Using MaBoSS for modeling heterogeneous cell behavior

Gautier Stoll, G. Kroemer's Lab

Université Paris Descartes, INSERM, Gustave Roussy

CoLoMoTo meeting (Paris, July 17 2017)

Collaboration with L. Calzone

Outline

1. Population interpretation of probabilistic Boolean space
2. MaBoSS for probabilistic Boolean modelling
3. UpPMaBoSS.pl for dynamical and heterogeneous population modeling
4. Example with (TNF → cell fate) model

1. Population interpretation of probabilistic Boolean space

Boolean node state: 0 or 1 for node *i*.

Boolean network state: vector $\vec{S}$ of Boolean node state for a given network. For each node *i*, $S_i \in \{0,1\}$.

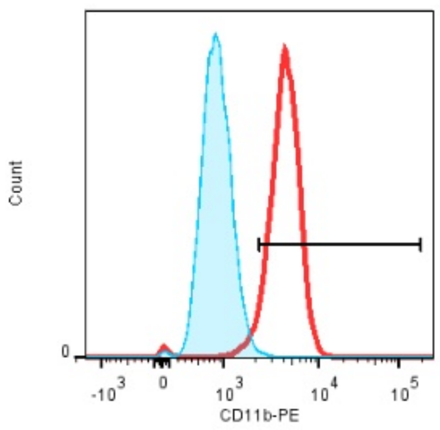
Network state space: set of possible Boolean network state, $\Sigma = \{\vec{S}\}$ for a given network.

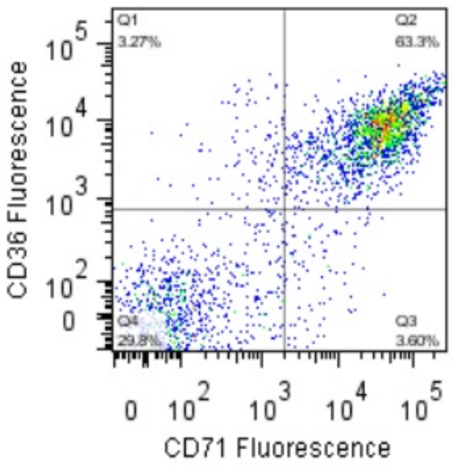
Probability space over network state space: give a probability for each network space.

$$P(\vec{S}) \in [0,1], \sum_{\vec{S} \in \Sigma} P(\vec{S}) = 1$$

1. Population interpretation of probabilistic Boolean space

Biological interpretation of network state probability: ratio of cells (among a population), that are in the given state (described by a Boolean network state). Can be used to confront theoretical prediction to cytometry data.





Missing:

- Global size of cell population
- Paracrine interaction, ie population dependant interaction

ALLCELLS

2. MaBoSS for probabilistic Boolean modelling

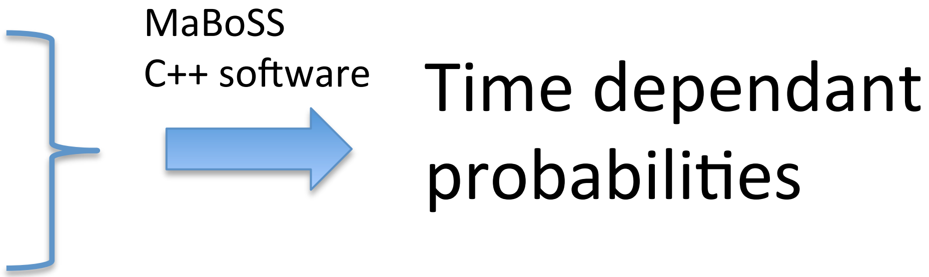
MaBoSS: $$P(\vec{S},t) \in [0,1], \sum_{\vec{S}\in\Sigma} P(\vec{S},t) = 1$$

time dependant probabilities (over network states)

within <u>Markov hypothesis</u>.

Inputs:

- Transition rates
- Initial conditions



MaBoSS
C++ software

Time dependant probabilities

Transition rate: $$\rho_{\vec{S}\to\vec{S}'} = \lim_{\Delta t\to 0} \frac{P[\vec{S}'(\Delta t) \mid \vec{S}(0)]}{\Delta t} \in [0,\infty[$$

2. MaBoSS for probabilistic Boolean modelling

Two files that define the model: a **bnd file** and a **cfg file**.

1) Transition rate, define rate value (time^{-1}) and condition for activation and inhibition: logical operators (AND,OR, XOR, NOT) with real operators (+, -, *, /) and ternary conditional (? :).

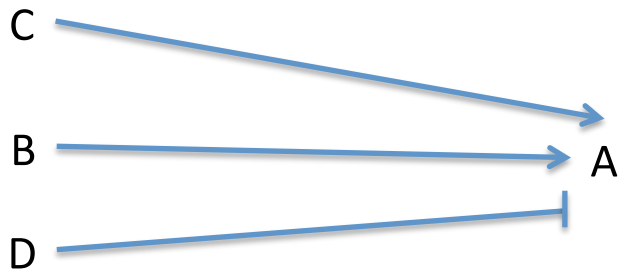
In **bnd file**:

```
Node A {
      rate_up  = (C AND B OR NOT D) ? 1.0 : 0.0;
      rate_down  = (C AND B OR NOT D) ? 0.0 : $Extern_variable; }
```

(condition ? value_if_cond_true : value_if_cond_false)

In **cfg file**:

```
$Extern_variable = 30;
```



2) Initial condition:

p[(A,B,C,D)=(0,0,0,0)]=0.7

p[(A,B,C,D)=(1,0,0,0)]=0.2

p[(A,B,C,D)=(1,1,1,1)]=0.1

In **cfg file**:

```
[A,B,C,D].istate = 0.7 [0,0,0,0] , 0.2 [1,0,0,0] , 0.1 [1,1,1,1];
```

3. UpPMaBoSS.pl for population modeling

The model is initially based on signalling pathways inside each cell, to which nodes representing cell death and cell division are added and inter-cell communications are specified.

Population dynamics using MaBoSS include:

1. Cell death,
2. Cell division,
3. Inter-cell communication (ligand → receptor for instance).

Possibility to model mutant effect on population dynamics for instance.

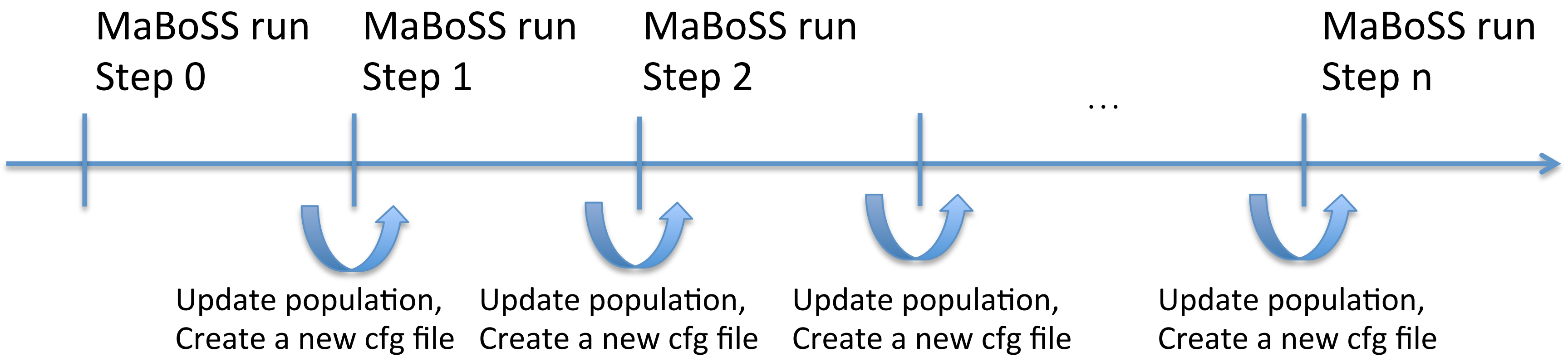
Outputs:

- size of cell population over time,
- probabilities of “network states” over time

UpPopMaBoSS.pl

Run MaBoSS several times (“steps”). For each MaBoSS run:

1. take the previous condition for “*Population update*”: cell death, cell division and inter-cell effects,
2. create a new **cfg file** for the following step with a new initial condition.



- New cfg file: take initial condition from final probability distribution of previous step, remove cell death state, double division state, update external variables that represent inter-cell interaction
- Total time = (max_time of a MaBoSS run) * (number of steps).

Outputs

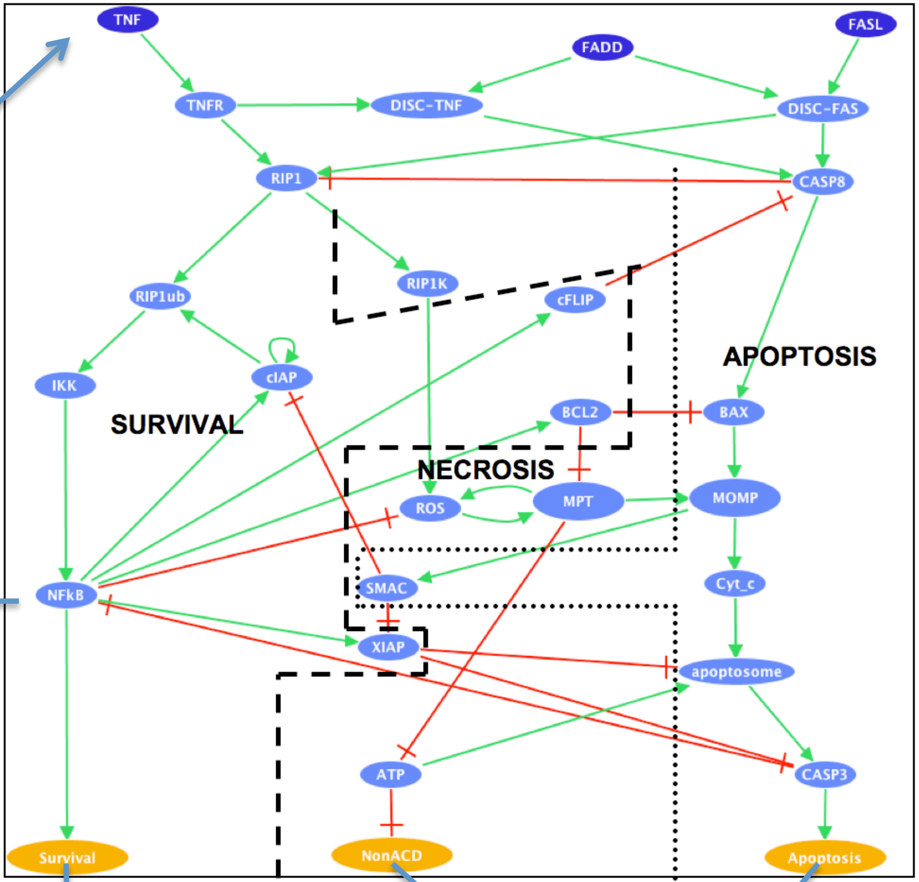
1) MaBoSS output csv files (probabilities over network states over time).
2) Final probability distribution over network state for each step/MaBoSS run.
3) Population ratio for each step/MaBoSS run.

The “population ratio” is a number that represents the relative size of the cell population compared to the initial population size. Therefore:

- Population ratio = 1 → Stable size
- Population ratio > 1 → Growing population
- Population ratio < 1 → Decreasing population

Example with TNF → cell fate

Use published cell fate model (GINsim/MaBoSS), available on the web page



Production of TNF induced by NFkB

Cell Division / Proliferation

Cell Death

In the **bnd file**:

- add a node for cell death, a node for cell division
- modify the TNF node, in order that its activation is controlled by an external variable ($TNF_induc).

```
node TNF
{ rate_up = $TNF_induc;
  rate_down = $Degr_TNF;}
```

TNF can be induced by setting the initial condition or controlling the external variable $TNF_induc

```
node Proliferation
{ rate_up = Survival ? $DivRate : 0.0;
  rate_down = 0.0;}
```

Survival induces cell division, and thus proliferation

```
node Death
{ rate_up = (NonACD | Apoptosis) ? 1 : 0.0;
  rate_down = 0.0; }
```

Cell death is induced by either Apoptosis or NonACD (non-apoptotic cell death)

In the initial **cfg file** (used for step 0):

- set cell division ($DivRate) (cell death rate is 1 hour in the bnd file)
- define an external variable that control the strength of TNF production by NFkB ($ProdTNF_NFkB)
- set carefully the max_time (tricky)
- set the initial condition for step 0

```
[TNF].istate = 1 [1] , 0 [0];
$ProdTNF_NFkB = 1;
$TransRate = 1/24;
$DivRate = 1/24;
$Degr_TNF = 1/6;
$TNF_induc = 0;
max_time = 3;
```

Treat cells with TNF → initial state is 1

Production of TNF by NFkB is switched on

Mean transcription time = 24 hours
Mean Division time = 24 hours
Mean TNF degradation time = 6 hours

TNF is only induced by initial condition (not by NFkB initially)

Time for population update = 3 hours (tricky to set up)

In the **upp file**:

- Define the cell death node, the division node, the number of steps.
- Define how the external variable that control TNF is updated (according to NFkB status).

Node for cell death

Node for cell division

External variable $TNF_induc is updated according to state probabilities at the end of each MaBoSS run

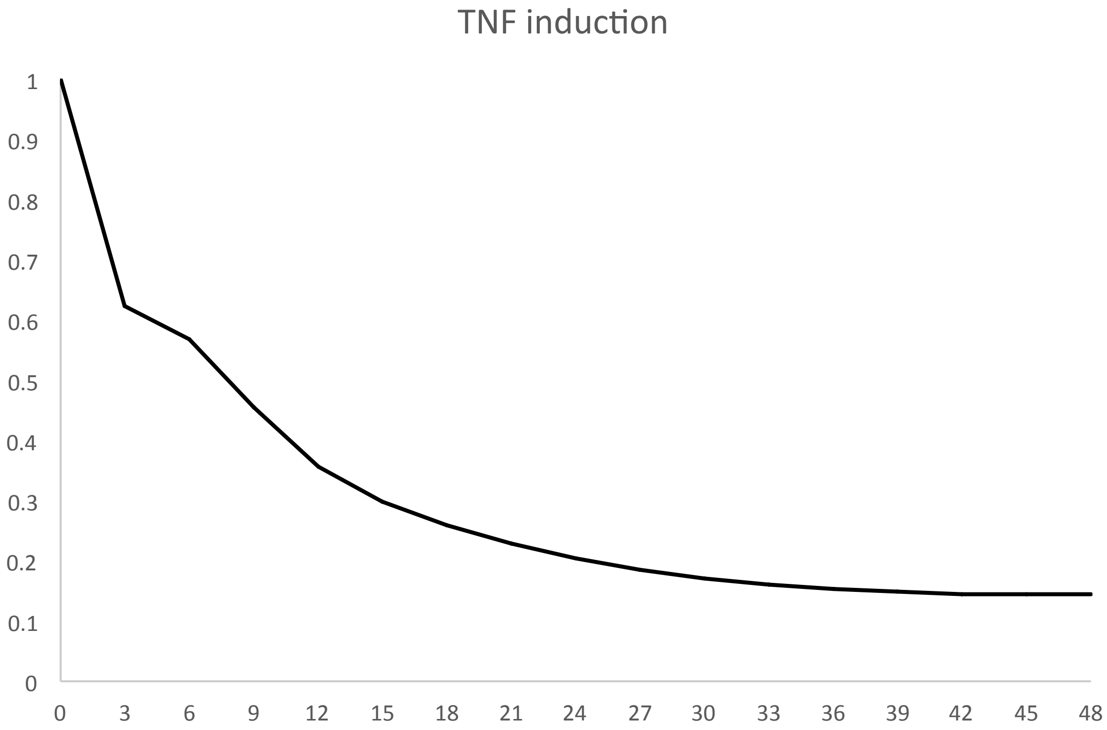
```
death = CellDeath;
division = Proliferation;

$TNF_induc u= $ProdTNF_NFkB*p[(NFkB,Death) = (1,0)];

# u= means update of the external variable

# NB: more than one variable can be defined as such
# (here representative of the microenvironment)

# $ProdTNF_NFkB corresponds to the strength of the feedback
# that way it is easier to cancel the feedback when put to 0

# 16 steps are chosen to account for 48 hours because max_time = 3

steps = 16;

MaBoSS = ./MaBoSS
```

Update operator

Number of steps → total time = 16*3 hours

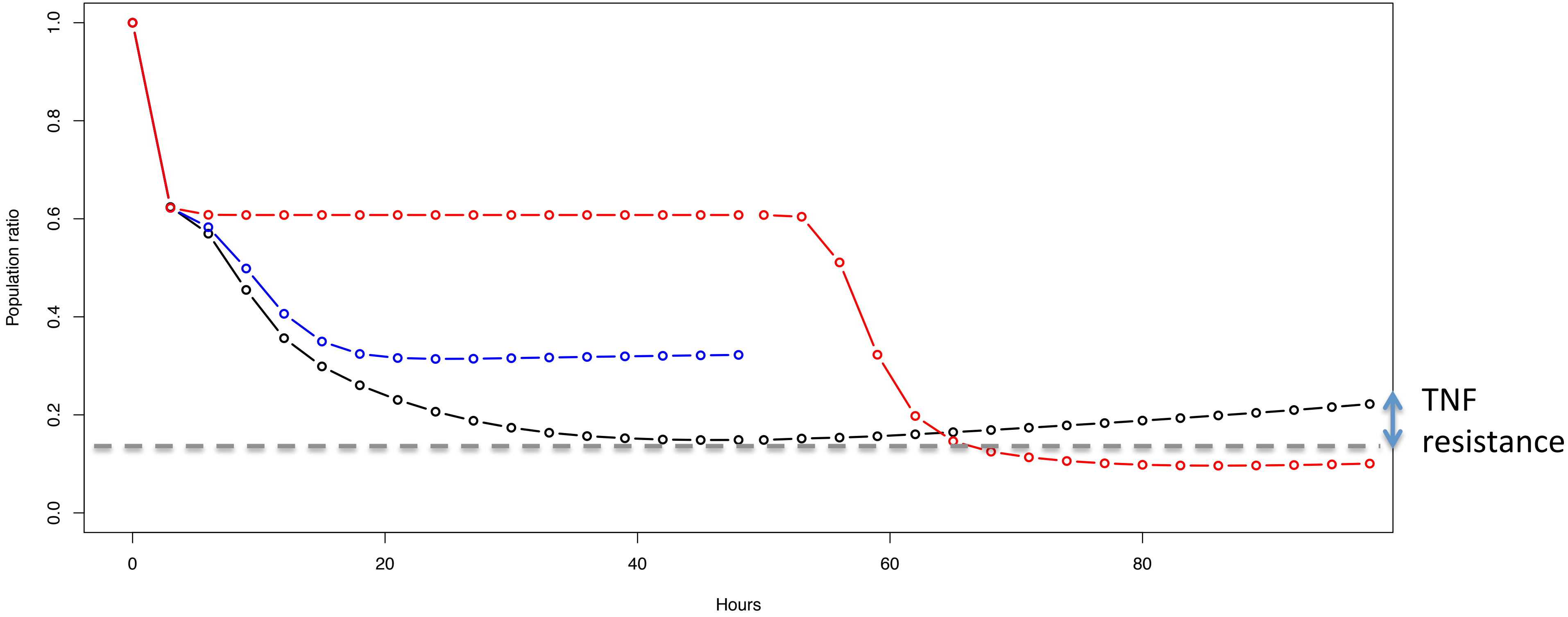
MaBoSS executable name

Population ratio of TNF induction model



```
$ProdTNF_NFkB = 1;
$TNF_induc = 0;
[TNF].istate = 1 [1] , 0 [0];
```

TNF resistance from population ratio over time

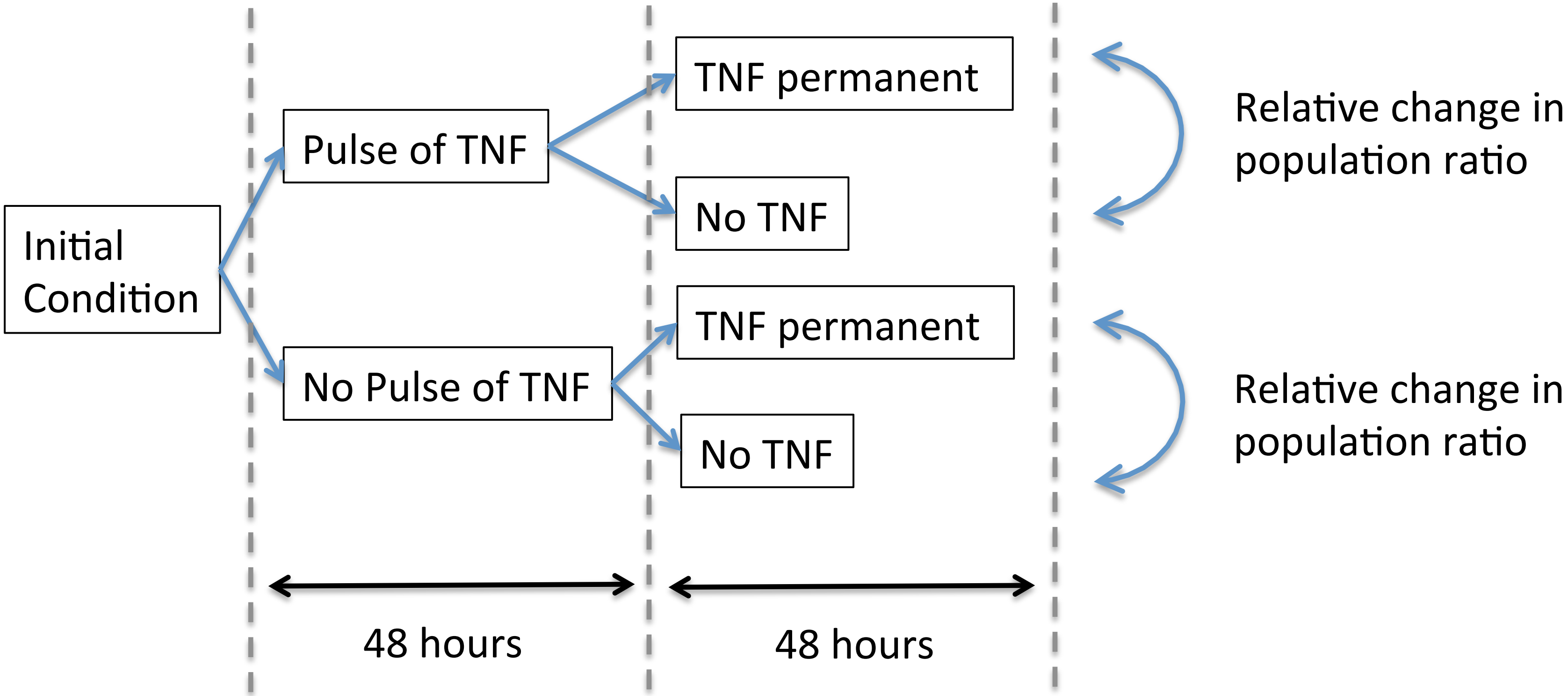


No TNF / then TNF permanent induction
$ProdTNF_NFkB = 1;
$Degr_TNF = 1/6;
$TNF_induc = 0; (at 0 hours)
$TNF_induc = 20; (at 48 hours)
[TNF].istate = 0 [1] , 1 [0]; (at 0 hours)
[TNF].istate = 1 [1] , 0 [0]; (at 48 hours)

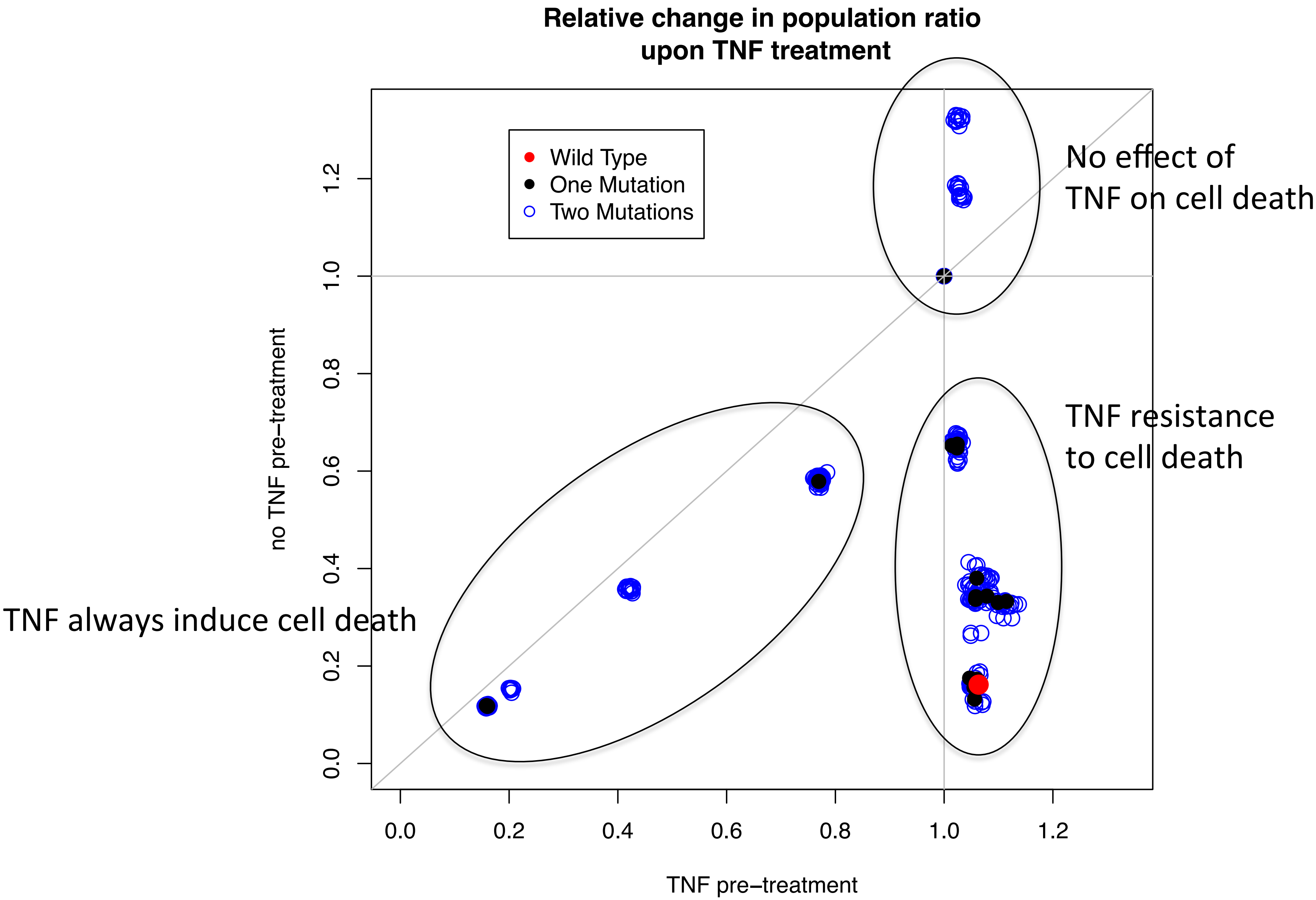
TNF (pulse) then TNF permanent induction
$ProdTNF_NFkB = 1;
$Degr_TNF = 1/6;
$TNF_induc = 0; (at 0 hours)
$TNF_induc = 20; (at 48 hours)
[TNF].istate = 1 [1] , 0 [0]; (at 0 hours)
[TNF].istate = 1 [1] , 0 [0]; (at 49 hours)

TNF (pulse)
No TNF feedback from NFkB
$ProdTNF_NFkB = 0;
$Degr_TNF = 1/6;
$TNF_induc = 0;
[TNF].istate = 1 [1] , 0 [0];

In silico “Experimental” protocol



Simulation for every single and double mutations



No effect of
TNF on cell death

TNF resistance
to cell death

TNF always induce cell death

Perspective

- Study time step dependance
- Apply UpPMaBoSS.pl to C. Hernandez model of T-cell differentiation
- Apply UpPMaBoSS.pl to Immungenic Cell Death
- Asynchronous population update?

Acknoledgements

- D. Thieffry and C. Hernandez
- E. Viara