

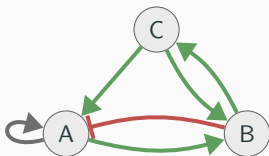
Inference of Qualitative Models from Steady-State Data via Weighted MaxSMT

Ondřej Huvar Nikola Beneš Martin Jonáš David Šafránek **Samuel Pastva**

CMSB 2026

Masaryk University, Brno, Czechia

Boolean networks (just in case)



influence graph

- activation
- inhibition
- monotonicity unspecified

- Variables A, B, C with **Boolean** domains $\{0, 1\}$
- Each variable governed by an **update function**:

$$F_A(a, b, c) = (a \vee c) \wedge \neg b$$

$$F_B(a, c) = a \vee c$$

$$F_C(b) = b$$

- State x is a **fixed point** if $F_i(x) = x[i]$ for all i

This model has two fixed points:

$(0, 0, 0)$ and $(0, 1, 1)$ e.g. *two cell-type phenotypes*

Everything in this talk generalizes to **multi-valued** domains too — Boolean here for clarity.

The inference problem

Given

- **prior-knowledge influence graph:**
who regulates whom (+ monotonicity, essentiality)
- **steady-state observations:**
expression measurements of (putative) phenotypes

Find

- update functions consistent with the influence graph
- ... such that the observed states are (near) fixed points of the model

Why exact methods (SAT/SMT/ASP/BDD)?

Guarantees: every returned model provably satisfies the *entire* specification.

Inference is fragile

Exact inference answers a yes/no question: *does a model of the specification exist?*

- Real biological data is **noisy**: measurement errors propagate into the specification
- Conflicting observations $\Rightarrow$ specification **UNSAT**
- UNSAT result is hard to reconcile:
 - no model at all — nothing to analyse
 - design flaw vs. technical measurement error? Hard to tell!
- Consequence: **under-specification** — we often drop observations out of fear of UNSAT

Goal: Always return “the most viable” model, using conflicts to explain UNSAT results.

Weighted MaxSMT

- **Hard constraints:** must hold in every solution
- **Soft constraints:** each carries a **weight** (confidence);
- Goal: maximize the total weight of satisfied soft constraints.

Example

Hard: $x + y = z, y \geq 10$ Soft: $x \geq 0$ ($w = 8$), $z \leq 0$ ($w = 3$), $x + z \geq 40$ ($w = 6$)

Optimum: $x = 20, y = 10, z = 30$ with score $8 + 6 = 14$ ($z \leq 0$ sacrificed; cheapest conflict)

- **Priority levels** $\mathcal{C}_1, \mathcal{C}_2, \dots$: scores maximized **lexicographically**
level-1 score matters more than everything below it, etc.
- **Anytime:** solvers report the best model so far + a bound on the gap to optimum
stop early $\Rightarrow$ still a valid model, with certified suboptimality — *foreshadowing*...

Weighted model inference — problem statement

Hard (structure, prior knowledge):

- **Influence graph**: update function of v_i may only depend on its regulators
- **Essentiality**: an essential regulator must observably influence its target
- **Monotonicity**: regulators provably act as activators / inhibitors

Hard or soft (data):

- m **fixed-point states** $s^1, \dots, s^m$ must exist (optionally pairwise distinct)
- **Observation constraints** $s_i^j \bowtie k$ or $s_i^j \bowtie s_{i'}^{j'}$, $\bowtie \in \{=, \neq, \leq, \geq, <, >\}$
- weights = confidence; optionally partitioned into priority levels $\mathcal{C}_1, \dots, \mathcal{C}_l$

Solution

Update functions + fixed-point valuations satisfying all hard constraints, lexicographically maximising the weight of satisfied soft constraints.

SMT encoding: uninterpreted functions

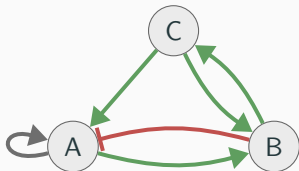
- One **uninterpreted function** f_i per variable v_i (arity = number of regulators)
- Fixed-point state variables s_i^j become integer (or Boolean) SMT variables

Four families of constraints:

1. **Fixed points**: $\sigma_s \equiv \bigwedge_i s_i = f_i(s_{r_1}, \dots, s_{r_k})$
2. **Essentiality**: $\eta \equiv \exists a, b, e_2, \dots, e_k. f_j(a, e_2, \dots, e_k) \neq f_j(b, e_2, \dots, e_k)$
3. **Monotonicity**: wait for it...
4. **Observations**: soft/hard assertions over the s_i^j with weights

Optimal SMT model $\equiv$ optimal inferred network.

Monotonicity: quantification



F_A : monotone in c , anti-monotone in b , free in a :

$$\forall (a, b, c), (a', b', c'). (a = a' \wedge b \geq b' \wedge c \leq c') \\ \Rightarrow F_A(a, b, c) \leq F_A(a', b', c')$$

In general, with positive $\mathcal{P}$, negative $\mathcal{N}$, and free inputs $\mathcal{R}$:

$$\forall \bar{x}, \bar{y}. \left(\bigwedge_{i \in \mathcal{P}} x_i \leq y_i \wedge \bigwedge_{i \in \mathcal{N}} y_i \leq x_i \wedge \bigwedge_{i \in \mathcal{R}} x_i = y_i \right) \Rightarrow g(\bar{x}) \leq g(\bar{y})$$

Problem

universal quantifiers + uninterpreted functions = slow

Solving monotonicity with quantifier instantiation

Theorem [Sofronie-Stokkermans '07; evaluated in our SAT '26 paper]

Formula φ is satisfiable with monotone f iff the following is satisfiable (ψ is the monotonicity “formula template” without quantifiers):

$$\varphi \wedge \bigwedge_{f(\bar{t}), f(\bar{s}) \in \text{terms}(\varphi)} \psi(f, \bar{t}, \bar{s})$$

No quantifiers needed: instantiate the template only for *pairs of function terms that already appear in the formula*. Quadratic in the number of terms, but lemmas can be added lazily.

Example: Fixed points (0,0,0) and (1,1,1) put exactly two F_B -terms in the formula: $F_B(0,0)$, $F_B(1,1)$ (F_B is an activator in both arguments). **Two monotonicity lemmas:**

$$\begin{aligned} \psi(F_B, (0,0), (1,1)) &\equiv (0 \leq 1 \wedge 0 \leq 1) \Rightarrow F_B(0,0) \leq F_B(1,1) && \leftarrow \text{active} \\ \psi(F_B, (1,1), (0,0)) &\equiv (1 \leq 0 \wedge 1 \leq 0) \Rightarrow F_B(1,1) \leq F_B(0,0) && \text{vacuous} \end{aligned}$$

Note: For soft constraints, fixed points are variables $\Rightarrow$ cannot easily simplify lemmas...

Where do observation constraints come from?

Two flavours of steady-state observations

You are in charge!

Disclaimer: The method is general — we don't dictate how constraints should be built. However, there are some patterns and assumptions to keep in mind...

Value-based (level constraints)

$$s_i^j = k$$

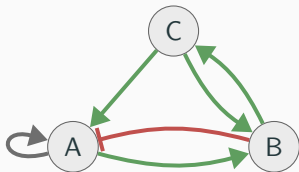
- From **discretisation** of expression data (scBoolSeq, BiTrinA, ...)
- Weight = discretisation confidence
- Commits to absolute activity levels

Comparison-based (ordering constraints)

$$s_i^a \bowtie s_i^b$$

- From **differential expression** (or similar) between cell types
- Weight = confidence in the ordering
- No explicit discretization threshold

Value-based constraints: toy example, Scenario I



two fixed points s^1, s^2 (distinct)

12 soft constraints $s_i^j = k$, weight = confidence (bold = most confident per variable):

value	s_A^1	s_B^1	s_C^1	s_A^2	s_B^2	s_C^2
0	0.74	0.65	0.53	0.07	0.57	0.20
1	0.26	0.35	0.47	0.93	0.43	0.80

- most confident choice: $s^1 = (0, 0, 0)$, $s^2 = (1, 0, 1)$, weight 4.22 ...
- ... but **infeasible**: essentiality + activation force $F_C = id$, so **$B = C$ in every steady state** of every consistent model

Optimal solution (weight 4.08 out of 6.00):

$$s^1 = (0, 0, 0), \quad s^2 = (1, 1, 1)$$

Comparison-based constraints: ordering without thresholds

For gene i and cell types a, b : three **mutually exclusive** soft constraints

$$s_i^a < s_i^b \quad s_i^a = s_i^b \quad s_i^a > s_i^b$$

- statistical model (e.g. scVI on scRNA-seq) estimates $\xi = P(\text{expression of } i \text{ lower in } a \text{ than in } b)$
- weights are functions of ξ — but **how exactly** is a modelling decision
- no discretisation threshold θ anywhere

Two weighting schemes (next slide) differ in one thing only:

is equality or inequality the default?

Equality-preferring vs. inequality-preferring weights

Cautious (equality is the default):

$$w(<) = \xi, \quad w(=) = 1, \quad w(>) = 1 - \xi$$

- equality **always** wins locally ($1 \geq \xi$)
- strict inequalities appear only when **forced** by higher-priority constraints
- quantitative change $\nrightarrow$ qualitative change

Committed (sufficient difference wins):

$$w(<) = \xi^2, \quad w(=) = 2\xi(1 - \xi), \quad w(>) = (1 - \xi)^2$$

- equality optimal only for $\xi \in (\frac{1}{3}, \frac{2}{3})$
- assumes: quantitative change $\Rightarrow$ qualitative change
- *which* level is chosen stays flexible

	$\xi = 0.9$			$\xi = 0.5$		
	<	=	>	<	=	>
cautious	0.90	1.00	0.10	0.50	1.00	0.50
committed	0.81	0.18	0.01	0.25	0.50	0.25

The “safest” model of a dataset with no additional assumptions is an **empty model**.

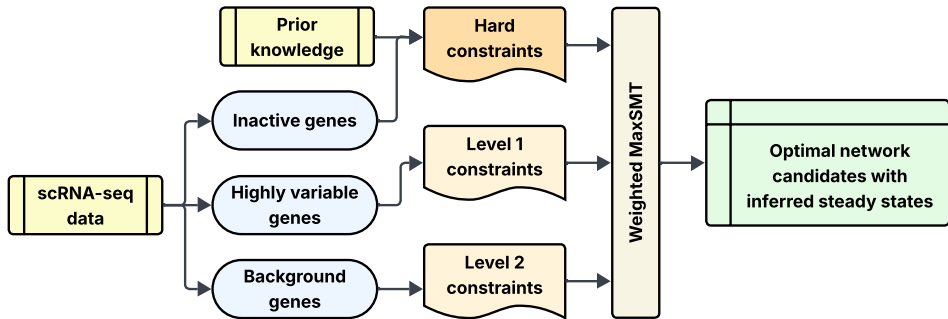
Every strict inequality in your specification is a modelling assumption —
e.g. *“assume the model should explain all highly variable genes.”*

Where the assumptions live in our workflow

- **Cautious** scheme, level $\mathcal{C}_2$: background genes
 - ⇒ without higher-priority pressure, constant update functions win
 - ⇒ the inferred model's inequalities trace back to **explicit** assumptions
- **Committed** scheme, level $\mathcal{C}_1$: phenotype-defining genes (markers, highly variable genes)
 - ⇒ the assumption *"the model should explain HVG differences"* is ours, stated once, applied uniformly
- Choice of which genes are phenotype-defining remains the modeller's responsibility
an (unavoidable) part of qualitative modelling: deciding which qualitative differences the model must explain

Workflow and Evaluation

From data to models: the evaluated workflow



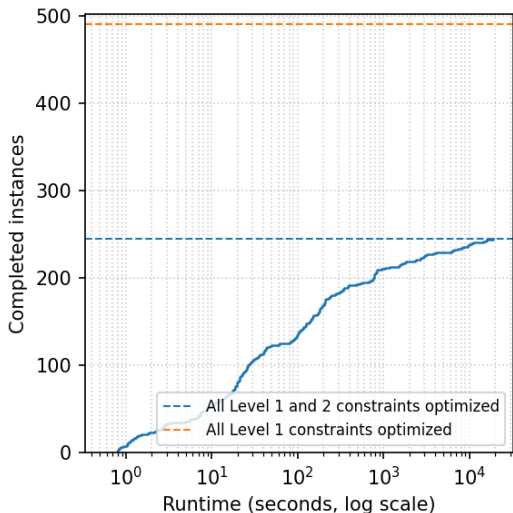
- implemented in a Rust tool (+ Z3); influence graphs in .aeon format

Benchmarks: neural differentiation at scale

- scRNA-seq atlas of mouse cerebral cortex + Omnipath prior-knowledge network
- **502 instances**: all combinations of 2–9 cell types; 6h timeout

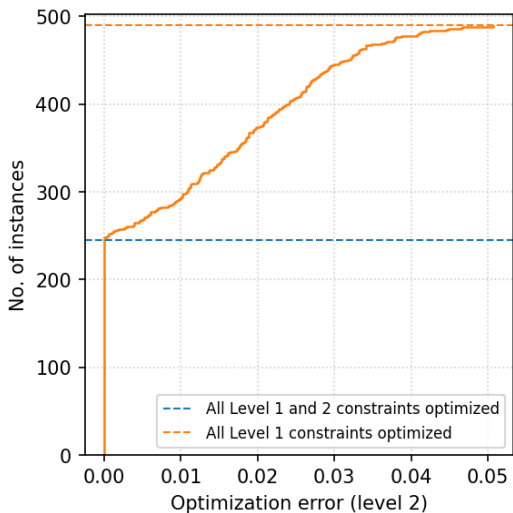
cell types	instances	genes	soft constraints	$\mathcal{C}_1$ optimal	$(\mathcal{C}_1, \mathcal{C}_2)$ optimal
2	36	183–612	362–869	36	36
3	84	211–865	1,255–4,326	84	84
4	126	280–1,004	3,278–10,676	126	89
5	126	349–1,136	6,743–20,672	126	28
6	84	409–1,225	11,847–34,158	84	8
7	36	524–1,275	21,110–50,249	32	0
8	9	955–1,307	51,108–69,238	2	0
9	1	1,326	90,964	0	0

Results: optimisation *converged*



- 245/502 instances solved to full optimality within 6h
- 490/502: level $\mathcal{C}_1$ fully optimised (phenotype-defining genes explained provably optimally)
- fully optimal up to 6 cell types; $\mathcal{C}_1$ -optimal up to 8

Results: optimisation *did not converge* — and that's OK



- for the remaining instances, the solver still returns a **guaranteed feasible model** + a bound
 - all 490 within **5 %** of the total $\mathcal{C}_2$ weight
 - MaxSMT **never answers UNSAT**: conflicting soft constraints are dropped, cheapest first
- ⇒ violated constraints = **root-cause diagnosis** of what a SAT approach would call UNSAT

Wrap-Up

Limitations

- cannot enforce that prescribed fixed points are the **only** ones (needs alternating quantifiers)
remember the extra fixed point $(1, 0, 0)$ in the toy example
- model enumeration currently one-by-one (blocking clauses)

Future work

- richer objectives: minimise regulations, minimise domain sizes, ...
- long-term behaviour beyond fixed points (trap spaces, attractors)
- batched enumeration via partially-specified functions

1. **Weighted MaxSMT makes exact inference robust**: noisy data no longer means UNSAT — you always get a model, a suboptimality bound, and the violated observations as diagnosis
2. **Value-based vs. comparison-based**: ordering constraints remove the discretisation threshold; with cautious weights, *“the best model with no assumptions is the empty model”* — every inequality is an assumption you explicitly own
3. **It scales**: uninterpreted functions + monotonicity instantiation handle $\geq 1,300$ genes / $\geq 90k$ soft constraints; multi-valued domains supported (slower, but tractable)

Tool: github.com/sybila/biodivine-algo-smt-inference

Artefact: doi.org/10.5281/zenodo.19508672

Backup: the weighted inference problem, formally

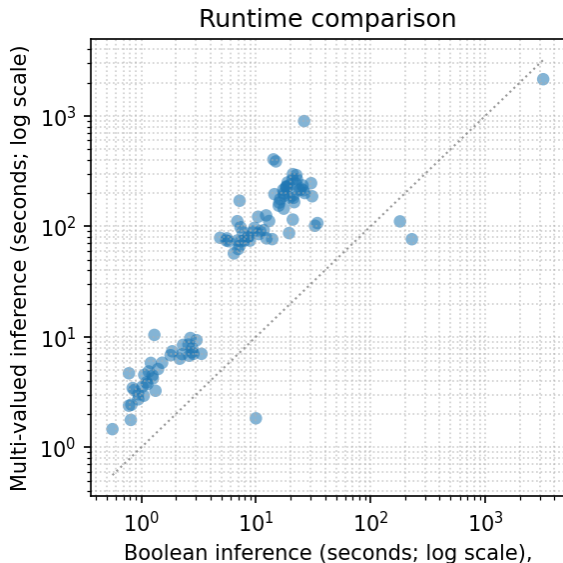
Input:

- domains $D_1, \dots, D_n$ (Boolean or finite integer intervals)
- influence graph $\mathcal{I} = (V, E)$ with regulation constraints (Ess, Mon)
- state variables s_i^j (m fixed points), distinctness flag
- observation constraints $\mathcal{C} = \mathcal{C}_{hard} \cup \mathcal{C}_1 \cup \dots \cup \mathcal{C}_l$ with weights

Feasible solution $(\mathcal{F}, \mathcal{S})$: MVN consistent with $\mathcal{I}$ and (Ess, Mon) ; each s^j a fixed point; hard constraints hold. **Optimal solution**: lexicographic maximum of the level-weight tuple

$$(w_1(\mathcal{S}), \dots, w_l(\mathcal{S})), \quad w_k(\mathcal{S}) = \sum \{w(c) \mid c \in \mathcal{C}_k, \mathcal{S} \models c\}.$$

Backup: Boolean vs. multi-valued



- same encoding, larger domains
- order-of-magnitude slowdown, still tractable
- here: worst case (domain size = out-degree); practical domains much smaller

Questions?