Analysing the microRNA-17-92/Myc/E2F/RB Compound Toggle Switch by Theorem Proving

Giampaolo Bella and Pietro Liò

@ Catania and Cambridge

Motivation

- Genetic networks stretch human intuition
  - Big problem: miRNA target prediction
  - How 'bout miRNA influence on network regulation
- Tools from
  - Computational analysis (differential eqs, …)
  - Computer-aided visualization (GNA, VisualGNA,…)
  - Formal/symbolic approaches
    - Pen-and-paper (Petri nets, Pi-calculus,…)
    - Computer-aided (model checking)
- Use theorem proving!

Model Checking vs Theorem Proving

- Pros
  - Press-button tool
  - May validate properties quickly
- Cons
  - Only handles small systems
  - State explosion

- Pros
  - Handles unbounded-size systems
  - Very expressive
- Cons
  - Needs human intervention
  - Steep learning curve

Predictive Modeling of Signaling Crosstalk during *C. elegans* Vulval Development

Jasmin Fisher[1*], Nir Piterman[1], Alex Hajnal[2], Thomas A. Henzinger[1,3]

1 School of Computer and Communication Sciences, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2 Institute of Zoology, University of Switzerland, 3 Electrical Engineering and Computer Science, University of California Berkeley, Berkeley, California, United States of America

BIOINFORMATICS

Model Checking Genetic Regulatory Networks with Parameter Uncertainty

Grégory Batt[1], Calin Belta[1], and Ron Weiss[2]

[1] Center for Information and Systems Engineering and Center for BioDynamics, Boston University, Brookline, MA, USA

[2] Department of Electrical Engineering and Department of Molecular Biology, Princeton University, Princeton, NJ, USA

batt@bu.edu, cbelta@bu.edu, rweiss@princeton.edu

Temporal logic patterns for querying dynamic models of cellular interaction networks

Pedro T. Monteiro[1,2,3], Delphine Ropers[1], Radu Mateescu[1], Ana T. Freitas[2] and Hidde de Jong[1,*]

Analysis of Genetic Regulatory Networks: A Model-Checking Approach

Grégory Batt[1*], Hidde de Jong[1], Johannes Geiselmann[2], and Michel Page[1,3]

BIOINFORMATICS   **ORIGINAL PAPER**

Model Checking Genetic Regulatory Networks using GNA and CADP

Robustness analysis and tuning of synthetic gene networks

Grégory Batt[1,*], Boyan Yordanov[2], Ron Weiss[3] and Calin Belta[1]

[1]Centers for Information and Systems Engineering and for BioDynamics, Boston University, Boston, MA, USA,
[2]Department of Biomedical Engineering, Boston University, Boston, MA, USA,

égory Batt, Damien Bergamini, Hidde de Jong, Hubert Garavel, and Radu Mateescu

BIOINFORMATICS

A general computational method for robustness ana with applications to synthetic gene networks

Aurélien Rizk, Gregory Batt, François Fages and Sylvain Soliman

INRIA Paris-Rocquencourt, 78153 Le Chesnay Cedex, France

Toward Integration of Systems Biology Formalism: The Gene Regulatory Networks Case

Raffaella Gentilini

gentilin@dimi.uniud.it

Università di Udine (DIMI), Via Le Scienze 206, 33100 Udine, Italy

The trail we've walked so far

- Analogies computer/genetic networks
- General model for gene/protein networks
- Formal analysis of the genetic toggle
- Extensions with miRNAs
- Towards analysis of compound toggle featuring miRNA-17-92

The main theorem

```
theorem no_instability: nt ∈ network ⟹
                ¬ (concentration A nt > threshold 2 A ∧
                   concentration B nt > threshold 2 B)
apply (erule network.induct)
apply (insert threshold_2A_pos)
apply auto
done
```

The complete guarantees

- *"Bistability: the toggle may only reach two stable states s.t. either protein prevails"*

   1. *BistabilityA*: protein A may exceed its second threshold while B doesn't exceed its own

   2. *BistabilityB*: protein B may exceed its second threshold while A doesn't exceed its own

   3. *NonInstability*: both proteins may not exceed their respective second thresholds at the same time

The microRNA-17-92/Myc/E2F/RB compund toggle switch

The tool: Isabelle

- Generic, interactive, user-driven proof-assist
- May use HOL as meta-language
- Its *simplifier* does conditional term rewriting
- Its *automatic provers* do classical reasoning
- Proof attempts may fail due to either insufficient skill or sound counterexamples

General model: basics

typedecl *gene*
typedecl *protein*
typedecl *mirna*

datatype
```
 event = Produces gene protein |
         Triggers protein gene |
         Inhibits protein gene |
         Degrades protein      |

         Expresses gene mirna   |
         Silences mirna gene
```

General model : initial polymerase

consts
  initialpolymerase :: int

axioms
  initialpolymerase_value [iff]:
    initialpolymerase > 0

General model : induced polymerase

```
consts
  inducedpolymerase :: gene ⇒ event list ⇒ int

primrec
  inducedpolymerase x [] = 0
  inducedpolymerase x (head#rest) = (case head of
      Produces y Y ⇒ inducedpolymerase x rest
    | Triggers Y y ⇒ if x=y then inducedpolymerase x rest + 1
                              else inducedpolymerase x rest
    | Inhibits Y y ⇒ if x=y then inducedpolymerase x rest - 1
                              else inducedpolymerase x rest
    | Degrades Y   ⇒ inducedpolymerase x rest
  )
```

General model : current polymerase

constdefs
currentpolymerase :: gene ⇒ event list ⇒ int
currentpolymerase x nt ==
  initialpolymerase + inducedpolymerase x nt

General model : protein concentration

```
consts
 concentration :: protein ⇒ event list ⇒ int
primrec
   concentration X [] = 0
   concentration X (head#rest) = (case head of
       Produces y Y ⇒ if X=Y then concentration X rest + 1
                                 else concentration X rest
     | Triggers Y y ⇒ concentration X rest
     | Inhibits Y y ⇒ concentration X rest
     | Degrades Y   ⇒ if X=Y then concentration X rest – 1
                                 else concentration X rest
   )
```

Verifying the general model

```
lemma inducedpolymerase_Produces:
  inducedpolymerase x (Produces x Y # nt) =
  inducedpolymerase x nt
apply simp
done
```

Verifying the general model

```
lemma inducedpolymerase_Produces:
  inducedpolymerase x (Produces x Y # nt) =
  inducedpolymerase x nt
apply simp
done


lemma currentpolymerase_Triggers:
  currentpolymerase x (Triggers Y y # nt) =
    (if x=y then currentpolymerase x nt + 1
            else currentpolymerase x nt)
apply (simp add: currentpolymerase_def)
done
```

Verifying the general model

```
lemma inducedpolymerase_Produces:
  inducedpolymerase x (Produces x Y # nt) =
  inducedpolymerase x nt
apply simp
done


lemma currentpolymerase_Triggers:
  currentpolymerase x (Triggers Y y # nt) =
    (if x=y then currentpolymerase x nt + 1
            else currentpolymerase x nt)
apply (simp add: currentpolymerase_def)
done


lemma concentration_Degrades:
 concentration X (Degrades Y # nt) =
   (if X=Y then concentration X nt - 1 else concentration X nt)
apply simp
done
```

Modelling the genetic toggle

- Model is set of all possible evolutions
- Each evolution is a list
- Model defined by structural induction

Modelling the genetic toggle

- Model is set of all possible evolutions
- Each evolution is a list
- Model defined by structural induction

- Qualitative model amenable to TP
- Birth and death factors abstracted away
- Inductive model subsumes stochastic one

A model toggle: base and production

inductive_set *comptoggle* :: *event list set* **where**

   base: *[]* ∈ *comptoggle*

| *mycpr:* ⟦ *ne1* ∈ *comptoggle; currentpolymerase myc ne1 > 0* ⟧
       ⟹ *Produces myc Myc # ne1* ∈ *comptoggle*

| *e2fpr:* ⟦ *ne2* ∈ *comptoggle; currentpolymerase e2f ne2 > 0* ⟧
       ⟹ *Produces e2f E2F # ne2* ∈ *comptoggle*

| *rbpr:*   ⟦ *ne3* ∈ *comptoggle; currentpolymerase rb ne3 > 0* ⟧
       ⟹ *Produces rb Rb # ne3* ∈ *comptoggle*

miRNA is expressed and silences…

$\mid$ $micrex:$ $\llbracket$ $ne4$ $\in$ $comptoggle;$ $currentpolymerase$ $microRNA1792$ $ne4 > 0$ $\rrbracket$
$\implies$ $Expresses$ $microRNA1792$ $miR1792$ $\#$ $ne4$ $\in$ $comptoggle$

$\mid$ $miRsi:$ $\llbracket$ $ne11$ $\in$ $comptoggle;$
$Expresses$ $microRNA1792$ $miR1792$ $\in$ set $ne11$ $\rrbracket$
$\implies$ $Silences$ $miR1792$ $e2f$ $\#$ $ne11$ $\in$ $comptoggle$

Myc regulates and degrades…

$$| \ \mathit{Myctr1:} [\![\ ne5 \ \in \ comptoggle; \ concentration \ Myc \ ne5 \ \geq \ threshold \ 1 \ Myc \ ]\!]$$
$$\Longrightarrow \ Degrades \ Myc \ \# \ Triggers \ Myc \ microRNA1792 \ \# \ ne5 \ \in \ comptoggle$$

Conclusions and developments

- TP opens up genetic networks to logicians
- Tool support scales up linearly to larger networks
- Toggle switch analysed, now completing holistic analysis of compound toggle
- More and more flavours of TP expected!