Data-driven design of logic-based models of biological processes

Samuel Pastva
Masaryk University, Brno

There is this theory...

At some point in the near future, we won't be able to perform traditional "explainable" science anymore; the only things capable of making *increasingly better predictions* of reality will be black box models like LLMs.

This is rather bleak; I don't believe it is our future, but there is perhaps a grain of truth in it...

Complex systems

We are gathering an unprecedented amount of data about complex natural systems; to some extent, we simple don't known what to do with it.

We are very good at explaining *first-order* phenomena.

> Mutation of HTT gene directly causes Huntington's disease.

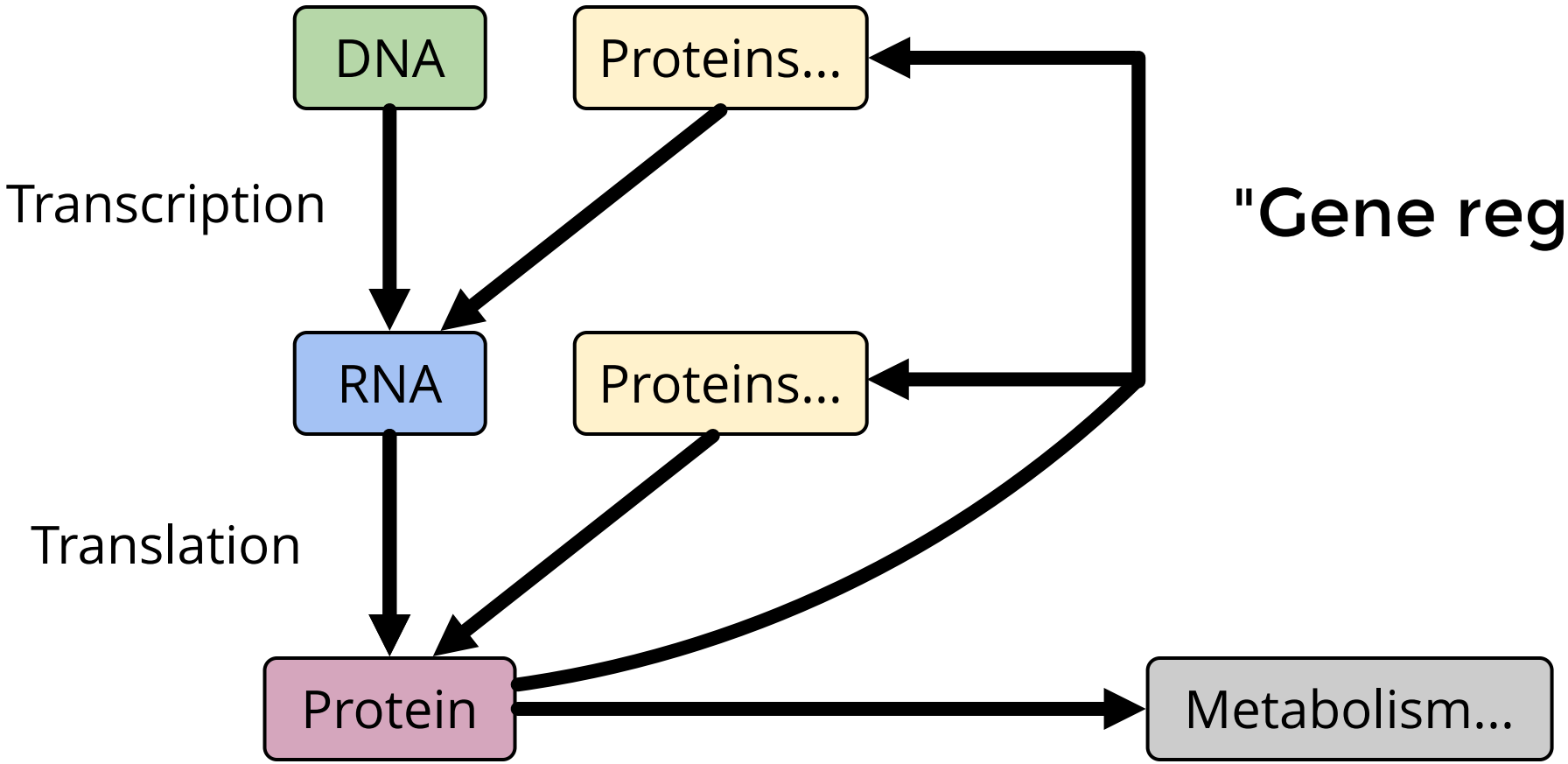
We are very bad at explaining *emergent* phenomena.

> There is no "first-order" explanation of human height. It is an emergent property based on hundreds of factors.

There is this theory...

Maybe what we are running out of are not *explainable scientific hypotheses*, just explainable *first-order phenomena*. We can still get better at learning and understanding the emergent ones.

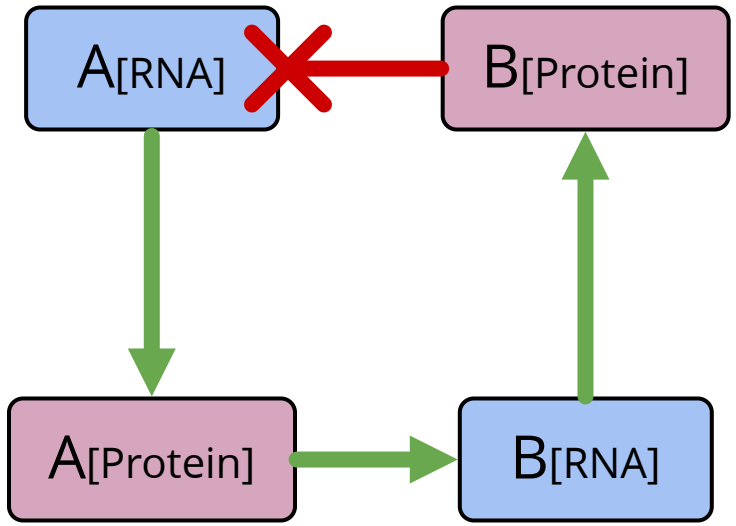
Complex systems in biology



"Gene regulation"

Boolean network model of gene regulation

- Network variables; interacting entities, each being either active (1) or inactive (0).
- Each variable has a single Boolean update function which defines its next state.
- Variables update non-deterministically, one at a time (approximates stochastic chemistry).
- Emergent properties: What happens to the network state "in the long term"?
- Attractors ⇒ Minimal regions in state space that cannot be escaped by network dynamics.



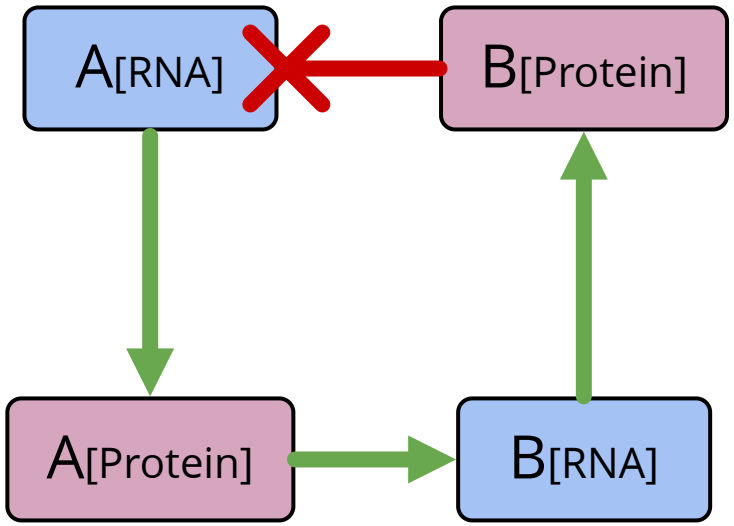
Boolean network model of gene regulation

Update functions

$$A_{RNA}^{(t+1)} = \neg B_{Protein}^{(t)}$$
$$A_{Protein}^{(t+1)} = A_{RNA}^{(t)}$$
$$B_{RNA}^{(t+1)} = A_{Protein}^{(t)}$$
$$B_{Protein}^{(t+1)} = B_{RNA}^{(t)}$$

Hypothetical trajectory

$$1000 \rightarrow 1100$$
$$1100 \rightarrow 1110$$
$$1110 \rightarrow 1111$$
$$1111 \rightarrow 0111$$
$$0111 \rightarrow 0011$$
$$0011 \rightarrow 0001$$
$$0001 \rightarrow 0000$$
$$0000 \rightarrow 1000$$



Linear relationships between $X_{[RNA]}$ and $X_{[Protein]}$ are often simplified to single variable X.

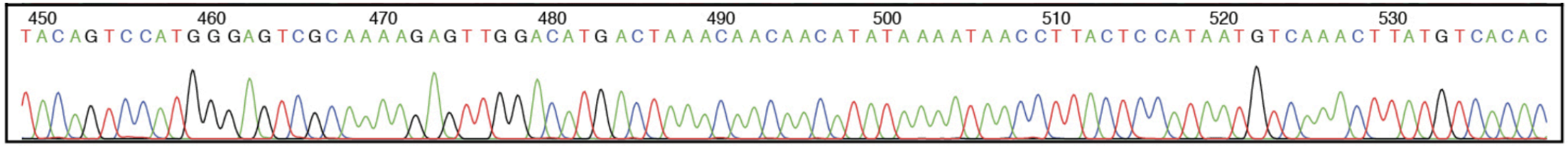
Boolean network model of gene regulation

- Historically, researchers built Boolean models by hand:
  - as a demonstration of a hypothesis,
  - to compactly share domain knowledge, ...
- **What modern tools and measurements can we use today to build Boolean models driven by data?**

Note: *gene* is a particular *sequence of DNA*, but colloquially it can sometimes mean the RNA/protein molecule derived from that sequence (e.g. gene A regulates B).

DNA sequencing

- Static information about everything that can be (hypothetically!) created by a cell.
- Boolean model: DNA defines the possible variables of the model, i.e. the genes (and proteins) that can interact with each other.
  - Note: Sometimes, the environment is also important!
- Genes have a relatively well defined "start" and "end" sequence that derives from the molecules that perform transcription (~10.000-20.000 genes in mammals).



Single-cell RNA sequencing (RNA-seq)

- Table of X cells and Y genes, each value giving the number of RNA molecules of that gene found in the cell (can be 1.000+ cells).
- Captures all RNA that was recently transcribed; *usually*, this also means the corresponding proteins are being produced.
- A "snapshot" of cell dynamics. In a Boolean model, this is (after binarization) the state of model variables in that cell.
- We often group cells with similar expression patterns (and thus functionality) into *phenotypes*.
- Note: Some smaller RNA molecules are not translated to proteins, but fulfill other functions.

	Gene A	**Gene B**	**Gene C**
Cell 1	12	1	5
Cell 2	4	258	0

	Gene A	**Gene B**	**Gene C**
Cell 1	true	false	true
Cell 2	false	true	false

Single-cell "uncertainty principle"

- In computer science, we often assume that observed data comes as a *time series*; i.e. we have an exact time point for each measurement, and the measurements follow each other.
- *This is very much not true for single-cell data.* Each cell can be only measured once (it is destroyed by the measurement).
- We can sample cells from a population evolving over time, but we can never have true time series of subsequent states.
- *Pseudo-time* of stable vs. transient phenotypes:

Day 1	**Day 2**	**Day 3**	**Day 4**
25x **P1**	22x **P1** 5x **P2**	3x **P1** 18x **P2** 3x **P3**	6x **P2** 12x **P3**

P1 >>> **P1 & P2** >>> **P1 & P2 & P3** >>> **P2 & P3**

Computing RNA "velocity"

- As RNA exits nucleus, it is "cleaned up" (spliced). With sufficiently precise sequencing methods, we can distinguish RNAs before and after splicing ("old" vs. "new" RNA).
- We can identify whether RNA is being actively replenished.
- In the Boolean model, this is equivalent to having both the gene state and the output of gene's update function in that state.
- Compared to normal RNA counts, RNA velocity is still somewhat less precise due to the high sequencing quality it requires.

	Gene A	**Gene B**	**Gene C**
Cell 1	true (+)	false (+)	true (-)
Cell 2	false (-)	true (+)	false (+)

RNA tells us what is being produced, but does not tell us why? (Why aren't cells doing everything everywhere all at once?)

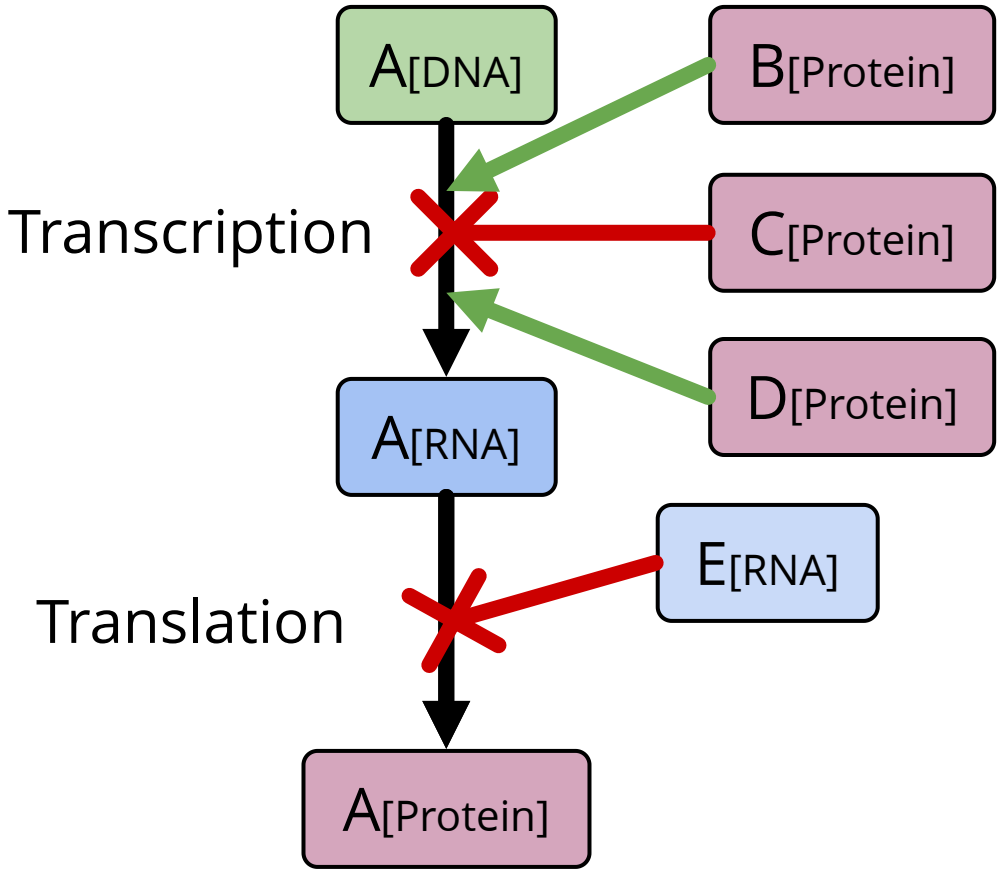
- Transcriptional regulation (affects DNA ⇒ RNA processes)
- Translational regulation (affects RNA ⇒ Protein processes)

Translational regulation

- Molecules that "attack" and destroy RNA once it exits the nucleus.
- These molecules are often "small RNAs" that don't produce proteins of their own.
- Can adapt much faster than transcriptional regulation (e.g. immunity response), but is very resource intensive; hence it rarely plays a lasting role in long-term / irreversible processes.
- How can we tell if translational regulation is taking place?
  - Ribosome sequencing (Ribo-seq) detects RNA that is actively being turned into protein.
  - Targeted techniques detecting molecules binding to RNA (generally much harder than RNA-seq or even Ribo-seq).

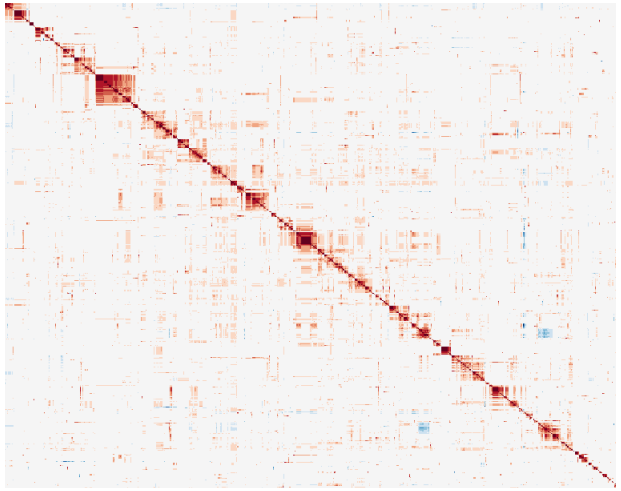
Transcriptional regulation

- Transcription factors (TFs): *activators* and *repressors*.
- Bind to regions (enhancers and silencers) adjacent to the target gene (but adjacent can still mean quite some distance!).
- Actively attract or repel proteins responsible for transcription.
- Relatively small sub-group of ~1.000 - 2.000 proteins.
- Transcription factors don't act independently: For most genes, there is some non-trivial but unknown activation logic which requires / forbids combinations of TFs.
- Hence the motivation for logic-based modeling!
- How to detect transcription factors?
  - Perturbation experiments: Delete a specific binding region and see if expression of RNA changed (costly!).
  - ChIP-seq: Detects if a particular protein is actually bound to a DNA region at the time of measurement.


A[DNA]
B[Protein]
Transcription
C[Protein]
D[Protein]
A[RNA]
E[RNA]
Translation
A[Protein]

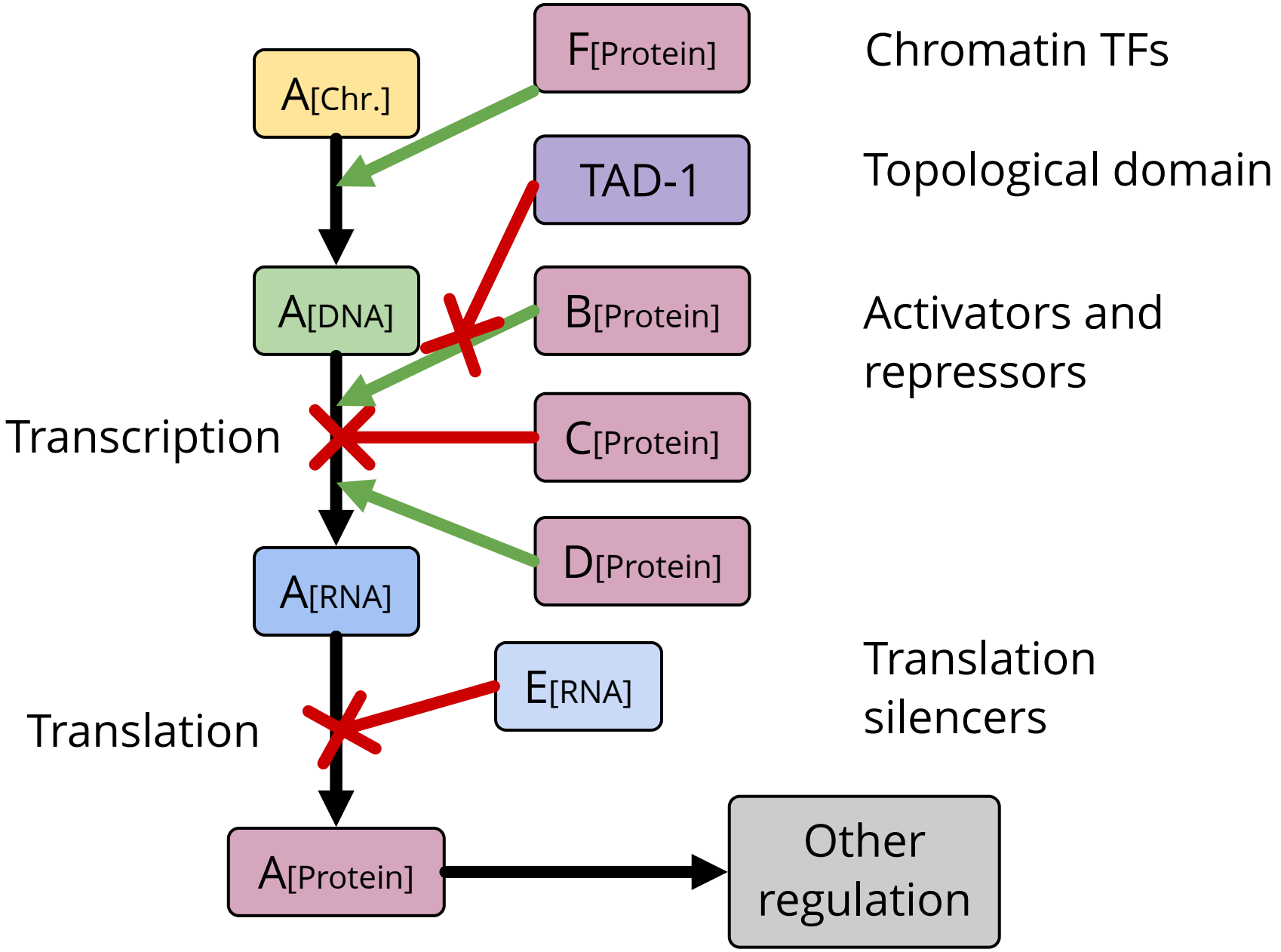
Topological domains

- Transcription factors can act over a relatively long distance, but they have to physically interact with the DNA strand.
- Cells form "topologically associating domains" (TADs) over DNA by isolating certain DNA compartments from each other.
- TFs cannot act outside of the compartment that they are in.
- Formation of TADs is relatively slow (e.g. during the span of organism development).
- How do we measure them? Hi-C mapping counts how often are sequences of DNA seen in close contact.
- In the Boolean model, this specifies which activators / repressors are actually admissible.



Chromatin accessibility

- DNA is tightly "compressed" (even within a single TAD); Before transcription, it must unwind and become "open".
- There are several mechanisms that cooperate in this process, but in general, there are certain special pioneer TFs that can open small sections of DNA.
- Additional TFs that bind in that open region then trigger unwinding of other close areas.
- Using special sequencing methods, we can detect parts of DNA that are currently open (ATAC-seq).
- In the Boolean model, this gives additional necessary conditions for RNA transcription (If chromatin is closed, there can be no RNA production).


F[Protein]
A[Chr.]
TAD-1
A[DNA]
B[Protein]
Transcription
C[Protein]
D[Protein]
A[RNA]
E[RNA]
Translation
A[Protein]
Other regulation
Chromatin TFs
Topological domain
Activators and repressors
Translation silencers

Bottom line: We can learn a lot about gene regulation from data, but we can't directly measure exact regulatory logic of every gene.

Partially specified Boolean networks (PSBNs)

- How do we formalize what we know about gene regulation?
- Instead of having exact update functions, we consider a BN that can use uninterpreted functions.
- Such partially specified BN then compactly represents an biologically admissible ensemble of BN candidates.
- We can combine uninterpreted functions with known regulatory logic, or break them up into smaller, independent regulatory mechanisms:

$$A^{(t+1)} = B \wedge (\neg T \Rightarrow f(C, D)) \wedge \neg g(R_1, R_2)$$

- B is known *required* transcription factor.
- C, D are transcription factors interacting via *unknown logic*, but only if topological domain T has *not formed*.
- An independent translational silencing depending on R1 and R2 is hypothesized.

Restricting PSBN dynamics ("BN Sketch")

- PSBNs describe the assumed relationships, but not all observations can be captured by the update rules directly.
- First-order properties ("static"):
  - Propositional formulas with first-order quantification and uninterpreted functions.
  - Essential inputs of functions ("Function *f* must use input A").
  - Local monotonicity of functions ("Input A is positively monotonic in function *f*").
  - Input-output relationships ("In state X, update function of A evaluates to true").
  - Stability properties ("In state X, update function of A evaluates to X[A]").

Restricting PSBN dynamics ("BN Sketch")

- Temporal properties ("dynamic"):
  - Hybrid CTL formulas with quantification over states of the Boolean model.
  - Reachability ("There is a path from state X to state Y").
  - Concrete attractor states ("State X can always eventually reach itself").
  - Abstract attractor states ("There exist two states in distinct attractors which differ in variable A").
  - Attractor commitment ("There is a state from which attractors A and B are reachable, but not attractor C").

Solving BN sketches

- SAT problem: Is there a concrete BN that instantiates all uninterpreted functions and satisfies static / dynamical properties?
  - Can be addressed by SAT/SMT/ASP solvers and/or model checkers (for dynamical properties).
- All-SAT problem: What is the full set of BN candidates?
  - Can be addressed by encoding the set of candidates as BDDs and running symbolic model checking to filter out invalid candidates, leaving a symbolic solution set.

Challenges ahead

- Partial observability: Due to measurement errors, we often don't know the full cell state. However, we can often quantify measurement error.
  - Current SAT/BDD tools can typically incorporate partial states into property specifications, but can't perform quantitative optimization over measurement errors.

	Gene A	**Gene B**	**Gene C**
Cell 1	?	?	true (60%)
Cell 2	false (90%)	true (90%)	false (80%)

Challenges ahead

- Identifiability: A PSBN often contains elements not required to reproduce the expected behavior, resulting in different candidates that are indistinguishable in their long-term behavior.
  - Currently, we can produce individual valid candidate BNs as well as (for smaller problems) full candidate sets.
  - We would like to quantify how big is the variance of possible behaviors of candidates in our solution space and how significant our candidate model is in this solution space.
- Refinement: When the result is poorly identified, we would like to select experiments whose results will optimally reduce our solution space.

Notable toolboxes

- AEON and Biodivine Sketchbook
  - Our group; Mostly relies on BDDs and symbolic model checking to address the all-solutions problem.
- Bonesis
  - Pauleve et al.; Uses ASP encoding to identify concrete BN candidates for PSBNs with wide range of static properties.
  - Can perform optimization to eliminate certain redundant elements of the PSBN.
- BREIN and Reasoning engine
  - Yordanov et al.; Uses combination of SMT and model checking to solve many static and dynamic properties, but does not support full uninterpreted functions.

Thank you for your attention!

If you are interested, come chat with me;
I'm here all week :)

Or get in touch:

- sam.pastva@gmail.com
- https://sybila.fi.muni.cz