PINT: a Static Analyzer for Transient Dynamics of Qualitative Networks with IPython Interface

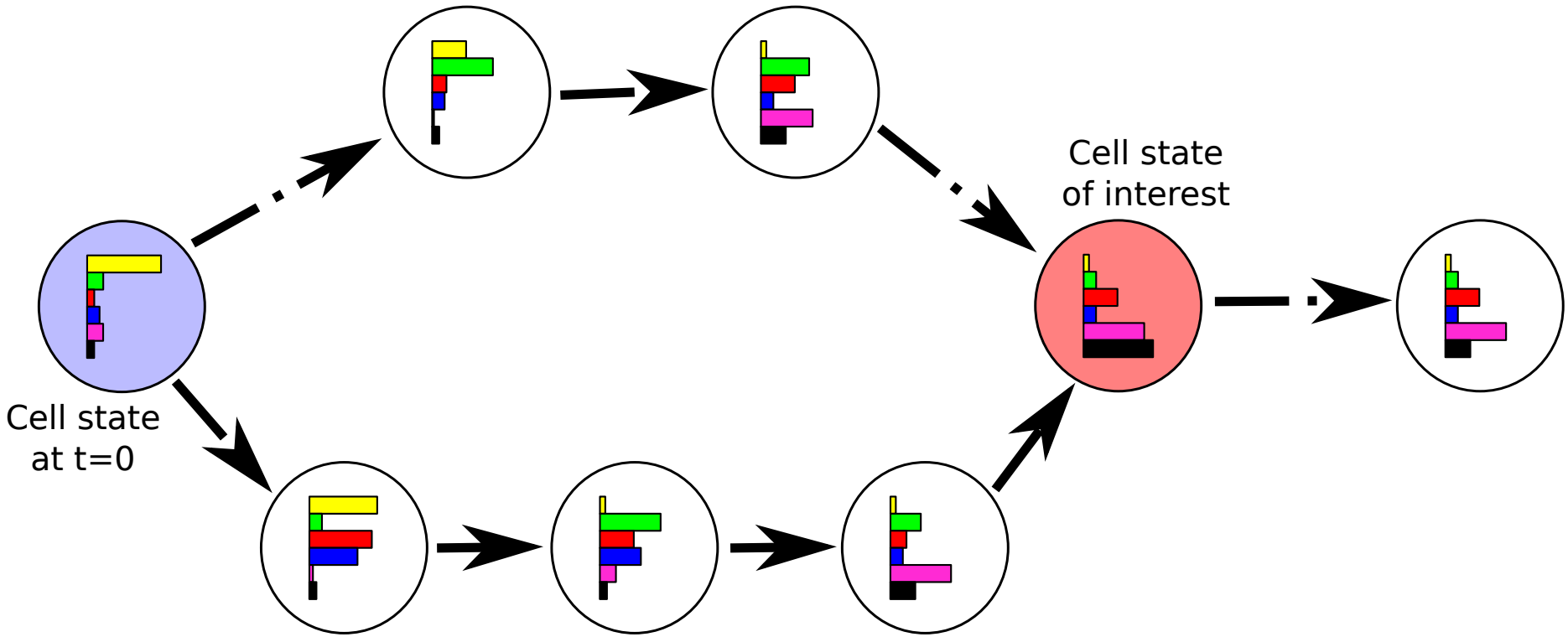
Loïc Paulevé

CNRS/LRI, Univ. Paris-Sud, Univ. Paris-Saclay – BioInfo team
loic.pauleve@lri.fr
http://loicpauleve.name

CMSB 2017 - 28th September 2017, Darmstadt, Germany

ANR-FNR “AlgoReCell” project (2017-2020)
CNRS PEPS INS2I “FoRCe” project (2017)

Cellular Dynamics

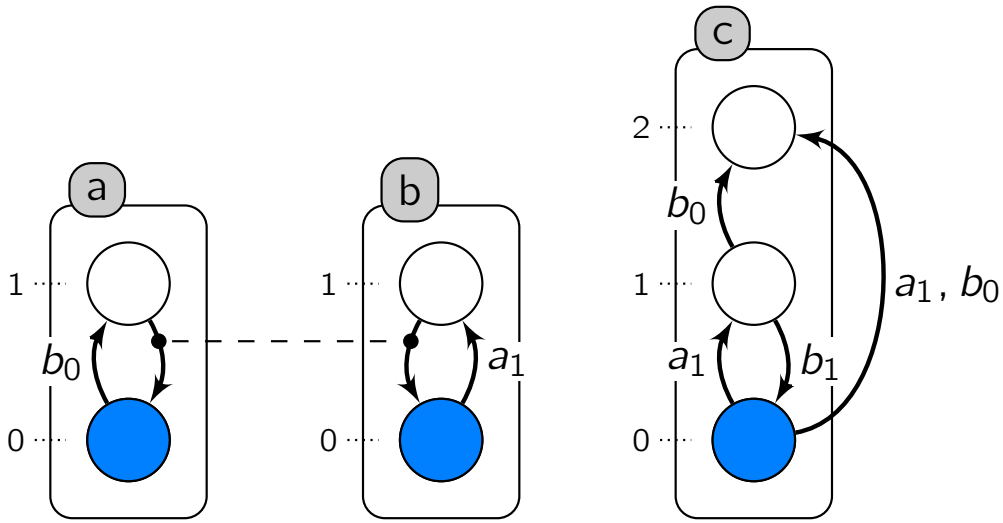


Initial state(s)/Goal state(s)

- Trajectory **existence** (reachability)
- Reasoning on all trajectories: e.g., **common features**
- **Control**: perturbations to avoid/enforce goal reachability

Input formalism

General framework: **Automata networks**




```
model.an:
a [0, 1]
b [0, 1]
c [0, 1, 2]

a 0 -> 1 when b=0
b 0 -> 1 when a=1
{ a 0 -> 0; b 1 -> 0}

c 0 -> 1 when a=1
c 0 -> 2 when a=1 and b=0
c 1 -> 2 when b=0
c 1 -> 0 when b=1
```

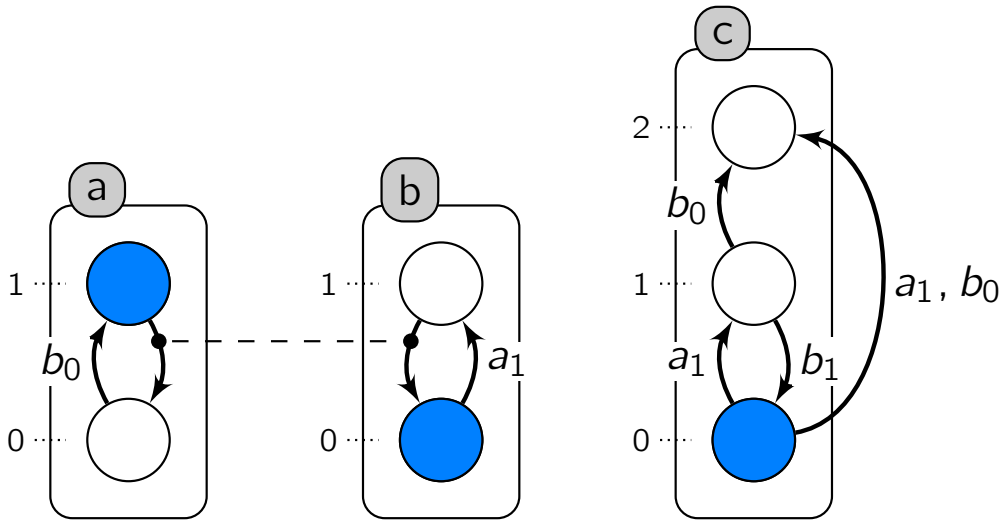
Model input

- Asynchronous Boolean/multi-valued networks
- Formats: any supported by GINsim/BioLQM (SBML-Qual, Boolsim, ..)
- Other: 1-b Petri nets (soon); Biocham (Boolean semantics); SBGN-PD
- Models can also be imported remotely:

```
pypint.load("http://ginsim.org/sites/default/files/drosophilaCellCycleVariants.zginml")
pypint.load("https://cellcollective.org/#4705/septation-initiation-network")
```

Input formalism

General framework: **Automata networks**




```
model.an:
a [0, 1]
b [0, 1]
c [0, 1, 2]

a 0 -> 1 when b=0
b 0 -> 1 when a=1
{ a 0 -> 0; b 1 -> 0}

c 0 -> 1 when a=1
c 0 -> 2 when a=1 and b=0
c 1 -> 2 when b=0
c 1 -> 0 when b=1
```

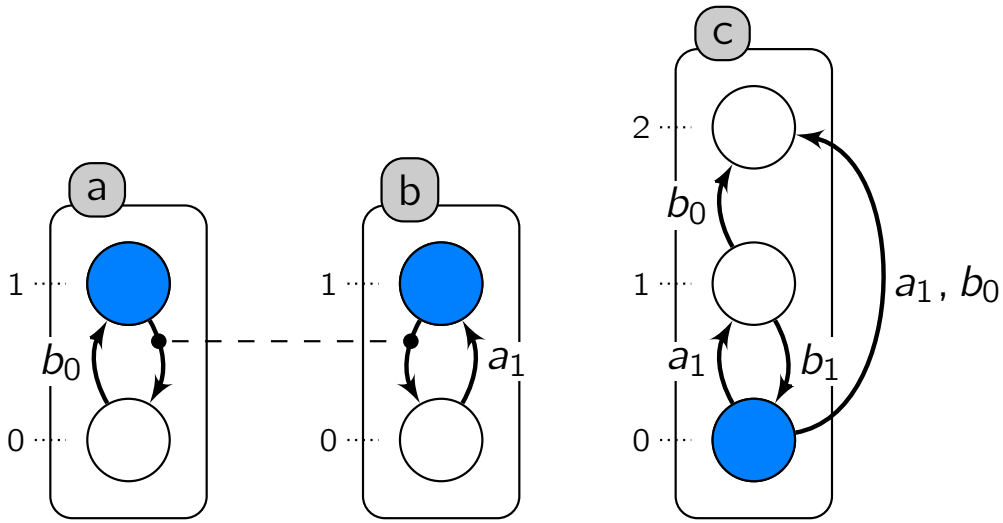
Model input

- Asynchronous Boolean/multi-valued networks
- Formats: any supported by GINsim/BioLQM (SBML-Qual, Boolsim, ..)
- Other: 1-b Petri nets (soon); Biocham (Boolean semantics); SBGN-PD
- Models can also be imported remotely:

```
pypint.load("http://ginsim.org/sites/default/files/drosophilaCellCycleVariants.zginml")
pypint.load("https://cellcollective.org/#4705/septation-initiation-network")
```

Input formalism

General framework: **Automata networks**




```
model.an:
a [0, 1]
b [0, 1]
c [0, 1, 2]

a 0 -> 1 when b=0
b 0 -> 1 when a=1
{ a 0 -> 0; b 1 -> 0}

c 0 -> 1 when a=1
c 0 -> 2 when a=1 and b=0
c 1 -> 2 when b=0
c 1 -> 0 when b=1
```

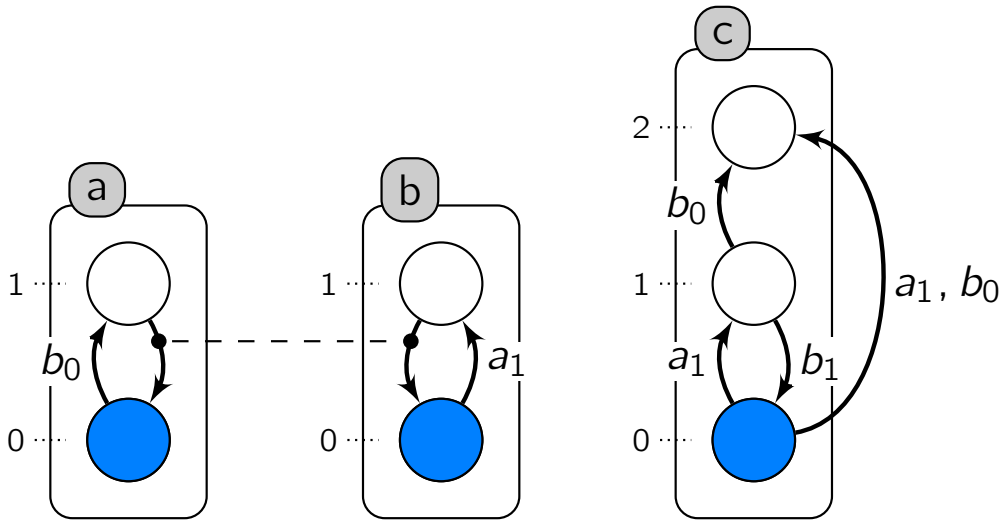
Model input

- Asynchronous Boolean/multi-valued networks
- Formats: any supported by GINsim/BioLQM (SBML-Qual, Boolsim, ..)
- Other: 1-b Petri nets (soon); Biocham (Boolean semantics); SBGN-PD
- Models can also be imported remotely:

```
pypint.load("http://ginsim.org/sites/default/files/drosophilaCellCycleVariants.zginml")
pypint.load("https://cellcollective.org/#4705/septation-initiation-network")
```

Input formalism

General framework: **Automata networks**



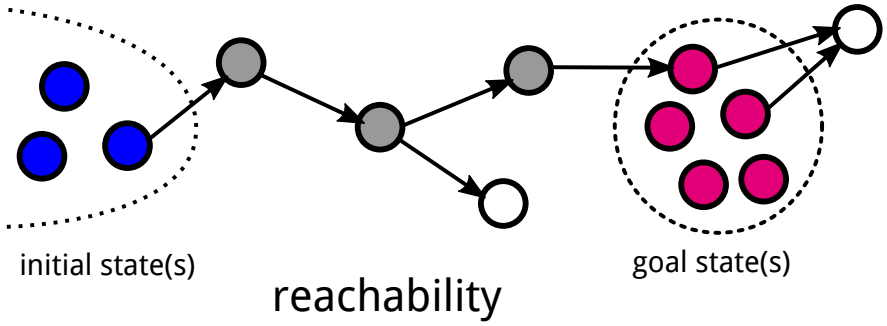

```
model.an:
a [0, 1]
b [0, 1]
c [0, 1, 2]

a 0 -> 1 when b=0
b 0 -> 1 when a=1
{ a 0 -> 0; b 1 -> 0}

c 0 -> 1 when a=1
c 0 -> 2 when a=1 and b=0
c 1 -> 2 when b=0
c 1 -> 0 when b=1
```

Model input

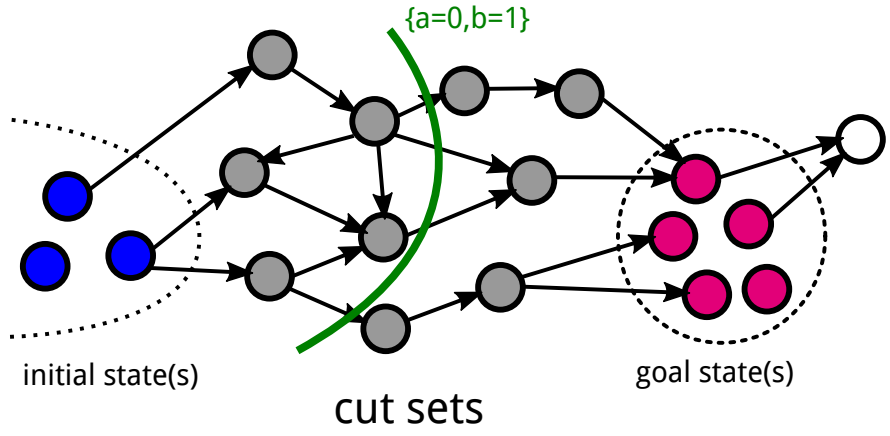
- Asynchronous Boolean/multi-valued networks
- Formats: any supported by GINsim/BioLQM (SBML-Qual, Boolsim, ..)
- Other: 1-b Petri nets (soon); Biocham (Boolean semantics); SBGN-PD
- Models can also be imported remotely:

```
pypint.load("http://ginsim.org/sites/default/files/drosophilaCellCycleVariants.zginml")
pypint.load("https://cellcollective.org/#4705/septation-initiation-network")
```

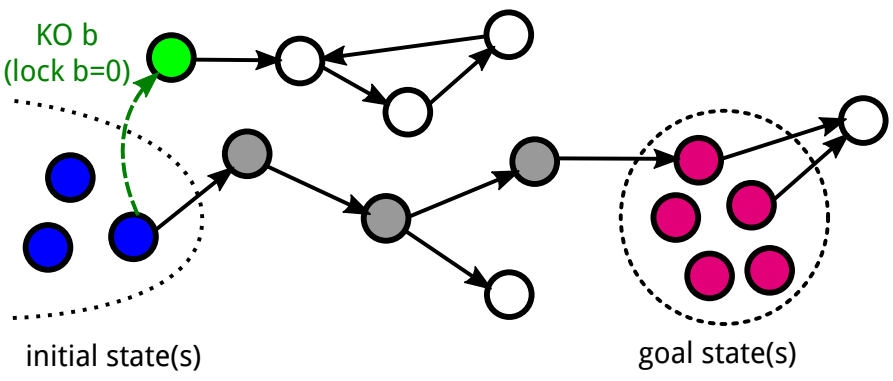
Main features



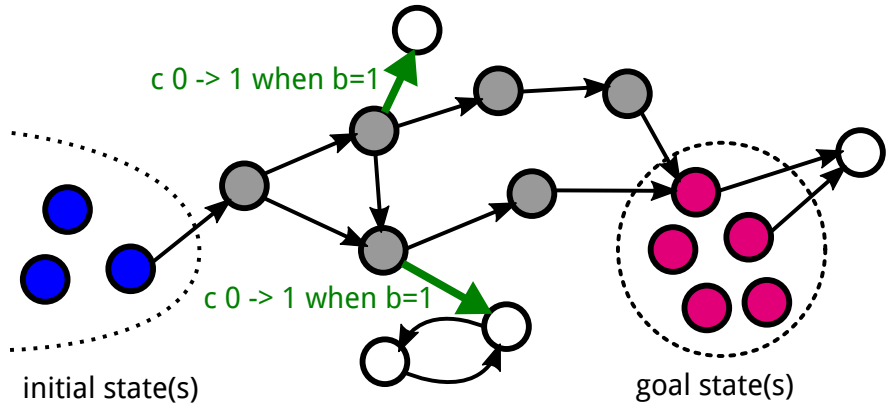
reachability



cut sets



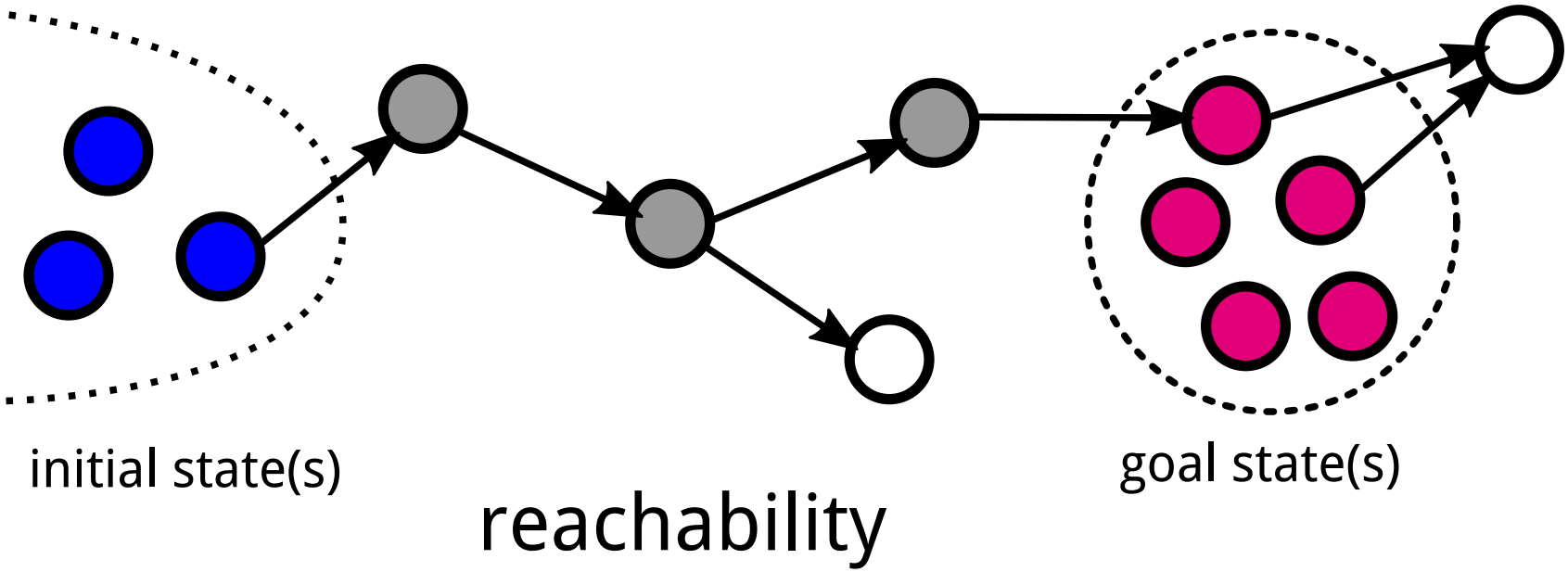
mutations



bifurcations

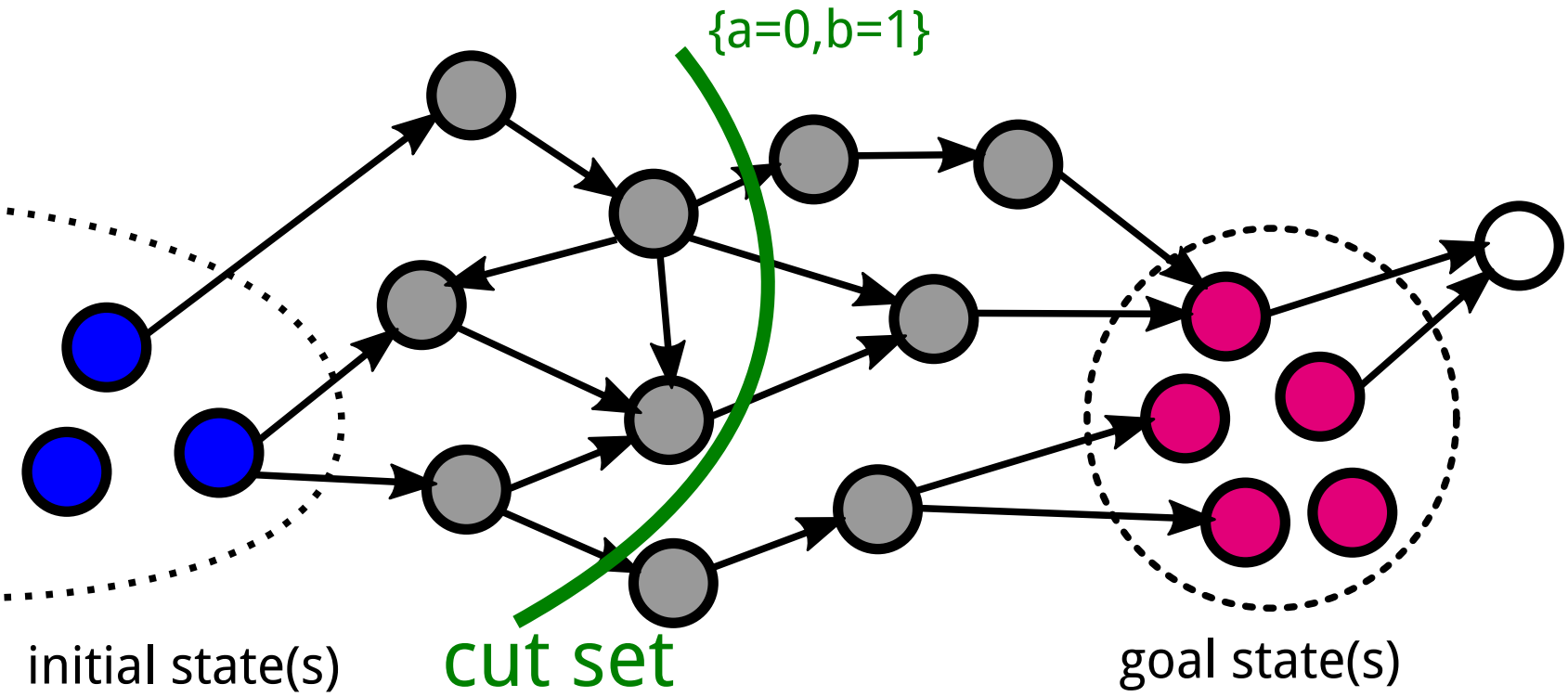
Scalability: networks with 100 - 10,000 components

Main features



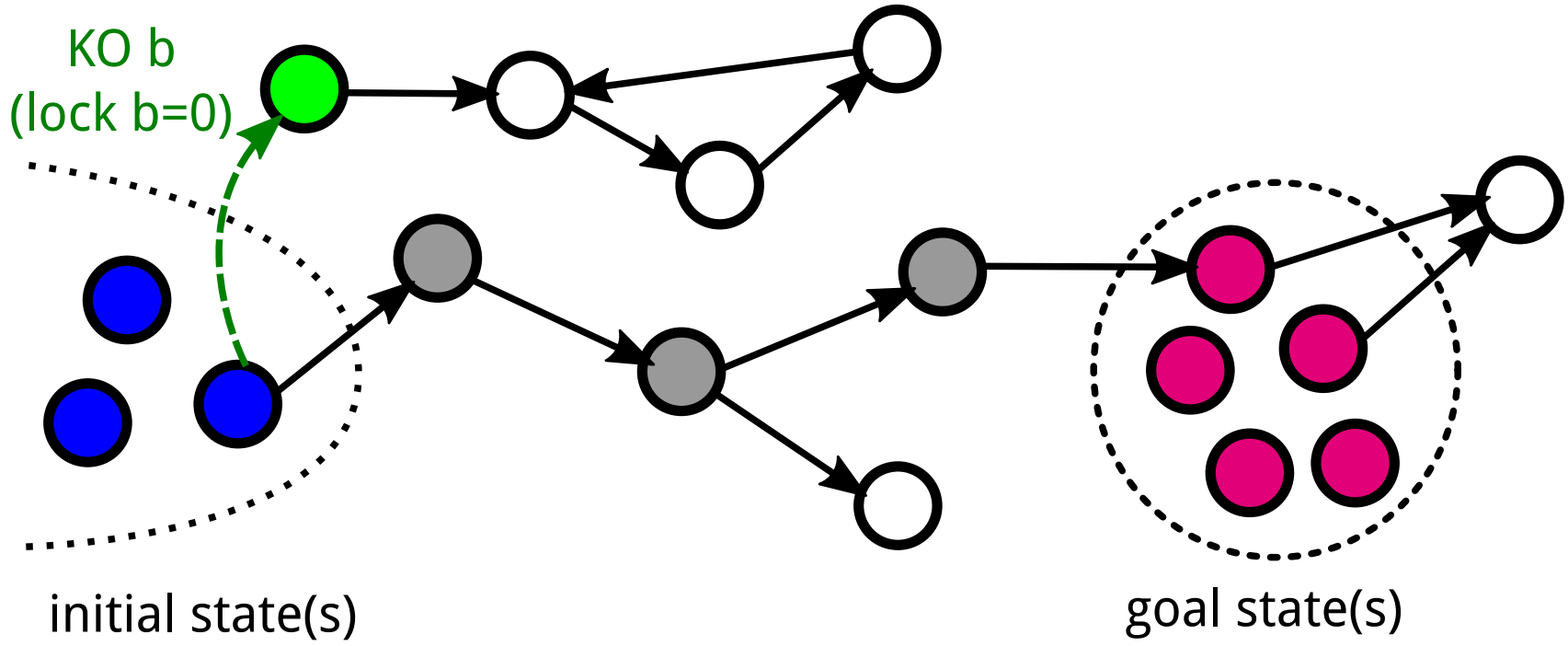
- Static analysis by abstract interpretation
- Model-checking when inconclusive, with model reduction beforehand.

Main features



- Common features of trajectories (necessary steps)
- Computed on abstract structure representing all the acyclic trajectories
- Under-approximation: only correct results; but incomplete.

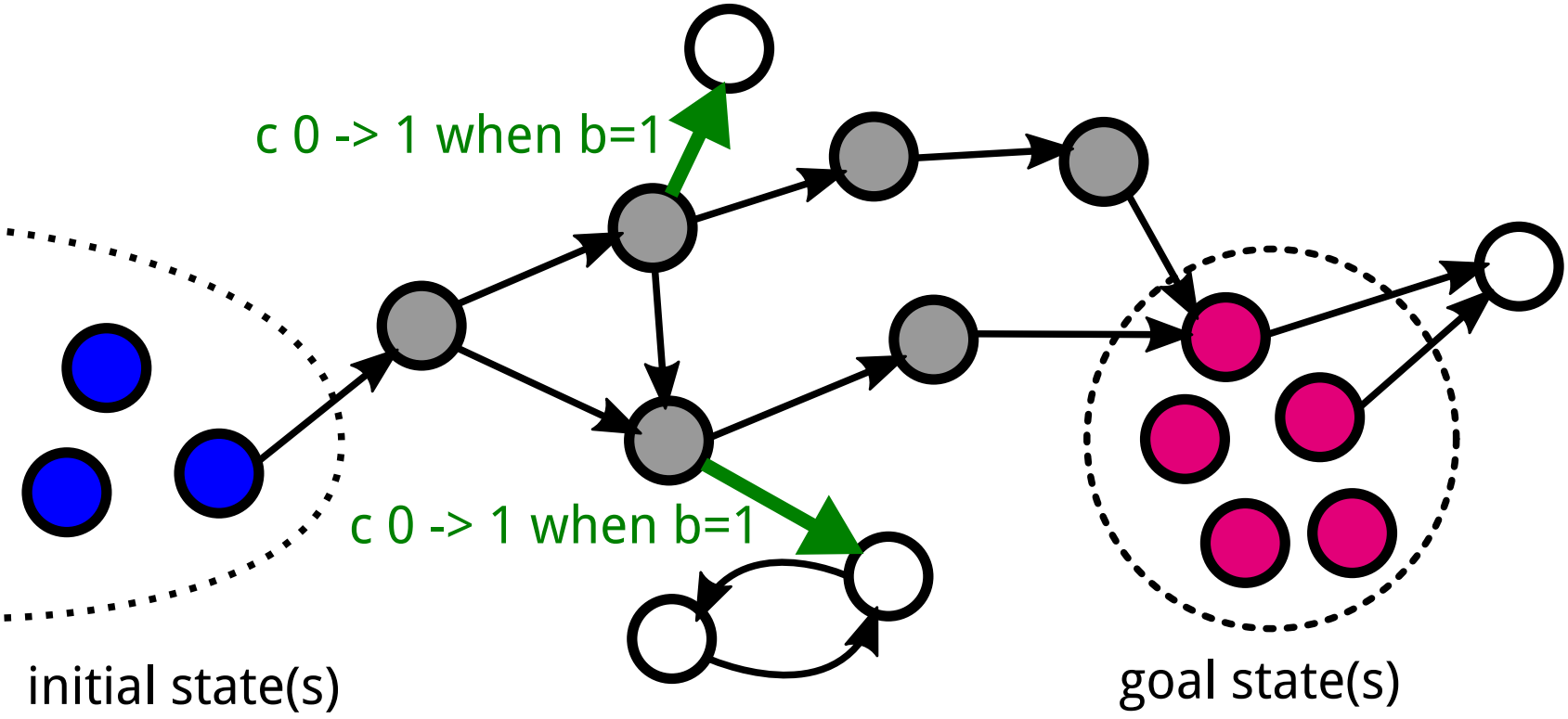
Main features



mutations

- Control of reachability
- Answer-Set Programming implementation
- Under-approximation: only correct results; but incomplete.

Main features



- Key transitions responsible for capability loss (differentiation)
- Answer-Set Programming implementation
- Under-approximation: only correct results; but incomplete.

Behind the scenes

Computational challenge

- Mentionned features are PSPACE-complete
- Avoid screening of mutations/cut-sets/bifurcations candidates

PINT technology

- Static analysis by abstract interpretation
- Avoid building the state transition graph (even symbolically)
  $\Rightarrow$ compute abstractions (Local Causality Graphs)
- Formal approximations (under-/over-approximations);
  reduce to NP (SAT) solving.

[Math. Struct. for Comput. Sci. 2012; CAV'13; TCS 2015; CMSB 2016]

Scalability of implementations

Computations with PINT:

- reachability: OCaml implementation + Answer-Set Programming with clingo
- k-cut sets: OCaml implementation (algorithm on abstract structure)
- k-mutations: ASP with clingo

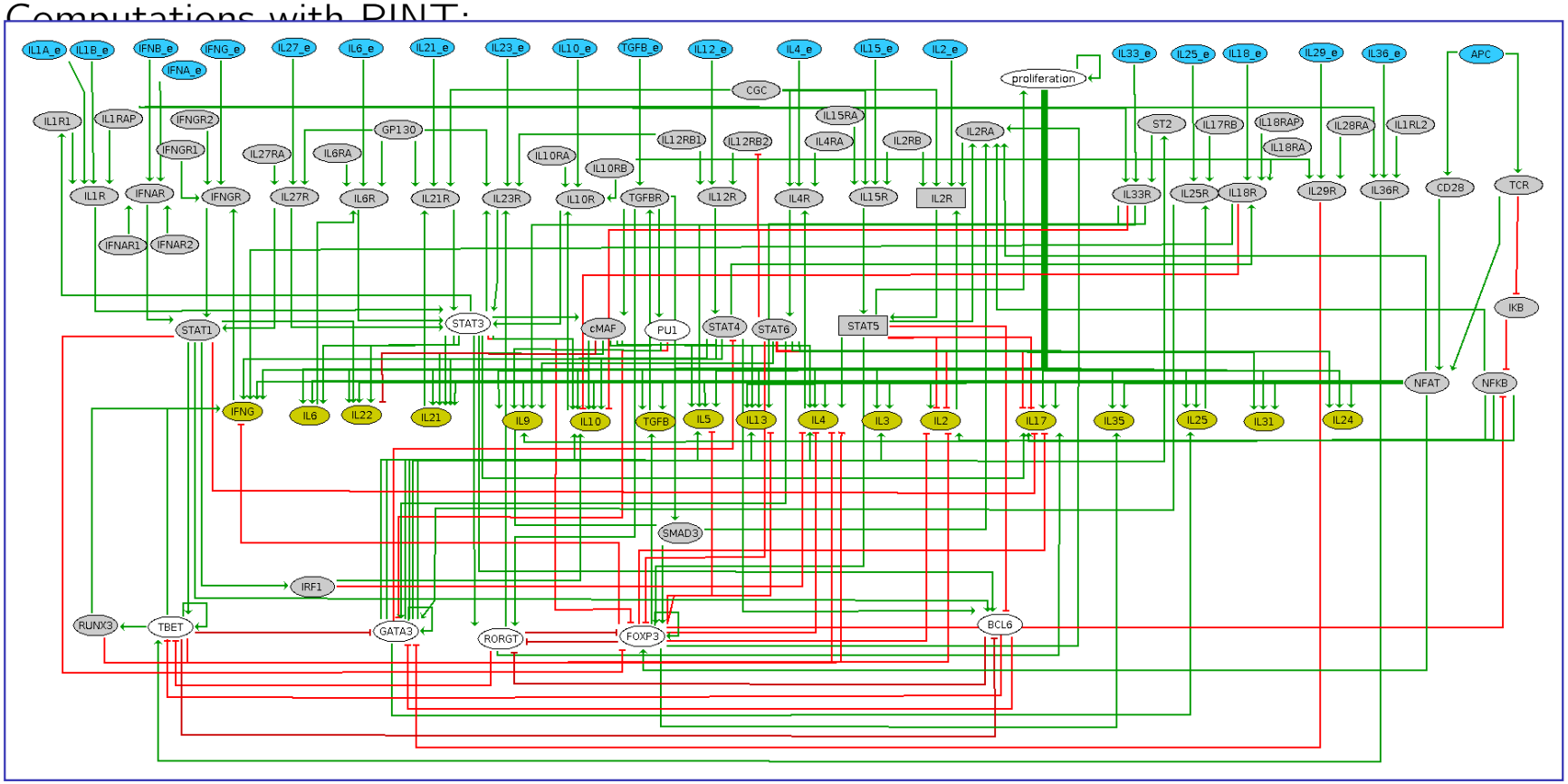
k: limit results to sets with at most k elements.

Goal	TCell-d (101) FOXP3=1		MAPK (309) ERK-PP=1		PID (10,229) SNAIL=1	
NuSMV+PINT red.	8min		KO		KO	
PINT-reach	5s		50s		4min (Inconc)	
4-cut sets	0.10s	101	0.1s	48	5s	37
6-cut sets	0.60s	495	1s	60	10min	907
4-mutations	0.30s	15	10s	1896	50min	67
6-mutations	0.30s	15	Too many		50min	367

TCell-d: Abou-Jaoudé et al. in Frontiers in Bioengin. and Biotech, 2015
MAPK: Schoeberl et al. in Nature biotech, 2002
PID: Pathway interaction database full extraction (Paulevé et al. at CAV 2013).

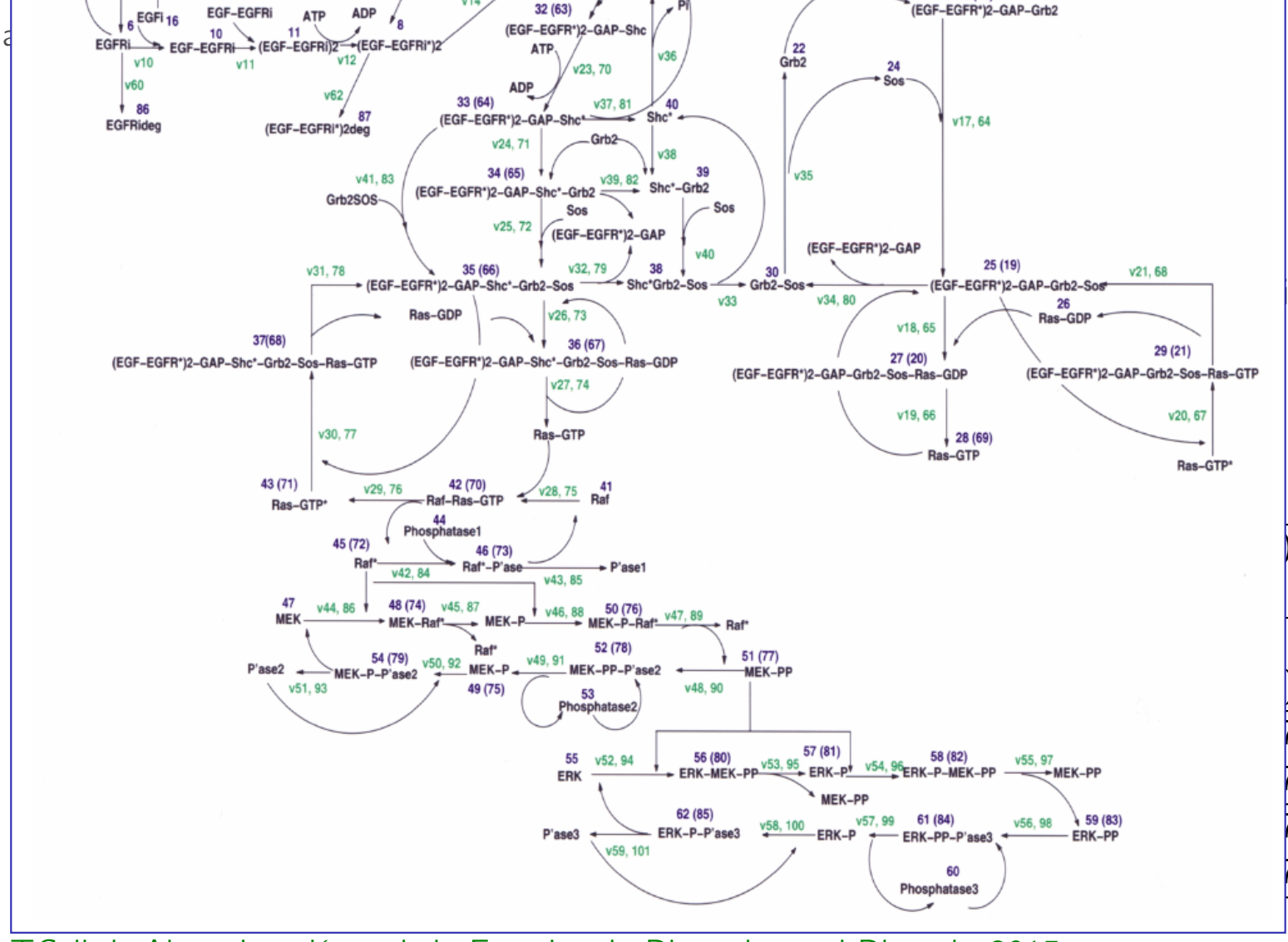
Scalability of implementations

Computations with PINT:



TCell-d: Abou-Jaoudé et al. in Frontiers in Bioengin. and Biotech, 2015
MAPK: Schoeberl et al. in Nature biotech, 2002

PID: Pathway interaction database full extraction (Paulevé et al. at CAV 2013).



TCell-d: Abou-Jaoudé et al. in Frontiers in Bioengin. and Biotech, 2015
MAPK: Schoeberl et al. in Nature biotech, 2002
PID: Pathway interaction database full extraction (Paulevé et al. at CAV 2013).

Scalability of implementations

Computations with PINT:

- reachability: OCaml implementation + Answer-Set Programming with clingo
- k-cut sets: OCaml implementation (algorithm on abstract structure)
- k-mutations: ASP with clingo

k: limit results to sets with at most k elements.

Goal	TCell-d (101) FOXP3=1		MAPK (309) ERK-PP=1		PID (10,229) SNAIL=1	
NuSMV+PINT red.	8min		KO		KO	
PINT-reach	5s		50s		4min (Inconc)	
4-cut sets	0.10s	101	0.1s	48	5s	37
6-cut sets	0.60s	495	1s	60	10min	907
4-mutations	0.30s	15	10s	1896	50min	67
6-mutations	0.30s	15	Too many		50min	367

TCell-d: Abou-Jaoudé et al. in Frontiers in Bioengin. and Biotech, 2015
MAPK: Schoeberl et al. in Nature biotech, 2002
PID: Pathway interaction database full extraction (Paulevé et al. at CAV 2013).

Distribution

`http://loicpauleve.name/pint`

Two interfaces:

- command line tools (binaries) - Linux and Mac OS X
- python interface (pypint module)

Jupyter integration – web application for interactive notebook

- makes easier the reproductibility and documentation of workflows
- becomes standard in data science/bioinformatics
- easy access to statistics/visualisation tools (scipy, pandas, matplotib, R, . . .)

Docker container (Linux, Mac OS X, Windows)

- Contains everything pre-installed, including jupyter web application
- `docker run -d -p 8888:8888 pauleve/pint`
  then go to `http://localhost:8888`.

Demo

Locally on my laptop

There is a public instance of Jupyter/pint at

`http://tmpnb.loicpauleve.name`

- For short tests only (limited number of sessions, limited in time)
- It runs in a virtual machine with limited resources on a personal server