

Exploring Life through Logic Programming

Alessandro Dal Palú
Andrea Formisano

Agostino Dovier
Enrico Pontelli

Dept. Computer Science, New Mexico State University, USA

University of Udine, Italy
December 2015

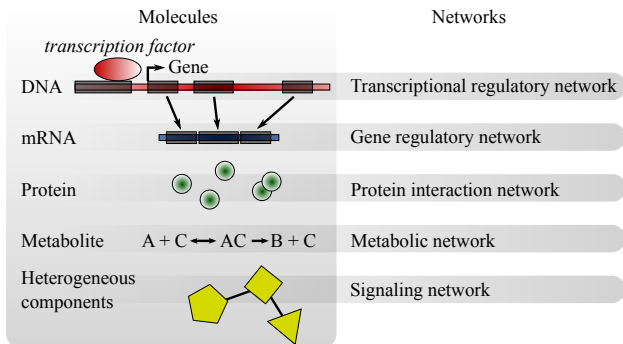
Systems Biology

Biological Networks

- A cell contains complex systems of interacting components
 - E.g. small molecules, DNA, proteins
- Interactions among these elements determine the cellular functions
- Each system can be modeled by means of **networks**
- Interactions can be analyzed at different levels:
 - *Transcriptional Regulatory Networks*: transcription factors control the transcription of genes in mRNA synthesis
 - *Gene Regulatory Networks*
 - *Protein Interaction Networks*: proteins interact and affect the products of the transcription step and form protein complexes
 - *Metabolic Networks*: biochemical reactions occur in the cell and involve metabolites (e.g., enzymes, substrates, etc.)
 - *Signaling Networks*: complex processes that took place in a cell in order to receive different signals from the cellular environment and from other cells.

Biological Networks

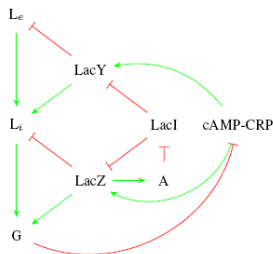
- The problem is to model a network from biological knowledge
- The model has to be validated w.r.t. experimental data
- Data is incomplete, sometimes unreliable
- Models need to be modified, repaired and/or extended
- Models can guide the design of new experiments



Influence Graph

Operon Lactose in *E. coli* (example from Gebser, Schaub, Thiele, Veber, 2011)

- Gene Regulatory Networks
 - Nodes are genes
 - Edges show how a gene influence other genes
 - Described by ordinary differential equations modeling relative quantities of constituents
- Simplest type: Influence Graphs
 - Edges restricted to positive (+) and negative (-) influences
 - Each node can be active or inhibited



Influence Graphs

- An **influence graph** is a directed graph $G = \langle N, E, \sigma \rangle$ s.t. $\sigma : E \rightarrow \{+, -\}$ is a labeling of the edges.
- σ can be partial. We consider it as total in this presentation.
- $i \longrightarrow j$ where $\sigma(i, j) = +$ means that i influences positively j (e.g., a positive (negative) variation of the level of i causes a positive (negative) variation of the level of j). It is often denoted as $i \rightarrow j$.
- $i \longrightarrow j$ where $\sigma(i, j) = -$ means that i influences negatively j (e.g., a positive (negative) variation of the level of i causes a negative (positive) variation of the level of j). It is often denoted as $i \dashrightarrow j$.

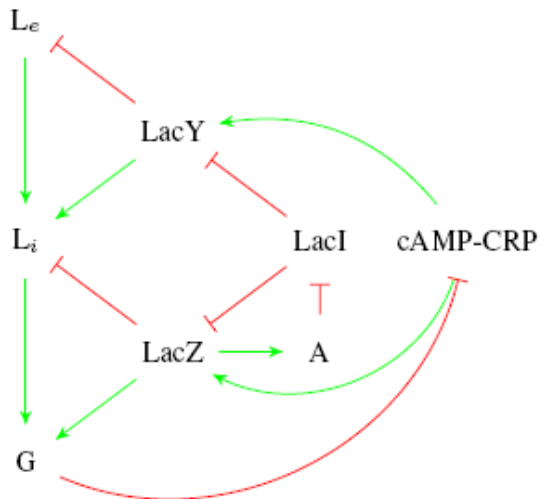
Influence Graphs

- Among the nodes there are **input** nodes, where we can increase or decrease the level of some substances
- From experimental results one builds a set of observations, namely, some **partial** assignments $\mu : N \rightarrow \{-, +\}$ for the “level” of the nodes.
- One of the first problems is understanding if these partial observations are “consistent” with the network
- $G = (N, E, \sigma)$ and μ *consistent*: there is a *total* extension μ' of μ (defined for all nodes in N) such that **for each non-input node** $n \in N$ there is an edge $(m, n) \in E$ such that

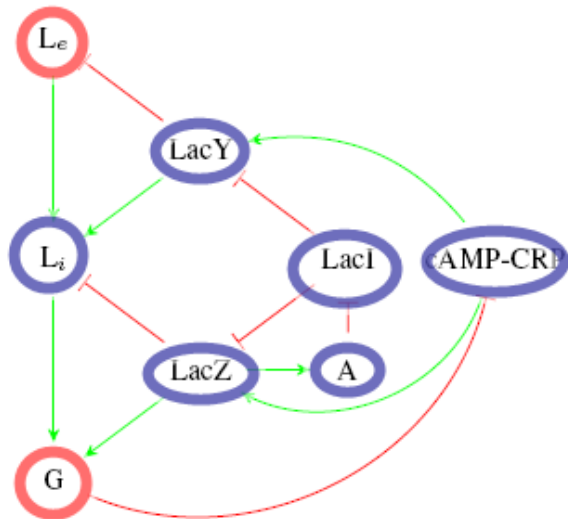
$$\mu'(m) \circ \sigma(m, n) = \mu'(n)$$

(i.e. $+ \circ + = - \circ - = +$, $+ \circ - = - \circ + = -$, using the rule of sign)

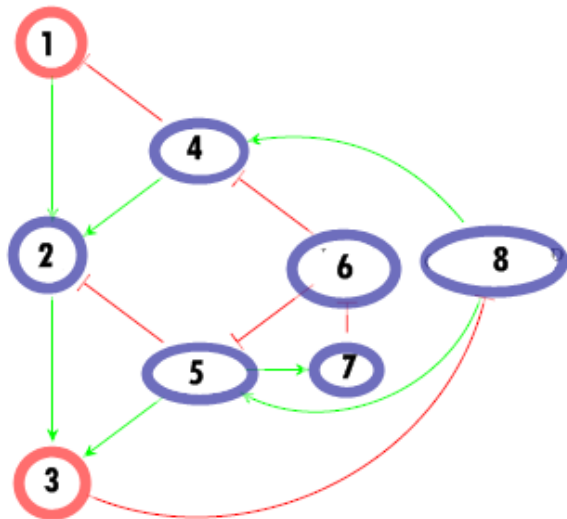
Operon Lactose in *E. coli*



Operon Lactose in *E. coli*

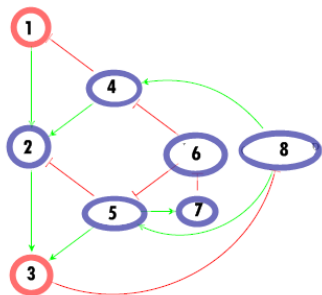


Operon Lactose in *E. coli*



Operon Lactose in *E. coli*

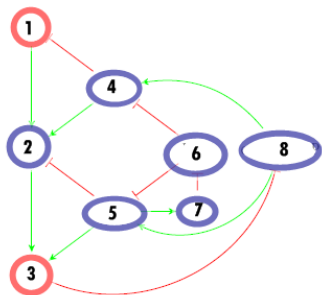
Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)

Operon Lactose in *E. coli*

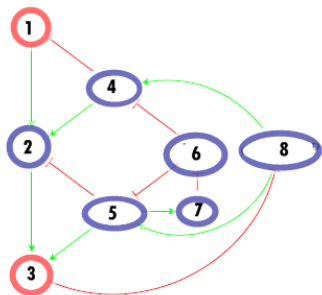
Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)
+	+	+	+	+	-	+	-	YES

Operon Lactose in *E. coli*

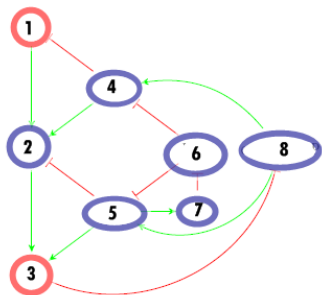
Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)
+	+	+	+	+	-	+	-	YES
+	-	-	?	?	?	?	+	SAT

Operon Lactose in *E. coli*

Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)
+	+	+	+	+	-	+	-	YES
+	-	-	?	?	?	?	+	SAT
+	-	-	-	-	+	-	+	YES
+	-	-	+	+	-	+	+	YES

Problem definition

Checking Consistency

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N , establish whether G and μ are consistent.

Problem definition

Checking Consistency

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N , establish whether G and μ are consistent.

If μ is total, it is just a polynomial check.

If μ is partial, it is NP-complete [Veber06]

Problem definition

Checking Consistency

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N , establish whether G and μ are consistent.

If μ is total, it is just a polynomial check.

If μ is partial, it is NP-complete [Veber06]

If inconsistent, try to **repair**:

- Add an edge
- Change the label of an edge
- Make a node an input
- Modify the data

We are interested in finding the minimal modifications on edges to make the network consistent.

Consistency Checking

ASP Model

- The input is given through facts:

```
vertex (Name) .
```

```
edges (Vertex1, Vertex2) .
```

```
input (Name) .
```

- The set of observation for nodes and edges is

```
observed (Name, Sign) .
```

```
observed (Vertex1, Vertex2, Sign) .
```

ASP Encoding

```
(1) sign(minus) .          sign(plus) .  
(2) opposite(minus,plus) . opposite(plus,minus) .  
(3) 1 {label_vertex(V,S) : sign(S)} 1 :- vertex(V) .  
(4) 1 {label_edge(U,V,S) : sign(S)} 1 :- edge(U,V) .
```

- We non-deterministically label each vertex with a sign
- We non-deterministically label each edge with a sign

ASP Encoding

- (1) `sign(minus).` `sign(plus).`
- (2) `opposite(minus,plus).` `opposite(plus,minus).`
- (3) `1 {label_vertex(V,S): sign(S)} 1 :- vertex(V).`
- (4) `1 {label_edge(U,V,S): sign(S)} 1 :- edge(U,V).`

- (5) `{wrong(U,V)} :- edge(U,V).`

- Each edge may be defined as *wrong*, i.e., not consistent

ASP Encoding

```
(1)  sign(minus).          sign(plus).
(2)  opposite(minus,plus). opposite(plus,minus).
(3)  1 {label_vertex(V,S): sign(S)} 1 :- vertex(V).
(4)  1 {label_edge(U,V,S): sign(S)} 1 :- edge(U,V).

(5)  {wrong(U,V)} :- edge(U,V).

(6)  label_vertex(V,S) :- observed(V,S).
```

- The labeled nodes are consistent with the input observations.

ASP Encoding

- ```

(1) sign(minus). sign(plus).
(2) opposite(minus,plus). opposite(plus,minus).
(3) 1 {label_vertex(V,S): sign(S)} 1 :- vertex(V).
(4) 1 {label_edge(U,V,S): sign(S)} 1 :- edge(U,V).

(5) {wrong(U,V)} :- edge(U,V).

(6) label_vertex(V,S) :- observed(V,S).

(7) label_edge(U,V,S) :- not wrong(U,V), observed(U,V,S).
(8) label_edge(U,V,S) :- wrong(U,V), observed(U,V,T),
 opposite(S,T).

```

- Labeled edges are consistent with observation, otherwise `wrong` is asserted

# ASP Encoding

```
(9) receive(V, plus) :- edge(U,V), sign(S),
 label_edge(U, V, S), label_vertex(U, S).
(10) receive(V, minus) :- edge(U,V), opposite(S,T),
 label_edge(U, V, S), label_vertex(U, T).
```

- A node receives a plus if there is an edge (U,V) and a node U with same label
- Otherwise the node receives a minus.

# ASP Encoding

```
(9) receive(V, plus) :- edge(U,V), sign(S),
 label_edge(U, V, S), label_vertex(U, S).
(10) receive(V, minus) :- edge(U,V), opposite(S,T),
 label_edge(U, V, S), label_vertex(U, T).
(11) :- label_vertex(V, S), not receive(V, S),
 not input(V).
```

- Constraint: all vertices (but input nodes) must be labeled and receiving a sign



# ASP Encoding

```
(9) receive(V, plus) :- edge(U,V), sign(S),
 label_edge(U, V, S), label_vertex(U, S).
(10) receive(V, minus) :- edge(U,V), opposite(S,T),
 label_edge(U, V, S), label_vertex(U, T).
(11) :- label_vertex(V, S), not receive(V, S),
 not input(V).
(12) edges_reversed(N) :- N = #count{ (U,V):wrong(U,V) }.
(13) #minimize { N:edges_reversed(N) }.
```

- Minimize the number of reversed edges to provide consistency

# Metabolic Networks

- Nodes are metabolites
- Edges represent reactions
- Intuition: reaction can only occur if all of its substrates are available as nutrients
- Construction:
  - Initial nutrient (seed)
  - Network expanded by adding all of the enabled reactions together with their products
  - Iterate until no further reactions are possible

# Model

- Metabolic Network: bipartite graph  $(R \cup M, E)$ ;  
 $E \subseteq (R \times M) \cup (M \times R)$
- $(m, r) \in E$ :  $m \in M$  is the **reactant** of the reaction  $r \in R$
- $(r, m) \in E$ :  $m \in M$  is the **product** of the reaction  $r \in R$
- $r \in R$  is **reachable** from  $S \subseteq M$  if all reactants of  $r$  appear in  $S$
- $m \in M$  is **reachable** from  $S \subseteq M$  if  $m \in S$  or  $m$  product of  $r$  reachable from  $S$
- the **scope** of  $S$  in  $G$  are the metabolites transitively reachable from  $S$

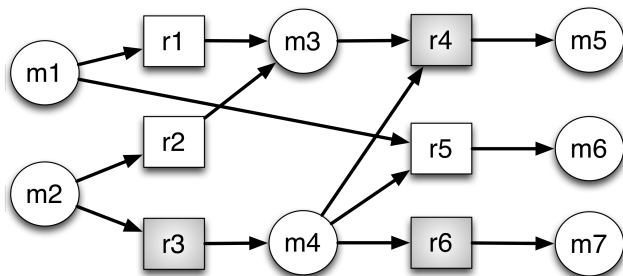
# Model: Metabolic network completion problem

- Let  $G' = (R' \cup M', E')$  be a **reference network**
- A **completion** of  $G$  from  $G'$  is  $R'' \subseteq R' \setminus R$  leading to  $((R \cup R'') \cup (M \cup M''), E \cup E'')$  where
  - $M'' = \{m \in M' \mid r \in R'', m \text{ is a product or reactant of } r\}$
  - $E'' = \{(m, r) \in E' \mid r \in R'', m \text{ reactant of } r\} \cup \{(r, m) \in E' \mid r \in R'', m \text{ is a product of } r\}$

# Model: Metabolic network completion problem

- Let  $G' = (R' \cup M', E')$  be a **reference network**
- A **completion** of  $G$  from  $G'$  is  $R'' \subseteq R' \setminus R$  leading to  $((R \cup R'') \cup (M \cup M''), E \cup E'')$  where
  - $M'' = \{m \in M' \mid r \in R'', m \text{ is a product or reactant of } r\}$
  - $E'' = \{(m, r) \in E' \mid r \in R'', m \text{ reactant of } r\} \cup \{(r, m) \in E' \mid r \in R'', m \text{ is a product of } r\}$
- Let  $S \subseteq M$  be the **seed** metabolites
- Let  $T \subseteq M$  be the **target** metabolites
- Find a completion of  $G$  from  $G'$  such that  $T$  is in the scope of  $S$  in the completion

# Example



- $R'' = \{r1\}$  is a completion for  $S = \{m1, m2\}$  and  $T = \{m5, m7\}$

# ASP Encoding

```
(1) draft(G) . %% original network G
(2) reaction(R,G) . %% reaction of G
(3) reaction(R,all) . %% reaction of reference G'
(4) reactant(M,R) . %% M reactant of R
(5) product(M,R) . %% M product of R
(6) seed(S) . target(T) . %% seeds and targets
```

- `draft(G)` identifies original network
- `all` denotes the reference network

# ASP Encoding

```
(1) draft(G). %% original network G
(2) reaction(R,G). %% reaction of G
(3) reaction(R,all). %% reaction of reference G'
(4) reactant(M,R). %% M reactant of R
(5) product(M,R). %% M product of R
(6) seed(S). target(T). %% seeds and targets

(7) type(all).
(8) type(x).
(9) type(N) :- draft(G).
```

- type  $x$  denotes  $R'$



# ASP Encoding

```
(10) reaction(R, x) :- reaction(R, G), draft(G).
```

```
(11) {reaction(R, x)} :- reaction(R, all).
```

- All reactions of  $G$  are in the completion
- All other reactions from  $G'$  might be added

# ASP Encoding

```
(10) reaction(R, x) :- reaction(R, G), draft(G).
(11) {reaction(R, x)} :- reaction(R, all).

(12) scope(M, T) :- seed(M), type(T).
(13) scope(M, T) :- type(T), product(M, R), reaction(R, T),
 scope(M2, T) : reactant(M2, R).
(14) :- target(M), not scope(M, x).
```

- `scope` designs reachability for type `T`
- Rule (14) assess that the target is reachable in `G`"

# ASP Encoding

```
(15) new(R) :- reaction(R,x), draft(Net),
 not reaction(R,Net).
```

- Determine all reactions that have been added

# ASP Encoding

```
(15) new(R) :- reaction(R,x), draft(Net),
 not reaction(R,Net).
```

```
(16) reactions(S) :- S = #count{ R:new(R) }.
```

```
(17) #minimize{S:reactions(S)}.
```

- Request the smallest number of additions to complete the graph

## Some references

- Siegel A., et al. 2006. Qualitative analysis of the relation between DNA microarray data and behavioral models of regulation networks. *Biosystems* 84, 2, 153–174.
- Guziolowski C. et al. 2009. Bioquali cytoscape plugin: analysing the global consistency of regulatory networks. *BMC Genomics*, 10.
- Corblin F. et al. 2009. A declarative constraint-based method for analyzing discrete genetic regulatory networks. *Biosystems* , 98(2):91-104. [Also in WCB05]
- Gebser, Schaub, Thiele, Veber. 2011 *Detecting Inconsistencies in Large Biological Networks with Answer Set Programming*. TPLP (2–3):323–360, 2011. [Also in WCB08]
- Guerra and Lynce. *Reasoning over Biological Networks using Maximum Satisfiability*. Proc. of CP2012.
- P. Veber, M. Le Borgne, A. Siegel, S. Lagarrigue, and O. Radulescu. Complex qualitative models in biology: A new approach. *Complexus*, 2(3-4):140–151, 2006.

# Acknowledgments

Thank you!

- School organizer: Agostino Dovier
- School participants (you)
- My main collaborators/co-authors/friends in Bioinformatics:  
Alessandro Dal Palú, Agostino Dovier, Federico Fogolari, Andrea Formisano, Ferdinando Fioretto, Federico Campeotto