Synthesis and Simulation of Ensembles of Boolean Networks for Cell Fate Decision

Stéphanie Chevalier[1], Vincent Noël[2]

[1]LRI, CNRS, UMR8623, Univ. Paris-Saclay, France
[2]Computational Systems Biology of Cancer,
Institut Curie, INSERM U900, Université PSL, Paris

20th International Conference on Systems Biology

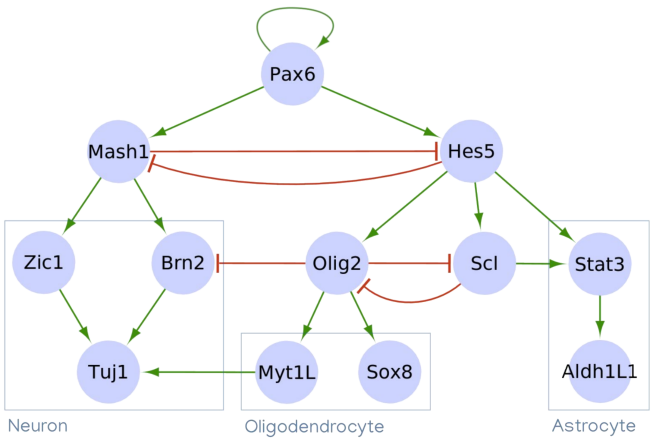
September 23, 2020

Introduction

- Context of the synthesis:

Computational models of molecular networks usually built from:

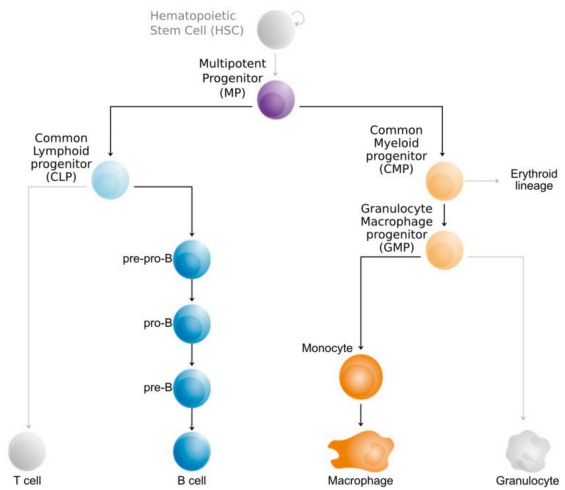
- the structure of the biological system (such as known interactions)



Prior Knowledge Network

- its dynamics

(such as measurements of expressions / activity at different time / conditions)



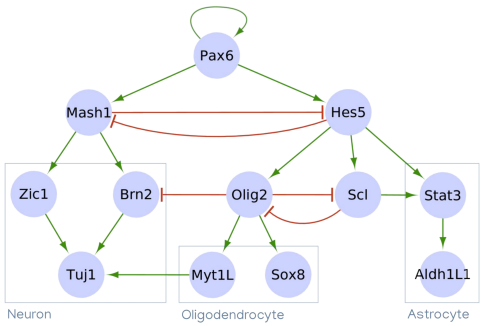
experimental data

Introduction

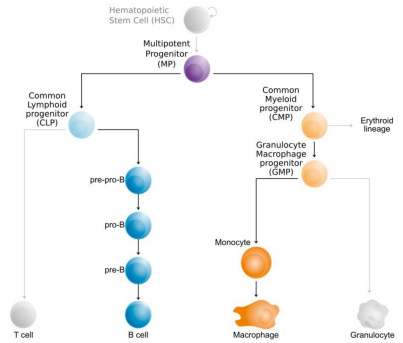
› Issue:

The model engineering problem largely under-specified:

⇒ **many potential candidate models**



Prior Knowledge Network



experimental data

model n°1
$f_1(x) = \neg x_2$
$f_2(x) = \neg x_1$
...

model n°2
$f_1(x) = \neg x_2 \wedge x_1$
$f_2(x) = \neg x_1$
...

model n°3
$f_1(x) = \neg x_2 \wedge x_1$
$f_2(x) = \neg x_1 \vee x_3$
...

model n°4
$f_1(x) = x_1$
$f_2(x) = x_3$
...

model n°...
...

Biases in subsequent predictions if an arbitrary single model is retain.

Introduction

- Our methodology:

Ensemble-based approach to Boolean modelling

Synthesizing and reasoning on dynamics of ensembles of Boolean networks:

- **synthesizing Boolean model ensembles**
  satisfying a set of biologically relevant constraints
- **reasoning on the dynamics of the ensembles of models**

The main lines of the work we present are:

- a synthesis method
- a simulation method based on ensemble of models
- an illustration on a model of molecular pathways regulating tumour invasion and migration *(Cohen et al. [2015] PLoS Comp. Bio.)*

Definitions

A Boolean network *(BN)* of dimension n is a function $f : \{0,1\}^n \to \{0,1\}^n$

$\forall i \in [n]$, $f_i : \{0,1\}^n \to \{0,1\}$ denotes the *local function* of the i^{th} components

Example: $f_1(x) = \neg x_2$; $f_2(x) = \neg x_1 \wedge x_2$

A configuration is a vector $x \in \{0,1\}^n$

BN semantics specify, from a configuration, how to compute the next possible ones. For synthesis, we consider the Most Permissive Semantics because:

- guarantees to not preclude any behaviour realisable in any quantitative refinement of the model
- any behaviour it predicts is realisable by a quantitative refinement of the BN using the asynchronous semantics
- the complexity for deciding main dynamical properties is considerably lower than with (a)synchronous semantics

"Reconciling qualitative, abstract, and scalable modeling of biological networks", Paulevé et al. [2020]

Synthesis

› Prior work

Synthesis problem formulated as a Boolean satisfiability problem, implemented in Answer-Set Programming.

Based on our prior work on BNs synthesis from reachability and attractor properties with Most Permissive semantics

"Synthesis of Boolean Networks from Biological Dynamical Constraints using Answer-Set Programming." Chevalier S., Froidevaux C., Paulevé L., Zinovyev A.

It leverages:

- **a priory knowkedge** as constraints on the graph topology
- **experimental data** as constraints on the dynamical properties

A single logic program contains the whole set of constraints, providing non-redundant solutions.

Synthesis

- Extension

CMSB KONSTANZ2020

Boolean network synthesis extended by:

- **enabling universal properties on (reachable) fixed points**
- **considering different network perturbation settings**
- **using heuristics to drive the ASP solver in different regions of the solution space**

Synthesis

- Universal constraint

Method extended with **universal property**

Universal property: $\exists \rightarrow \forall$

Such a property not only ensures that a described behaviour is in the system dynamics, **it ensures it's the only possible behaviour**.

Example given a list of experimentally observed cell fates:

- existential: at least one attractor matches with each cell fate
- universal: every model attractor matched with at least one of the cell fate

Addressed in ASP thanks to the **saturation technique**

(presented by *Eiter and Gottlob [1995] in Annals of Mathematics and Artificial Intelligence*)

Synthesis

- Universal constraint

Encoded universal properties:

- Universal property on **fixpoints**
- Universal property on **fixpoints reachable from a given configuration**

Ensure that **all the fixed points of the BN (or those reachable from a configuration of interest) are compatible with a given set of markers.**

Combined with mutations and previous implemented constraints, **the method leverages observations about cell fates in different mutation conditions.**

Synthesis

- Diversity

Mechanism of enumeration by the solver:

- a 1st solution is identified
- followings come from successive slight variations

Consequence:

- partial enumeration $\rightarrow$ set of similar solutions
  not representative of the diversity of the comprehensive set of models

Our strategy to sample ensembles of diverse BNs:

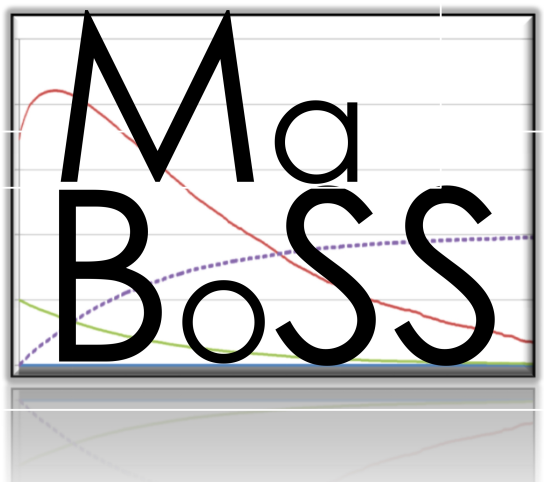
Tweak heuristics of the solver clingo to stir it towards distant solutions

- at each solution, we randomly select a subset of variables assignments
- we ask the solver to avoid them in the next iterations

Simulation

- MaBoSS

Markovian **Bo**olean **S**tochastic **S**imulator



https://maboss.curie.fr/

- Boolean
- State probability trajectories
- Physical time
- Handle different time scale processes (transcription, phosphorylation, etc.)
- Efficient (C++, parallel)

Simulation

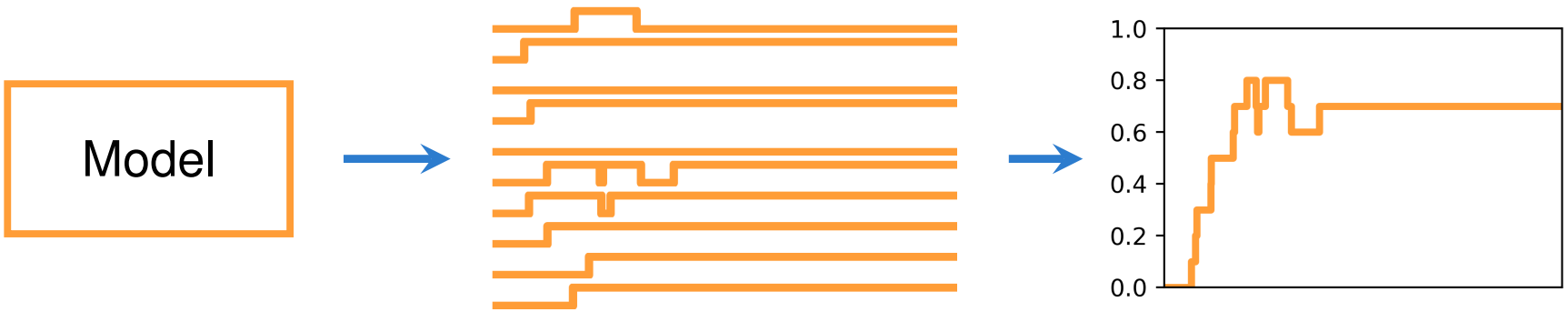
- MaBoSS

Continuous time Markov process applied on a Boolean network state space

Transition rate :

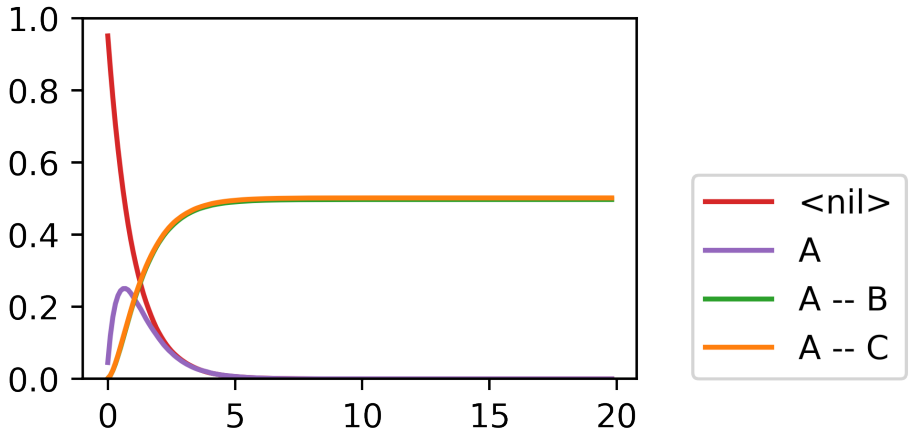
$$p(S \to S') = \begin{cases} R_{up}(S), & \text{if} S_i = 0 \\ R_{down}(S), & \text{if} S_i = 1 \end{cases}$$

$\Rightarrow$ In this study, we kept these rate parameters to 1

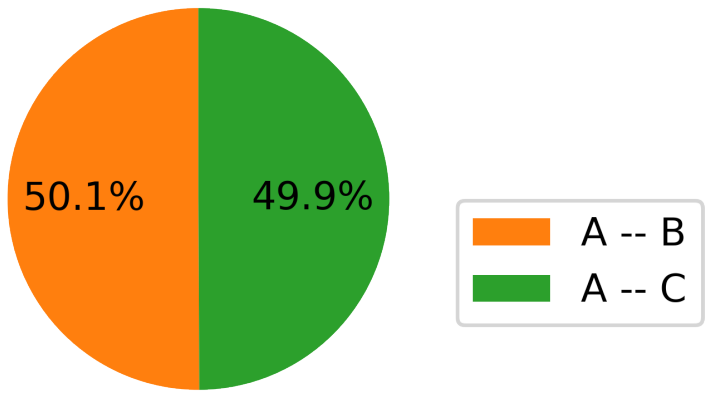


Simulation

- MaBoSS
  - State probability trajectories

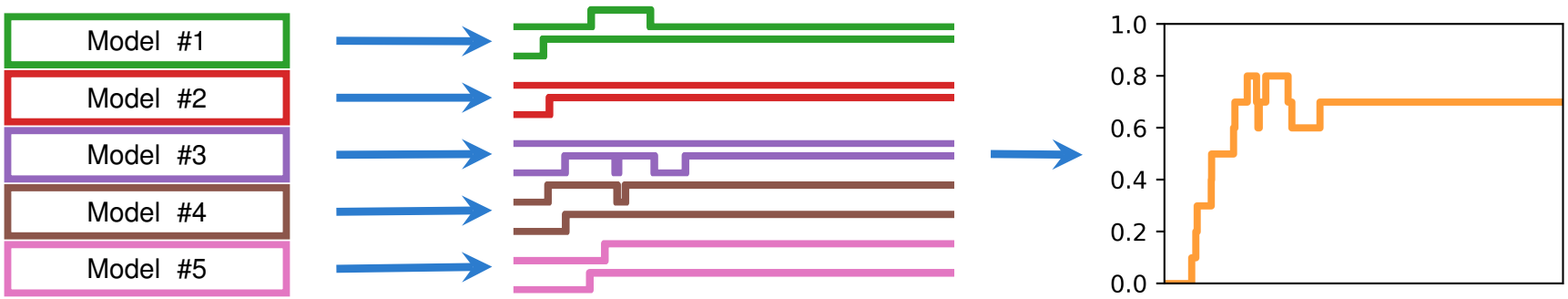


  - Steady state distribution



Simulation

- EnsembleMaBoSS

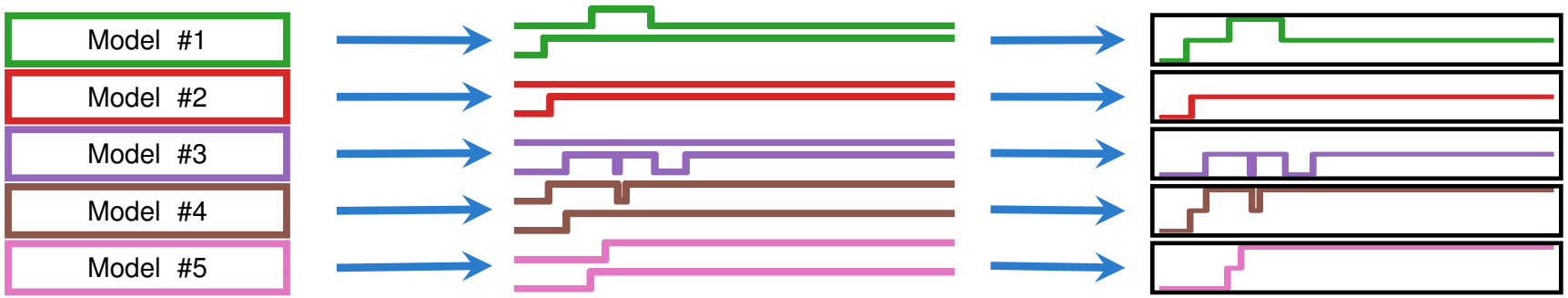


- MaBoSS needs few modifications to implement ensemble simulations

- We can select models randomly, or sample uniformely the model-space

Simulation

- Individual results

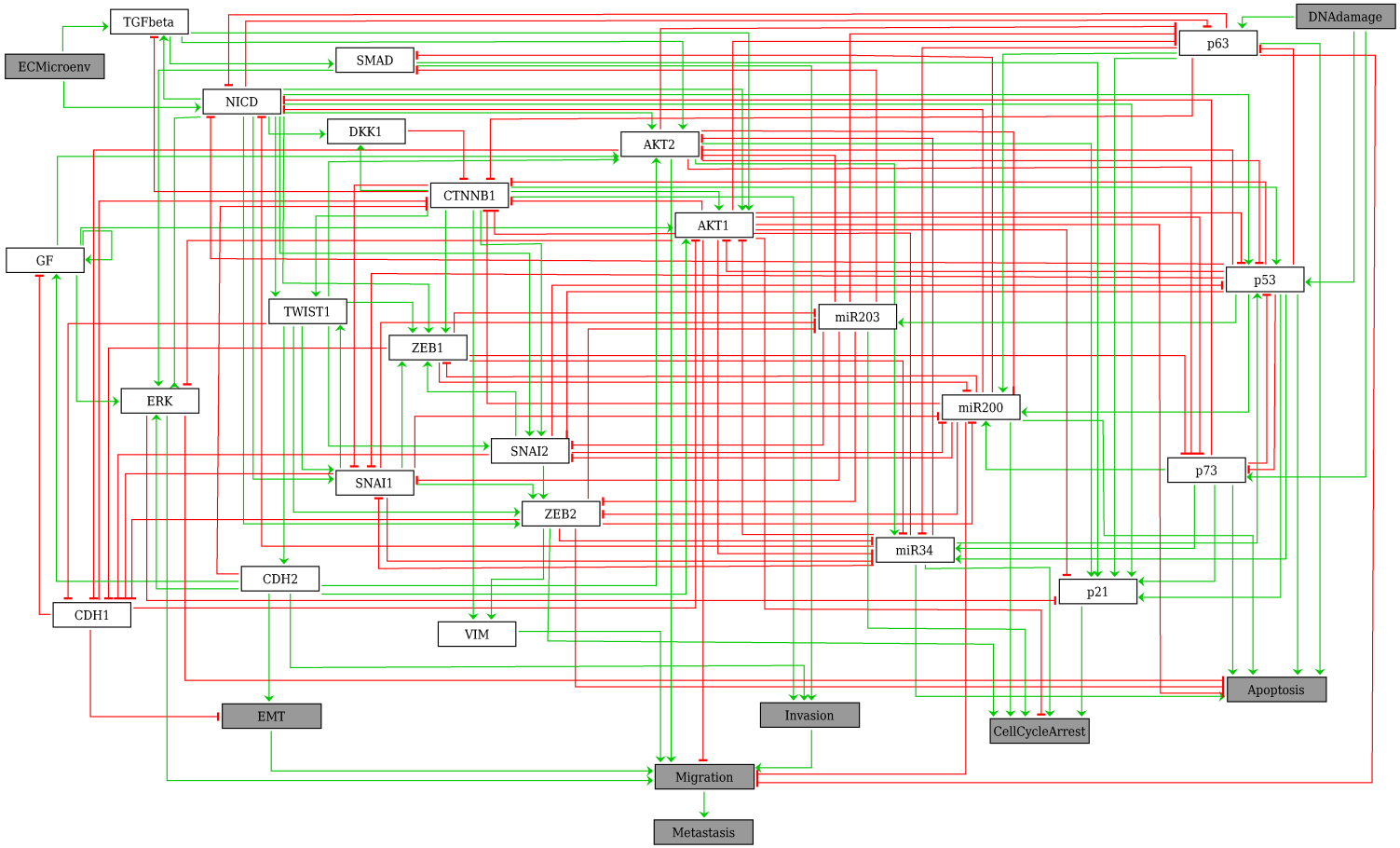


- MaBoSS needs few modifications to implement ensemble simulations
- We can select models randomly, or sample uniformly the model-space
- We can also save the network state probability distribution for each model

Application

- Cohen's model

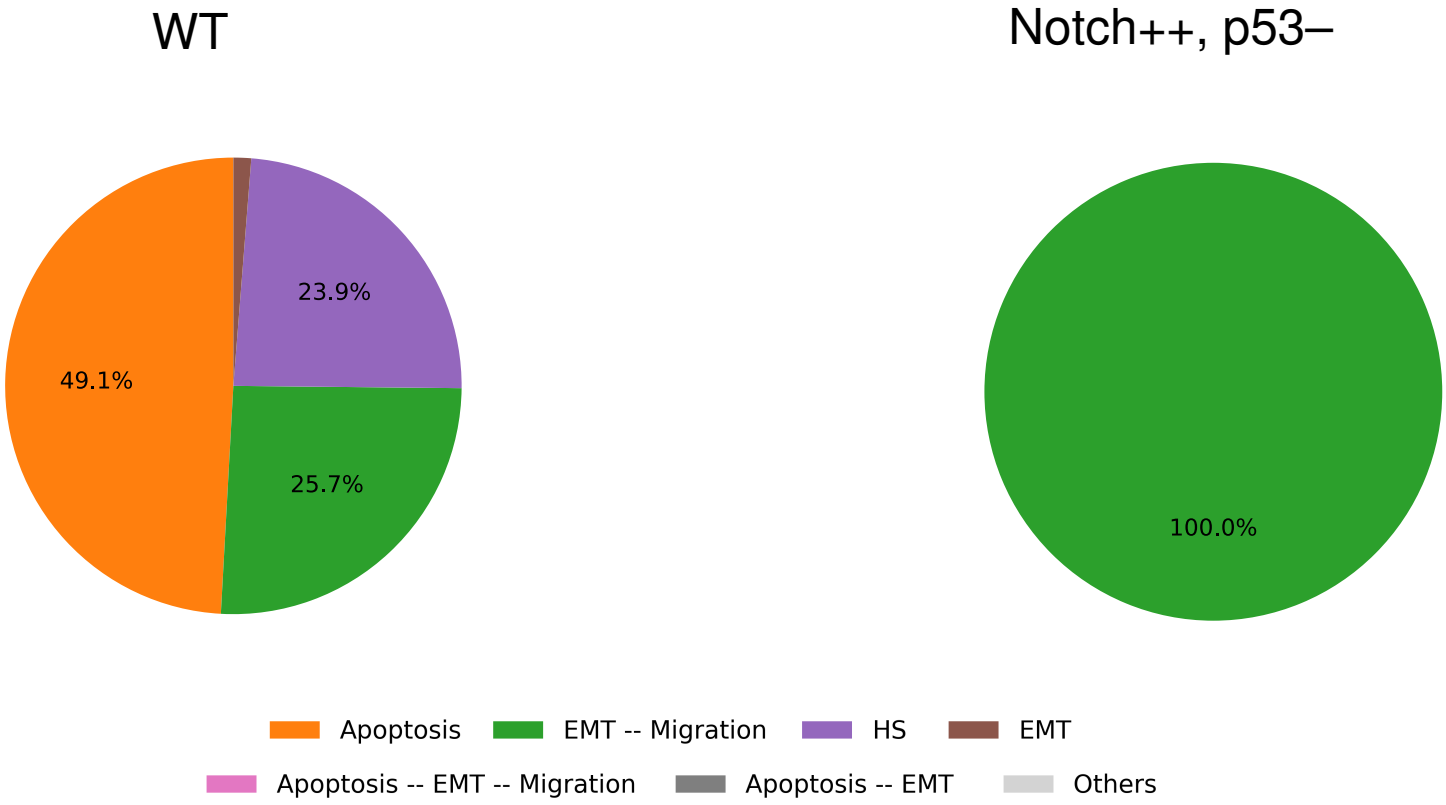
A model of a molecular pathways regulating tumour invasion and migration



Cohen et al. (2015) PLoS Computational Biology

Application

- Simulation results



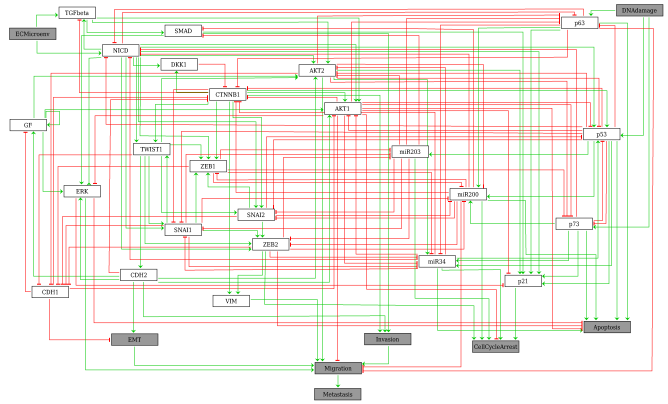
- Wild Type is mainly apoptotic, with ≈ 25% migration
- Notch++, p53– mutant turns full invasive (known experimental result)

Application

- Ensemble synthesis

Constraints:

- Cohen's interaction graph as Prior Knowledge Network
- Universally reachable fixpoints for a specific condition



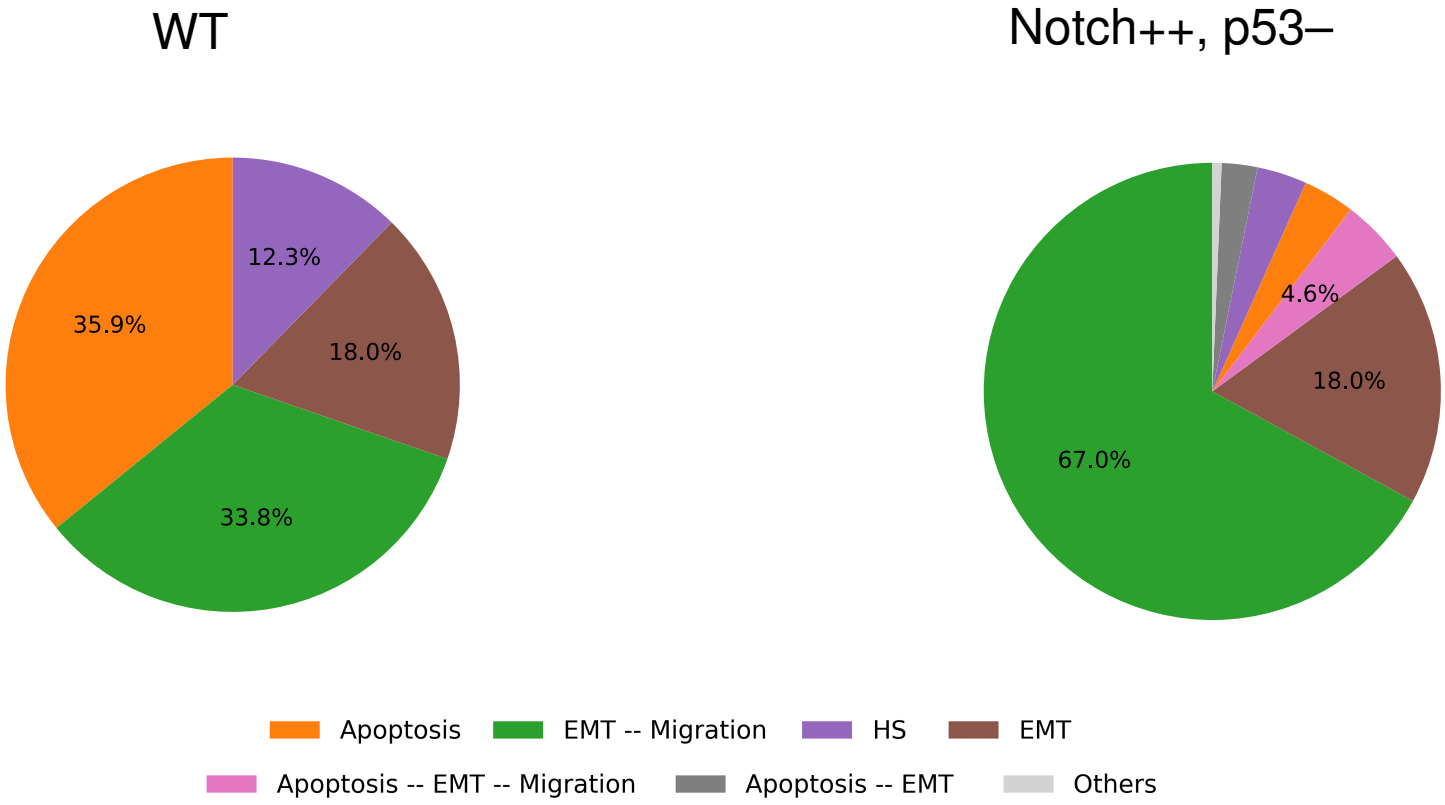
Cohen et al. (2015) PLoS Computational Biology

Two different ensembles :

- Reachability of Cohen's WT fixpoints
- Reachability of Cohen's WT + two mutants : Notch++ and p53– (two invidivual mutants)

Application

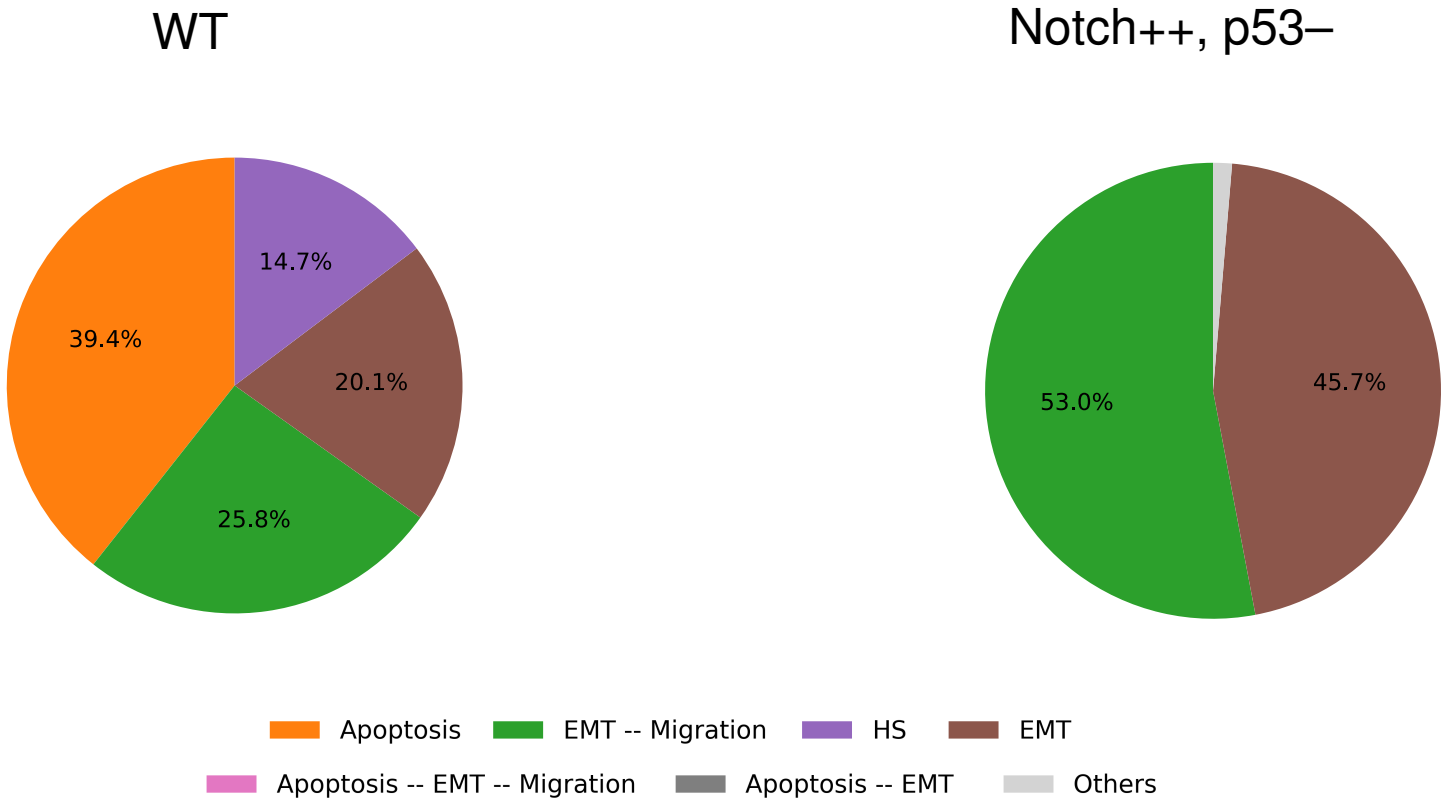
- Simulating our ensemble invasion model



- Our WT model is more invasive, with more activation of EMT
- Notch++, p53– mutant is more diverse, including some aberrant behaviour

Application

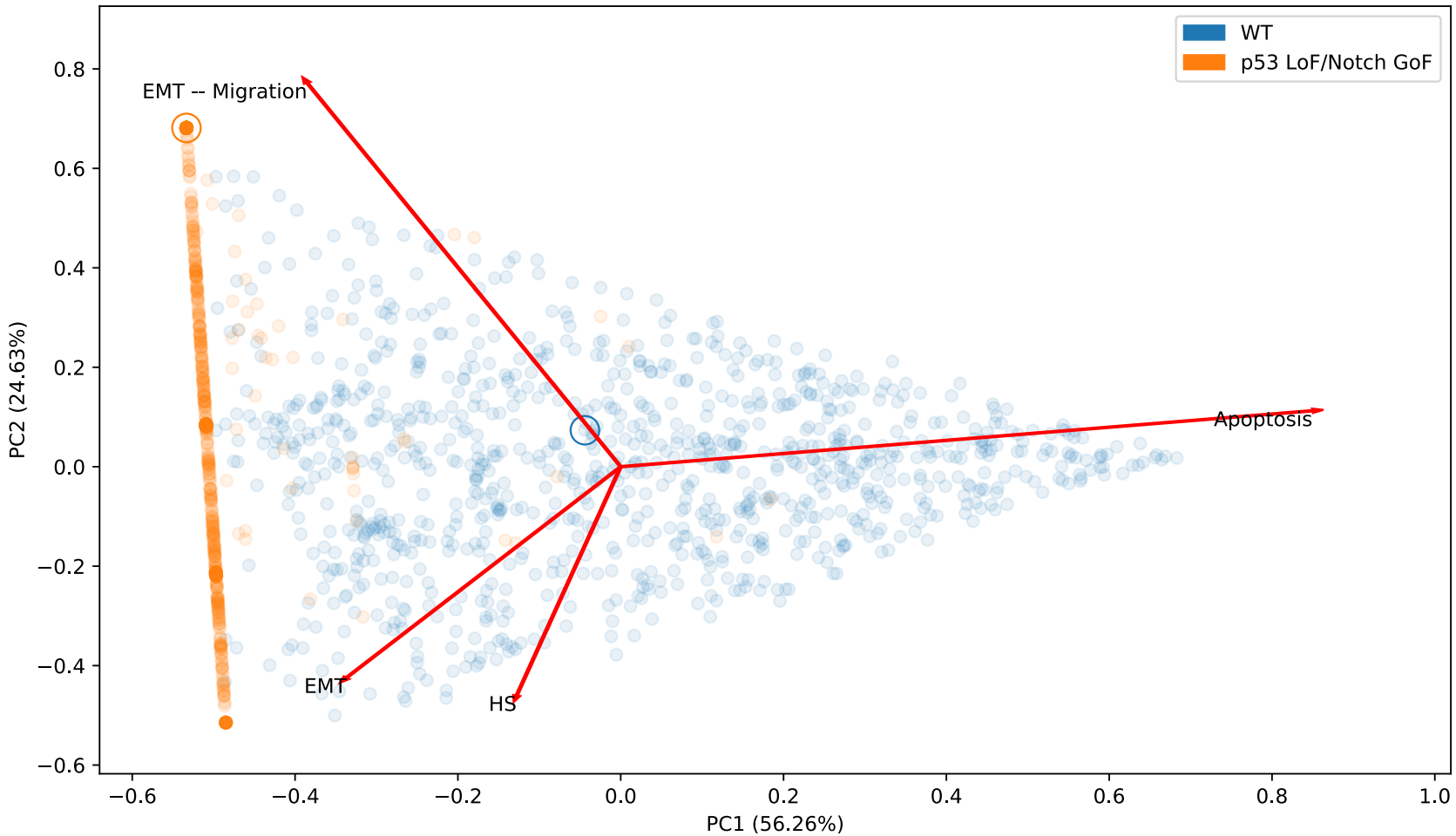
- Simulating our ensemble invasion model with additional constraints



- Our WT model build with global is more invasive
- Less aberrant behaviour

Application

- Simulating our ensemble invasion model with additional constraints



Conclusion

- Now we can generate ensembles of models applying universal constraints
- EnsembleMaBoSS can efficiently simulate these ensembles
- Diversity of models can be visualized using dimensionality reduction methods
- Diversity of models needs to be further evaluated

Conclusion

Our complete pipeline is available as Jupyter notebooks.

`https://doi.org/10.5281/zenodo.3938904`

BoNesis, our python library for synthesis of ensembles of boolean models

`https://github.com/bioasp/bonesis`

pyMaBoSS, our python library for simulation of (ensembles of) boolean models

`https://github.com/colomoto/pyMaBoSS`