Reasoning over Biological Networks using Maximum Satisfiability

João Guerra and Inês Lynce

INESC-ID/Instituto Superior Técnico,
Technical University of Lisbon,
Portugal

CP 2012, Québec

Current State of Systems Biology

- High-throughput methods
  - Large sets of comprehensive data
- Models are incomplete
- Data is inconsistent
- Aberrant measurements

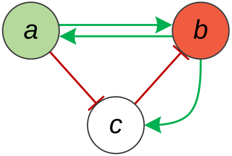
- We propose a SAT-based framework to
  - Detect inconsistencies
  - Repair inconsistencies
  - Predict unobserved variations

Outline

1. Modelling
   - Influence Graphs
   - Sign Consistency Model
   - Maximum Satisfiability
2. Reasoning
   - Checking Consistency
   - Repairing
   - Predicting
3. Experimental Evaluation
   - Setup
   - Results
4. Concluding Remarks

Influence Graphs

- Biological networks are represented by influence graphs
- An influence graph is a directed graph $G = (V, E, \sigma)$
  - V is a set of vertices representing the genes
  - E is a set of edges representing the interactions between the genes
  - $\sigma : E \to \{+, -\}$ is a (partial) labelling of the edges
- An experimental profile $\mu : V \to \{+, -\}$ is a (partial) labelling of the vertices
  - Each vertex is also classified as input or non-input



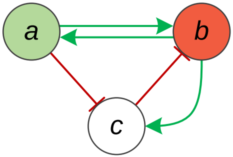
$$\sigma = \left\{ \begin{array}{ll} a \to b = +, & a \to c = -, \\ b \to a = +, & b \to c = +, \\ c \to b = - & \end{array} \right\}$$

$$\mu = \{a = +, \quad b = -\}$$

Sign Consistency Model

- The labelling $\mu(v)$ of a non-input vertex v is consistent if
  - There is at least one influence that explains its sign
  - One edge $u \rightarrow v$ such that $\mu(u) \cdot \sigma(u \rightarrow v) = \mu(v)$
- An influence graph $G = (V, E, \sigma)$ and an experimental profile μ are mutually consistent if
  - There are total labellings σ' and μ' (total extensions of σ and μ)
  - Such that $\mu'(v)$ is consistent for every non-input vertex v

Example



$$\sigma = \left\{ \begin{array}{ll} a \to b = +, & a \to c = -, \\ b \to a = +, & b \to c = +, \\ c \to b = - & \end{array} \right\}$$

$$\mu = \{a = +, \quad b = -\}$$

- The graph and profile are inconsistent
  - $\mu(a) = +$ while $\mu(b) \cdot \sigma(b \to a) = -$
- Why?
  - Incomplete model
  - Aberrant measurements
- Repairing (restoring consistency)
  - $\mu(a) = -$ or $\mu(b) = +$ (cardinality-minimal repairs)
  - Make a and b inputs (subset-minimal repair)

Maximum Satisfiability

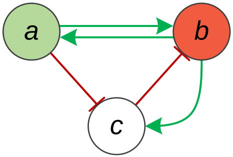
- Boolean Satisfiability (SAT)
  - Given a propositional formula φ, find an assignment to the variables that satisfies all clauses in φ
- Maximum Satisfiability (MaxSAT)
  - Optimization version of SAT
  - Find an assignment that maximizes (minimizes) the number of satisfied (unsatisfied) clauses
- Partial MaxSAT
  - Given a propositional formula $\varphi = \varphi_h \bigcup \varphi_s$, find an assignment to the variables that satisfies all hard clauses (φ_h) and the maximum number of soft clauses (φ_s)

Outline

Checking Consistency

- SAT solution for checking consistency
- 4 types of variables
  - vertices ($lvtx_v$) – 1 unit clause for each vertex with known label (μ)
  - inputs (inp_v) – 1 unit clause for each vertex
  - edges ($ledg_{uv}$) – 1 unit clause for each edge with known label (σ)
  - influences ($infl_{uv}$) – 2 constraints for each influence
- Ensuring consistency
  - 2 constraints for each vertex
- SAT call reveals whether the graph and profile are mutually consistent or not

Example



$$\sigma = \left\{ \begin{array}{ll} a \rightarrow b = +, & a \rightarrow c = -, \\ b \rightarrow a = +, & b \rightarrow c = +, \\ c \rightarrow b = - & \end{array} \right\}$$

$$\mu = \{a = +, \quad b = -\}$$

$lvtx_a \quad \neg lvtx_b \quad$ (no unit clause for vertex c)

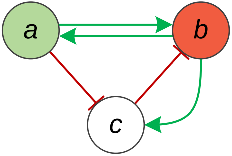
$\neg inp_a \quad \neg inp_b \quad \neg inp_c$

$ledg_{ab} \quad \neg ledg_{ac} \quad ledg_{ba} \quad ledg_{bc} \quad \neg ledg_{cb}$

$$infl_{ba} \longrightarrow (lvtx_b \wedge ledg_{ba}) \vee (\neg lvtx_b \wedge \neg ledg_{ba})$$
$$\neg infl_{ba} \longrightarrow (lvtx_b \wedge \neg ledg_{ba}) \vee (\neg lvtx_b \wedge ledg_{ba})$$

$$inp_a \vee (lvtx_a \longrightarrow infl_{ba})$$
$$inp_a \vee (\neg lvtx_a \longrightarrow \neg infl_{ba})$$

Example



$$\sigma = \left\{ \begin{array}{ll} a \to b = +, & a \to c = -, \\ b \to a = +, & b \to c = +, \\ c \to b = - & \end{array} \right\}$$

$$\mu = \{a = +, \quad b = -\}$$

$lvtx_a \quad \neg lvtx_b \quad$ (no unit clause for vertex c)

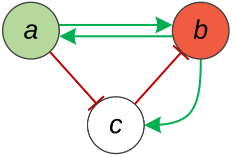
$$\neg inp_a \quad \neg inp_b \quad \neg inp_c$$

$$ledg_{ab} \quad \neg ledg_{ac} \quad ledg_{ba} \quad ledg_{bc} \quad \neg ledg_{cb}$$

$$infl_{ba} \longrightarrow (lvtx_b \wedge ledg_{ba}) \vee (\neg lvtx_b \wedge \neg ledg_{ba})$$
$$\neg infl_{ba} \longrightarrow (lvtx_b \wedge \neg ledg_{ba}) \vee (\neg lvtx_b \wedge ledg_{ba})$$

$$inp_a \vee (lvtx_a \longrightarrow infl_{ba})$$
$$inp_a \vee (\neg lvtx_a \longrightarrow \neg infl_{ba})$$

Example



$$\sigma = \begin{Bmatrix} a \rightarrow b = +, & a \rightarrow c = -, \\ b \rightarrow a = +, & b \rightarrow c = +, \\ c \rightarrow b = - & \end{Bmatrix}$$

$$\mu = \{a = +, \quad b = -\}$$

$lvtx_a \quad \neg lvtx_b \quad$ (no unit clause for vertex c)

$$\neg inp_a \quad \neg inp_b \quad \neg inp_c$$

$$ledg_{ab} \quad \neg ledg_{ac} \quad ledg_{ba} \quad ledg_{bc} \quad \neg ledg_{cb}$$

$$infl_{ba} \longrightarrow (lvtx_b \wedge ledg_{ba}) \vee (\neg lvtx_b \wedge \neg ledg_{ba})$$
$$\neg infl_{ba} \longrightarrow (lvtx_b \wedge \neg ledg_{ba}) \vee (\neg lvtx_b \wedge ledg_{ba})$$

$$inp_a \vee (lvtx_a \longrightarrow infl_{ba})$$
$$inp_a \vee (\neg lvtx_a \longrightarrow \neg infl_{ba})$$

Repairing

- Partial MaxSAT solution for repairing
- Only cardinality-minimal repairs
- 3 types of repair operations
  - flip vertices signs
  - make vertices inputs
  - flip edges signs
- Converting encoding into MaxSAT
  - Clauses corresponding to what we are repairing are made soft (only unit clauses)
  - The remaining clauses are hard
- MaxSAT call identifies the set of repairs (unsatisfied clauses)

Prediction

- What is common to all (optimal) solutions
- Backbone of the formula
- Intersection of all repairs (predicting under inconsistency)
  - Enumeration (feedback loop)
  - Only 1 blocking clause (the current prediction)
  - Only a subset of the variables is relevant

Predicting under Inconsistency

Input: Partial MaxSAT Formula $\mathcal{F}$
Output: Predicted Repairs of $\mathcal{F}$, *prediction*

$(out, opt, sol) \leftarrow \texttt{MaxSAT}(\mathcal{F})$ // compute initial solution
$optimum \leftarrow opt$
$prediction \leftarrow \texttt{Get-Repairs}(sol)$

while $|prediction| \neq 0$ **do**
  $(out, opt, sol) \leftarrow \texttt{MaxSAT}(\mathcal{F} \cup [\neg prediction])$ // block current prediction
  if $out == \texttt{UNSAT}$ **or** $opt > optimum$ **then**
    break
  $prediction \leftarrow prediction \cap \texttt{Get-Repairs}(sol)$ // update prediction

return *prediction*

- Either the prediction is reduced or the algorithm terminates
- At most n iterations (n = number of repair operations = *optimum*)

Outline

Setup

- SAT/MaxSAT vs ASP (Gebser et al. 2010, 2011)
- Instances
  - Randomly generated
  - GRN of *E. coli* along with 2 experimental profiles
- Timeout: 600 seconds
- Intel Xeon 5160 (3.00 GHz, 4 GB)
- ASP: *clasp*, *gringo*
- SAT: MiniSat, `minibones`
- MaxSAT: MSUnCore

Results

Consistency Checking, Predicting under Consistency

- SAT vs ASP
- Trivial for both approaches

Repairing, Predicting under Inconsistency

- MaxSAT vs ASP
- ASP could not solve the hardest instances

		Solved (%)	*Time*
Repair	ASP	2448 (87)	20471
	MaxSAT	2814 (100)	994
Predict	ASP	2440 (87)	14181
	MaxSAT	2814 (100)	8422

Concluding Remarks

- New SAT/MaxSAT framework for reasoning over biological networks
- SAT/MaxSAT approach more competitive than ASP approach
- Future
  - Minimal inconsistent cores (MICs)
  - More types of repair operations (e.g. add edges)
  - Subset-minimal repairs
  - Improve prediction under inconsistency

Questions?