ModRev - Model Revision tool for Boolean logical models of biological regulatory networks

Filipe Gouveia, Inês Lynce, and Pedro T. Monteiro

{filipe.gouveia,ines.lynce,pedro.tiago.monteiro}@tecnico.ulisboa.pt

INESC-ID / Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal

CMSB 2020

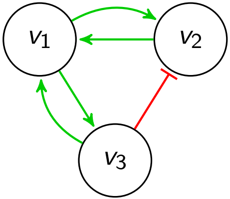
Table of Contents

Regulatory Networks

- Biological processes arise at the cellular level, governed by complex regulatory networks
- Regulatory network
  - Collection of molecular compounds (e.g. proteins, genes)
  - Compounds interact with each other
- Computational modelling allows
  - Functional understanding of the network
  - Test hypotheses
  - Identify predictions *in silico*
  - ...

Boolean Logical Model

- Different formalisms can be used [KS08]
  - We consider the Boolean logical formalism [Tho73].
- Compounds represented by a Boolean variable
  - active/inactive
- Interactions defined as positive (activation) or negative (inhibition)
- Regulations defined as Boolean functions



$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Motivation

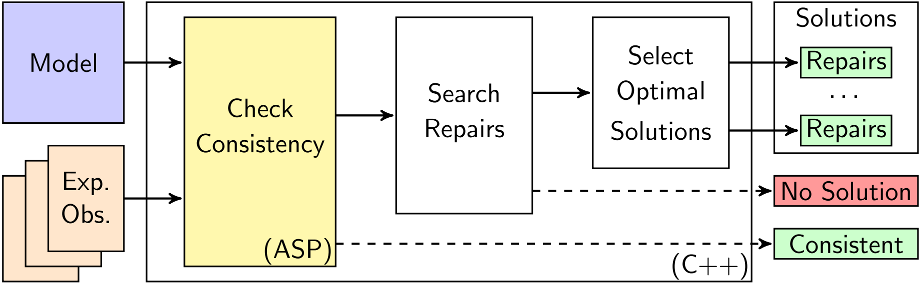
- As new experimental data becomes available,
  models may become inconsistent
  - Models may not be able to reproduce the new information
  - Models need to be revised
- Model Revision is mainly a manual task
  - Performed by a modeler
  - Prone to error
- How can we repair an inconsistent model?
  - Change a regulatory function?
  - Change the type of interaction?
  - Add or remove interactions?

Table of Contents

ModRev Tool

- ModRev is a freely available model revision tool
  - https://filipegouveia.github.io/ModelRevisionASP/
- Confronts a Boolean Logical Model with experimental observations
  - Stable state observations
  - Time-series observations
    - Synchronous
    - Asynchronous
- ModRev implements the following methods
  - Consistency check and reasons of inconsistency [GLM18]
  - Revision under stable state observations [GLM19]
  - Search for function repairs [GLM20a]
  - Revision under Time-series observations [GLM20b]

ModRev arquitecture



ModRev Tool

Repair Operations:

- Change regulatory Function
- Change interaction type
- Remove interaction
- Add interaction

ModRev Tool

Repair Operations:

- Change regulatory Function
- Change interaction type
- Remove interaction
- Add interaction

Optimization Criteria:

1. Minimize interaction addition/removal
2. Minimize interaction type changes
3. Minimize Boolean function changes

Table of Contents

Evaluation

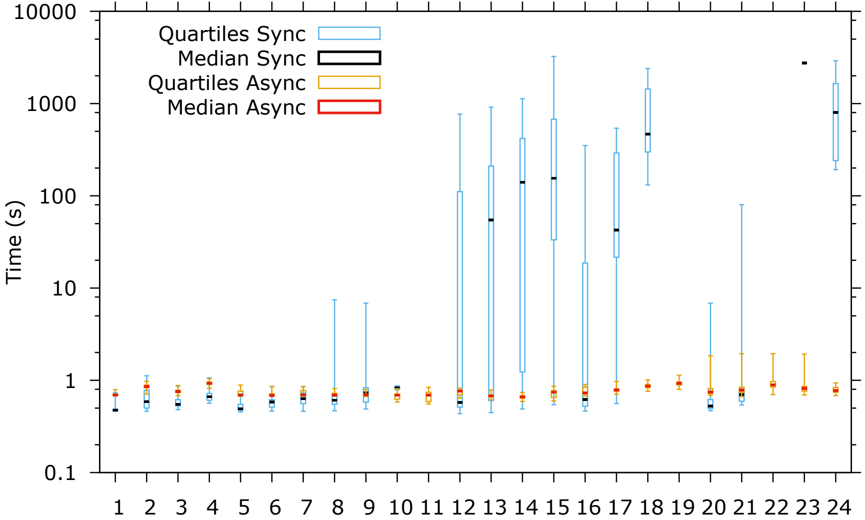
- Segment Polarity (SP) network [SCT02]

Conf.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
F	5	25	50	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	50	100	5	10
E	0	0	0	0	5	10	15	20	25	50	75	0	0	0	0	0	0	0	0	5	25	50	25	10
R	0	0	0	0	0	0	0	0	0	0	0	1	5	10	15	0	0	0	0	0	0	0	5	5
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	10	15	0	0	0	5	5

- Random changes were made according to probabilistic parameters
  - F% : Change a **F**unction
  - E% : Flip the sign of an **E**dge
  - R% : **R**emove an existing edge
  - A% : **A**dd a missing edge

- 100 corrupted models for each of the 24 configurations
- 5 time-series observations with 20 time-steps

Evaluation

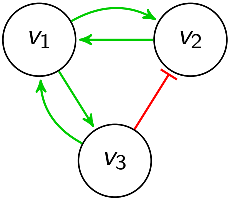

Quartiles Sync
Median Sync
Quartiles Async
Median Async
Time (s)
10000
1000
100
10
1
0.1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Evaluation

- ModRev repaired the model smaller # operations
- Models repaired mostly under 60 seconds
- Changing the topology of the network has the greatest impact
- Better performance under the asynchronous update scheme
  - Only on regulatory function is updated at each time step

Table of Contents

Tutorial



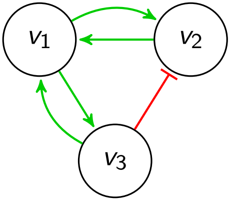

```
vertex(v1).
vertex(v2).
vertex(v3).
```

$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Tutorial



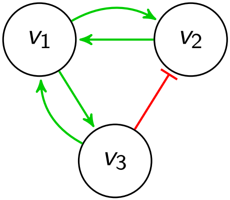

```
vertex(v1).
vertex(v2).
vertex(v3).

edge(v1,v2,1).
edge(v1,v3,1).
edge(v2,v1,1).
edge(v3,v1,1).
edge(v3,v2,0).
```

$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Tutorial



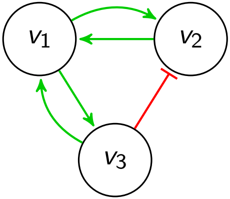
$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

```
vertex(v1).
vertex(v2).
vertex(v3).

edge(v1,v2,1).
edge(v1,v3,1).
edge(v2,v1,1).
edge(v3,v1,1).
edge(v3,v2,0).

functionOr(v1,1..1).
functionAnd(v1,1,v2). functionAnd(v1,1,v3).
functionOr(v2,1..2).
functionAnd(v2,1,v1). functionAnd(v2,2,v3).
functionOr(v3,1..1).
functionAnd(v3,1,v1).
```

Tutorial



Stable State observation

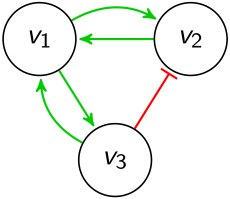
```
exp(p1).
obs_vlabel(p1,v1,0).
obs_vlabel(p1,v2,0).
obs_vlabel(p1,v3,1).
```

$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Tutorial



$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

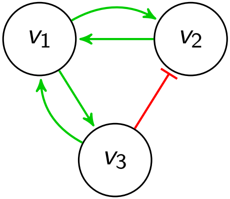
$$f_{v_3} = v_1$$

Stable State observation

```
exp(p1).
obs_vlabel(p1,v1,0).
obs_vlabel(p1,v2,0).
obs_vlabel(p1,v3,1).
```

```
$ ./modrev -m model.lp -obs obsSS.lp -ss
```

Tutorial



$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Stable State observation

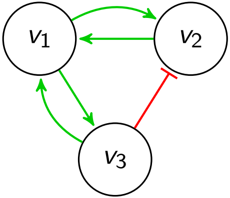
```
exp(p1).
obs_vlabel(p1,v1,0).
obs_vlabel(p1,v2,0).
obs_vlabel(p1,v3,1).
```

```
$ ./modrev -m model.lp -obs obsSS.lp -ss
```

```
### Found solution with 1 repair operation.
    Inconsistent node v3.
        Repair #1:
            Flip sign of edge (v1,v3).
```

Tutorial

Time-series observation



Node	Time 0	1	2
v_1	0	1	0
v_2	0	0	0
v_3	1	0	0

$$f_{v_1} = v_2 \wedge v_3$$

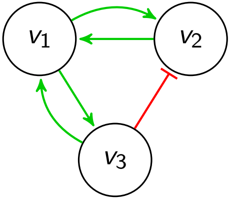
$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

```
#const t = 2.
exp(p2).
obs_vlabel(p2,0,v1,0). obs_vlabel(p2,0,v2,0).
obs_vlabel(p2,0,v3,1).
obs_vlabel(p2,1,v1,1). obs_vlabel(p2,1,v2,0).
obs_vlabel(p2,1,v3,0).
obs_vlabel(p2,2,v1,0). obs_vlabel(p2,2,v2,0).
obs_vlabel(p2,2,v3,0).
```

Tutorial

```
$ ./modrev -m model.lp -obs obsTS01.lp -up s
```



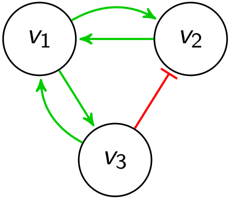
$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Tutorial

```
$ ./modrev -m model.lp -obs obsTS01.lp -up s
```



$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

```
### Found solution with 5 repair operations.
    Inconsistent node v1.
        Repair #1:
            Change function of v1 to (v2) || (v3)
   Inconsistent node v2.
        Repair #1:
            Change function of v2 to (v1 && v3)
            Flip sign of edge (v1,v2).
        Repair #2:
            Change function of v2 to (v1 && v3)
            Flip sign of edge (v3,v2).
   Inconsistent node v3.
        Repair #1:
            Change function of v3 to (v1 && v2)
            Add edge (v2,v3) with sign 1.
        Repair #2:
            Change function of v3 to (v1 && v3)
            Add edge (v3,v3) with sign 1.
```

Thank you!

ModRev https://filipegouveia.github.io/ModelRevisionASP/

Acknowledgements:

FCT Fundação para a Ciência e a Tecnologia

inesc id lisboa

TÉCNICO LISBOA

References

[GLM18] Filipe Gouveia, Inês Lynce, and Pedro T Monteiro. "Model Revision of Logical Regulatory Networks Using Logic-Based Tools". In: *ICLP 2018 (Technical Communications)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2018.

[GLM19] Filipe Gouveia, Inês Lynce, and Pedro T Monteiro. "Model Revision of Boolean Regulatory Networks at Stable State". In: *International Symposium on Bioinformatics Research and Applications*. Ed. by Zhipeng Cai, Pavel Skums, and Min Li. Springer International Publishing. 2019, pp. 100–112.

[GLM20a] Filipe Gouveia, Inês Lynce, and Pedro T. Monteiro. "Revision of Boolean Models of Regulatory Networks Using Stable State Observations". In: *Journal of Computational Biology* 27.2 (2020), pp. 144–155.

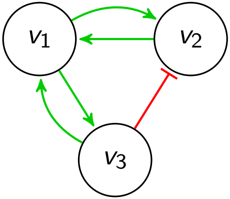
[GLM20b] Filipe Gouveia, Ines Lynce, and Pedro Tiago Monteiro. "Semi-automatic model revision of Boolean regulatory networks: confronting time-series observations with (a)synchronous dynamics". In: *bioRxiv preprint doi:10.1101/2020.05.10.086900* (2020).

[KS08] Guy Karlebach and Ron Shamir. "Modelling and analysis of gene regulatory networks". In: *Nature Reviews Molecular Cell Biology* 9.10 (2008), p. 770.

[SCT02] Lucas Sánchez, Claudine Chaouiya, and Denis Thieffry. "Segmenting the fly embryo: logical analysis of the role of the Segment Polarity cross-regulatory module". In: *Int. J. Dev. Biol.* 52.8 (2002), pp. 1059–1075.

[Tho73] René Thomas. "Boolean formalization of genetic control circuits". In: *J. Theor. Biol.* 42.3 (1973), pp. 563–585.

Tutorial



$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

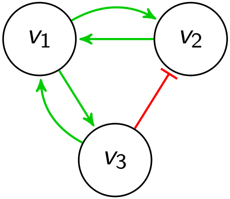
Incomplete time-series observation

Time

Node	0	1	2
v_1	0		1
v_2	1	0	0
v_3			

```
#const t = 2.
exp(p3).
obs_vlabel(p3,0,v1,0). obs_vlabel(p3,0,v2,1).
obs_vlabel(p3,1,v2,0).
obs_vlabel(p3,2,v1,1). obs_vlabel(p3,2,v2,0).
```

Tutorial



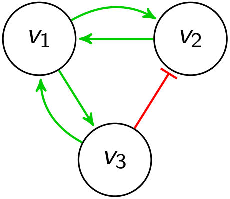

```
$ ./modrev -m model.lp -obs obsTS02.lp -up s
```

$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

Tutorial



$$f_{v_1} = v_2 \wedge v_3$$

$$f_{v_2} = v_1 \vee \neg v_3$$

$$f_{v_3} = v_1$$

```
$ ./modrev -m model.lp -obs obsTS02.lp -up s
```

```
### Found solution with 3 repair operations.
    Inconsistent node v1.
        Repair #1:
            Change function of v1 to (v2) || (v3)
            Flip sign of edge (v2,v1).
   Inconsistent node v2.
        Repair #1:
            Change function of v2 to (v1 && v3)
```

ModRev Input

- `vertex(V).` : `V` is a node of the network
- `edge(V1,V2,S).` : edge from `V1` to `V2` with sign `S` $\in \{0, 1\}$
- `functionOr(V,1..N).` : regulatory function of `V` in DNF is represented by a disjunction of `N` $\in \mathbb{N}$ terms
- `functionAnd(V,T,R).` : node `R` is present in the `T`-th term of the regulatory function of `V`

- `exp(E).` : `E` is an experimental observation
- `obs_vlabel(E,V,S).` : node `V` has an observed value of `S` $\in \{0, 1\}$ in experiment `E` (Stable State)
- `obs_vlabel(E,T,V,S).` : in time-step `T` node `V` has an observed value of `S` $\in \{0, 1\}$ in experiment `E` (Time-series)