in the plane. An Operator that terminates on an Actor, or that terminates within a Context, specifies that Actor or Context as the Object of an assertion. Similarly, the origin of the Operator specifies the Subject. The assertion will be evaluated within the context corresponding to the smallest region that completely encloses all of its elements.

2.1. Hypothesis Space

Let X be a set, and $\mathcal{P}(X)$ be the set of all subsets of X.

Definition 2.1 (Filter Basis). A collection $\mathcal{B} \subseteq \mathcal{P}(X)$ is a **filter basis for X** if

1. $\emptyset \notin \mathcal{B}$, and
2. For all $B1 \in \mathcal{B}$, for all $B2 \in \mathcal{B}$, there exists $B3 \in \mathcal{B}$, such that $(B3 \subseteq (B1 \cap B2))$

Definition 2.2 (Filter). A collection $\mathcal{F} \subseteq \mathcal{P}(X)$ is a **filter for X** if

1. $\emptyset \notin \mathcal{F}$
2. For all $F1, F2 \in \mathcal{F}, ((F1 \cap F2) \in \mathcal{F})$

Definition 2.3 (Filtering). A **filtering** is the set of all filters of an underlying space. Let X be a set. We write ΦX to denote the set of all filters of X.

Definition 2.4 (Evaluation Rule). Let $\mathcal{H}$ be the set of all sententials of the grammar $\mathcal{G}$.

An **evaluation rule** is any mapping that takes $\mathcal{H}$ to $\{TRUE, FALSE\}$.

Let $\mathfrak{R}$ denote the set of evaluation rules for a set of hypothesis about a specific biological system. Evaluation rules are grouped into families. Let $\mathfrak{R}_{i,j}$ denote an evaluation rule taken from $\mathfrak{R}$. The first subscript denotes the family to which the given sorting rule belongs, and the second indexes a rule from that family.

Definition 2.5 (Hypothesis Space). Let $\mathcal{G}$ be a grammar. Let $\mathcal{H}$ be the set of all sententials generated according to $\mathcal{G}$. Let $\mathfrak{R}$ be a set of evaluation rules.

We use $\mathfrak{H} = (\mathcal{H}, \mathfrak{R})$, to denote the **hypothesis space** formed by the application of the given rules to the given productions.

Definition 2.6 (Forbidden Set). Let $\mathcal{C}_{i,j}$ denote the set of expressible hypotheses that are contradicted by the particular rule: $\mathfrak{R}_{i,j}$. This is the **forbidden set** with respect to $\mathfrak{R}_{i,j}$.

Definition 2.7 (Useless Rule). An evaluation rule that contradicts no hypotheses is a **useless rule**.

Definition 2.8 (Pointless Rule). An evaluation rule that contradicts all possible hypotheses is a **pointless rule**.

Definition 2.9 (Hypothesis Valuation Filter Basis). Let $\mathfrak{R}_{i,j}$ be a set of evaluation rules, such that no rule is useless and no rule is pointless. Let $\mathcal{C}_{i,j}$ be the set of hypotheses contradicted by rule: $\mathfrak{R}_{i,j}$ (i.e. the function returns "FALSE"). Let $\mathcal{A}_{i,j} = \mathcal{H} \setminus \mathcal{C}_{i,j}$. Let $\mathcal{A}_i^! = \cap(\mathcal{A}_{i,\cdot})$

Then, we define $\mathcal{A}_i^{fb} = \mathcal{A}_i^! \cup_j \mathcal{A}_{i,j}$ to be the **hypothesis valuation filter basis** (i.e. the hypotheses that each rule within the family "j" fails to contradict, along with the hypotheses that contradict *no* rules from this family).

Lemma 2.1. *The hypothesis valuation $\mathcal{A}_i^{fb}$ is a filter basis.*

Proof. The union of sets of hypotheses is a subset of $\mathcal{P}(\mathcal{H})$. Condition 1 is satisfied by the assumed absence of pointless and useless rules. Condition 2 is satisfied by the definition of $\mathcal{A}_i$ to include $\cap\{\mathcal{A}_{i,\cdot}\}$. □

Thus, the set of all hypotheses accepted by the rules at a given level, together with the set of hypotheses accepted by *all* rules at that level, form a filter basis. Further, the set of all accepted hypotheses at any given level, plus all of their intersections, form a filter, as we now show.

Definition 2.10 (Hypothesis Valuation Filter). Let $\mathfrak{R}_{i,j}$ be a set of evaluation rules, such that no rules is useless and no rule is pointless. Let $\mathcal{A}_{i,j}$ be the set of hypotheses accepted by rule $\mathfrak{R}_{i,j}$. Let $\mathcal{A}_i^{!!} = \bigcup_{\mathcal{A}^S \subset \mathcal{A}_{i,\cdot}} (\cap \mathcal{A}^S)$.

We define $\mathcal{A}_i = \mathcal{A}_i^{!!} \cup_j \mathcal{A}_{i,j}$, the set of all possible intersections of hypotheses accepted by rules $\mathfrak{R}_{i,\cdot}$ together with the accepted hypotheses, to be the **hypothesis valuation filter**.

Lemma 2.2. *The hypothesis valuation $\mathcal{A}_i$ is a filter.*

Proof. Condition 1 follows as before. Condition 2 is satisfied by the inclusion of all possible intersections. □

2.2. Hypothesis Similarity Relationships

We wish to be able to formulate and test, in a rigorous sense, statements about the similarity of hypotheses.

Let $\mathcal{H}$ be the set of all hypotheses.

Roughly speaking, the neighborhood of hypothesis $x \in \mathcal{H}$, denoted $\mathcal{N}(x)$, indicates which other hypotheses that are "close to" hypothesis x. More formally:

Definition 2.11 (Neighborhood Function). A **neighborhood function** is a function from a set to the neighborhoods of points in that set, such that the entire set is in the furthest (largest) neighborhood of each point (**Property N0**).

A neighborhood of a point in hypothesis space inherits the **isotonicity** and **sublinearity** properties of collections in a straighforward manner:

Definition 2.12 (Isotonicity (Property N1)). A neighborhood function $\mathcal{N}(\cdot)$ is **isotone** if the following holds true:

$$\forall x, \forall N \in \mathcal{N}(x), N \subseteq N' \implies N' \in \mathcal{N}(x) \quad (1)$$

Definition 2.13 (Sublinearity (Property N2)). A neighborhood function $\mathcal{N}(\cdot)$ is **sublinear** if the following is true:

$$\forall x, \forall N \in \mathcal{N}(x), \forall N' \in \mathcal{N}(x), (N' \cap N) \in \mathcal{N}(x) \quad (2)$$

We also define:

Definition 2.14 (Expansiveness (Property N3)). We say that a neighborhood function $\mathcal{N}(\cdot)$ is **expansive** if the following holds true:

$$\forall x, \forall N \in \mathcal{N}(x), x \in N \quad (3)$$

2.3. Hypothesis Neighborhoods

Suppose a hypothesis is found to be in error, but the error is of a sort that is repairable given information from the knowledge base. In other words, there is a hypothesis that is “close to” the original hypothesis, for which more complete agreement with the experimental characterization of the system may be observed. We want to report back to the hypothesis’ composer both a diagnosis of the hypothesis and an alternate hypothesis suggestion that rectifies the contradiction.

We therefore wish to be able to ascertain, in a rigorous manner, which hypotheses are “close to” other hypotheses according to the structure imposed by the evaluation rules. We wish to define a neighborhood structure on the set of expressible hypotheses that is based on the satisfaction of the evaluation rules, as follows:

Let $v(\cdot,\cdot)$ be a valuation relation.

We will define $\mathcal{N}(\cdot)$ to be a generalized neighborhood function mapping X to $\mathcal{P}(\mathcal{P}(X))$, such that (minimally) for all $x \in X, (X \in N(x))$.

Remark 2.15. One notion of a “neighborhood” can be constructed as follows:

$$\mathcal{N}(x) = \cap \{F \in \Phi\mathcal{H}, ((F, x) \in v)\} \quad (4)$$

Remark 2.16. We choose a neighborhood function for $\mathcal{H}$ such that:

1. The hypothesis space is in the furthest neighborhood of every hypothesis, so that N0 is valid.
2. The addition of any hypothesis to a neighborhood of hypothesis generates a new, larger neighborhood (i.e. **isotoninity** holds).
3. Any given hypothesis is located within every hypothesis-neighborhood of itself, so that N2 is satisfied.
4. A hypothesis is in every neighborhood of itself, satisfying the **expansiveness** property.

Definition 2.17 (Pre-topological Space). A space whose associated generalized neighborhood function satisfies the isotonicity, sublinearity, and expansiveness properties constitutes a **pre-topological space** [7].

Fact 1. The Neighborhood Valuation Filter induces a pretopology upon the space of all expressible hypotheses.

As a hypothesis is composed and then iteratively refined, it may approach some underlying essential assertion in a manner analogous to the way in which a sequence converges to a limit value. The specification of a pre-topology allows us to make precise statements about the convergence and continuity of sequences of hypotheses. Fortunately, convergence has been defined for structures that include filters [7].

Definition 2.18 (Convergence). Let $c \in (\mathcal{P}(X) \times X)$, and let $F \in \Phi X$.

The filter F converges to x under c if the following conditions hold true:

1. $(F, x) \in c$
2. $\forall G \in \mathcal{P}(X), (F \subseteq G \implies (G, x) \in c)$

We will test for convergence of hypotheses with respect to relations derived from constraints enforced by the knowledge model and by the available data.

3. Constraints and Control Determination

In addition to testing hypotheses against existing data, we will rank hypotheses with respect to domain expert knowledge as well. To accomplish this, we specify a temporal logic for the expression of constraints. This logic includes the following operators: (Disjunction), (Conjunction), (Until), (Awaits), (Since), (Back to), (Next), (Henceforth), (Eventually), (Previous), (Weak Previous), (Always in the past), (some time in the future), and (negation).

To determine the control mechanisms that are operating in a given network, we must first define what conditions a control mechanism must satisfy to evidence that it is controls a given system for which we have experimental observations.

The classical formalism for determining control in discrete event systems [5] was developed by Ramadge and Wonham. They assume an alphabet, Σ, which is a set of symbols. In the Ramadge-Wonham framework, these symbols correspond to **atomic events** of the dynamical system. In the Ramadge-Wonham paradigm, atomic events are considered to occur spontaneously (with no auxiliary forcing mechanism), asynchronously (without reference to a timescale), and instantaneously (events may not be subdivided or interrupted).

In the standard controls paradigm, a **supervisor** is constructed, that restricts a model of the physical system (the **plant**) to displaying only desirable behaviors. Both plant and supervisor are described by regular languages over Σ and the question of control is phrased as a property of these languages:

Definition 3.1 (Controllability). Let M be a finite state automaton plant model and S be a supervisor for M. Let L be the language generated by M, and let K be the language of S controlling M. Then we say that S controls M if and only if the following statement holds:

$$\forall s \in \Sigma^* \forall u \in \Sigma_u \left(s \in \overline{K} \implies \left(su \in \overline{L} \iff su \in \overline{K}\right)\right)$$

The Ramadge-Wonham model of control is insufficient for several reasons. The only control mechanic available is the distinction between uncontrollable (Σ_u) and controllable ($\Sigma \setminus \Sigma_u$) events. There is no way to tell if a process is actually being actively controlled, only that it is *able* to be controlled. Experimental events are not asynchronous. Regulatory events are neither instantaneous, nor necessarily spontaneous. The Ramadge-Wonham model imposes a strict separation of the "controller" and the "plant model" which is not always applicable to biological systems. Regular languages are insufficient to express cyclic and chain reaction relationships. Finally, determining controllability for a hypothesis says nothing about the controllability of variants of that hypothesis. Accordingly, we extend the discrete event formalism to include a more physically based notion of control:

Definition 3.1 (Controlling Action). A **controlling action** A is a hypothesis event for which a neighborhood of acceptable observed behaviors N_a has been specified.

We note that controlling actions may be subdivided, according to $\mathcal{G}$, and may, in fact, constitute auxiliary forcing mechanisms for other events.

Definition 3.2 (Controlled). We say a biological system which has generated observational data $D(k)$ is **controlled** by a hypothetical control mechanism H if:

$$\exists H_i \in H, s.t. \forall e_j \in H_i, D(t') \subseteq N_a(e),$$

$$\tau(e_{j+1}) \geq t' \geq \tau(e_j),$$

and

$$D(t_0) \not\subseteq N_a(e), f.s.t_0 \leq \tau(e) \tag{5}$$

...where H_i are the paths through hypothesis H, and $\tau(e)$ is the time point associated with the occurrence of hypothesis event e.

A weaker version of this definition will also prove useful:

Definition 3.3 (De-Facto Controlled). We say a biological system which has generated observational data $D(t)$ is **de-facto controlled** by a hypothetical control mechanism H if:

$$\exists H_i \in H, s.t. \forall e \in H_i, D(t') \subseteq N_a(e),$$

$$f.s.t' \geq \tau(e) \tag{6}$$

...where H_i are the paths through hypothesis H, and $\tau(e)$ is the time point associated with the occurrence of hypothesis event e.

Each hypothesis defines a target neighborhood of events each time a "controlling" action appears in the hypothesis.

> (For example: "A induces B" may define a neighborhood of event streams as follows: All genes aside from A and B are free to exhibit any behaviors, and either: mRNA concentration increases for A and for B, mRNA concentration is stable for A and increases for B, or mRNA concentration decreases for A and B, for all microarray trials involving A and B. Behaviors where A decreases and B increases should not be in the neighborhood. Note that many, many other neighborhood specifications are possible, possibly involving specific numerical thresholds and time delays.)

Neighborhood boundary crossing from unexpected to expected behaviors generates a "controlled" judgment. Neighborhood boundary crossing from expected to unexpected behaviors generates an "uncontrolled" judgment, and cancels any prior "controlled" judgments. If the system event stream trace remains entirely within the target neighborhood at all times, "controlled" status is not contradicted, but neither is it supported. In this case, the judgment of "de-facto control" is applied. If the system event stream trace remains entirely without the target neighborhood at all times, "controlled" status is contradicted.

4. The Hypothesis Testing Process

Automata models and regular expressions are used to "fill in" any of the required events that the user who is composing the hypothesis has left out. This occurs when the user is issuing specifications at a high level of abstraction, is not well-versed in the specific system under study, or is forgetting something. The specification of this abstraction mechanic is inherited from the ontology.

Testing of hypotheses proceeds through several stages: First, the hypothesis is checked for proper grammatical structure. Next, all events the hypothesis requires (including abstracted events) are extracted. Then, all individual events along each critical path through the hypothesis are checked for contradictions. Next, controlling and controlled relationships are compared to the available time-sampled experimental data to check for contradictions. Then, support for all events along each path is assessed for each hypothesis that is not contradicted, and the viability partial order is established based upon the minimum support of all events on the maximally supported path (so that any chain of inferences is only as strong as its weakest link).

5. Future Work

We will expand upon the controlling action and controlled event definitions presented above to develop a general mathematical framework for the control of discrete event processes that includes concepts of (topologically) continuous mappings and control functions. We are writing the software tools to implement the approach outlined in this paper. We are designing a database and hypothesis composition and evaluation tools needed to apply our methodology to the galactose metabolic pathway in *Saccharomyces cerevisiae* [3]. We will then revise our formal language and use data accumulated for the TAIR project [6] to develop a hypothesis testing system for the stress response network of *Arabidopsis thaliana* (see http://www.arabidopsis.org).

6. Conclusions

We have developed a mathematical framework for the ranking of hypotheses about gene regulation. This framework has a unique combination of features which make it well-suited for modeling large, complex systems that involve many heterogeneous data sources.

We represent both data and hypotheses in an event-based formalism, enabling us to perform data fusion at the logical level. This allows us to easily extend our framework to incorporate new data types as they become available.

Our method is contradiction-based. Every piece of data rules out some portion of the space of all expressible hypotheses. In this way, we are guaranteed not to waste information, and are prevented from generating flawed models. This turns the information overflow into an advantage: the more data we have, the more structure we have for the hypothesis space and the tighter our bounds on the sets of allowable hypotheses.

Other Discrete Event frameworks mandate a strict separation between the controller and the plant model – a separation that is not often valid in biological systems, where structure and function are two sides of the same coin. We have defined controlling and controlled events to allow for pathway sharing and feedback regulation – features characteristic of biological networks that cannot be represented using standard Discrete Event models. We have developed an ontology and operator language in conjunction with our mathematical framework. Our hypothesis description language enables a seamless transition between representation and testing, and is has been designed to contain interoperable graphical and structured-text representations.

References

[1] R. B. Altman and S. Raychaudhuri. Whole-genome expression analysis: challenges beyond clustering. *Curr Opin Struct Biol*, 11(3):340–7, 2001. 0959-440x Journal Article Review Review, Tutorial.

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–9, 2000. 1061-4036 Journal Article.

[3] D. Lohr, P. Venkov, and J. Zlatanova. Transcriptional regulation in the yeast gal gene family: a complex genetic network.pg - 777-87. *Faseb J*, 9(9), 1995. 95324798 0892-6638 Journal Article Review Review, Tutorial.

[4] M. Peleg, I. Yeh, and R. B. Altman. Modelling biological processes using workflow and petri net models. *Bioinformatics*, 18(6):825–37, 2002.

[5] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of IEEE*, 77(1):81–98, 1989.

[6] S. Y. Rhee, W. Beavis, T. Z. Berardini, G. Chen, D. Dixon, A. Doyle, M. Garcia-Hernandez, E. Huala, G. Lander, M. Montoya, N. Miller, L. A. Mueller, S. Mundodi, L. Reiser, J. Tacklind, D. C. Weems, Y. Wu, I. Xu, D. Yoo, J. Yoon, and P. Zhang. The arabidopsis information resource (tair): a model organism database providing a centralized, curated gateway to arabidopsis biology, research materials and community. *Nucleic Acids Res*, 31(1):224–8, 2003. 22408340 1362-4962 Journal Article.

[7] B. M. R. Stadler and P. F. Stadler. Basic properties of filter convergence spaces. *J Chem Inf Comput Sci*, 42:577–585, 2002.

[8] R. Stevens. Tambis: transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16:184–5, 2000.

[9] J. Stoeckert, C. J. and H. Parkinson. The mged ontology: a framework for describing functional genomics experiments. *Comp Func Genomics*, 4:127–132, 2003.

[10] T. Sudkamp. *Languages and machines*. Addison-Wesley, Reading, 1988.

[11] O. Wolkenhauer. Systems biology: The reincarnation of systems theory applied in biology? *Briefings in Bioinformatics*, 2(3):258–270, 2001.