

Modélisation booléenne probabiliste avec MaBoSS, modélisation en population de cellules avec UpPMaBoSS

Gautier Stoll

Université Paris Descartes, Centre de Recherche des
Cordeliers, INSERM U1138/eq. Kroemer, Gustave Roussy

ASSTABIO: Rennes, 22 et 23 Mars 2018

Outline

1. Population interpretation of probabilistic Boolean space
2. MaBoSS for probabilistic Boolean modelling
3. UpPMaBoSS for dynamical and heterogeneous population modelling
4. Example with (TNF \rightarrow 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.

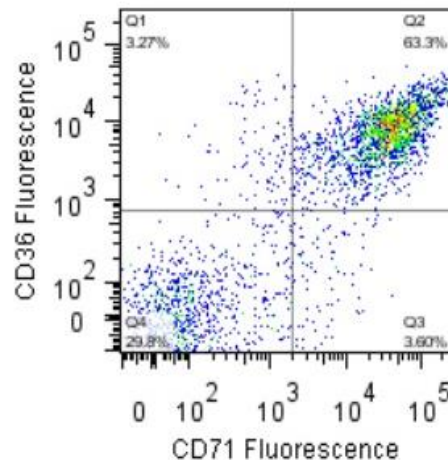
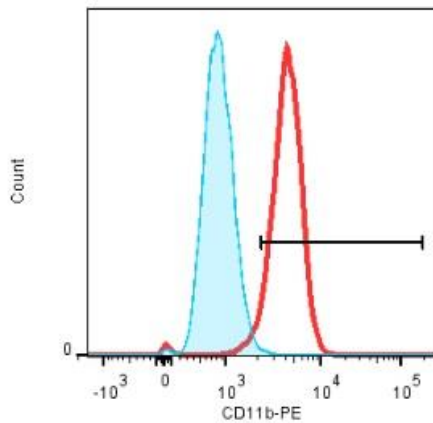
Probability space over network state space (not on Boolean node state): give a probability for each network space.

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

1. Population interpretation of probability space over network state 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

2. MaBoSS for probabilistic Boolean modelling

<https://maboss.curie.fr/>

Motivation:

- Probabilistic interpretation of node/network states
- Model transient behavior
- Model asymptotic behavior
- Handle different time scale processes (transcription, phosphorylation, etc.)

→ Fill the gap between ODE and Boolean modeling:
continuous Markov process on Boolean network state space

2. MaBoSS for probabilistic Boolean modelling

<https://maboss.curie.fr/>

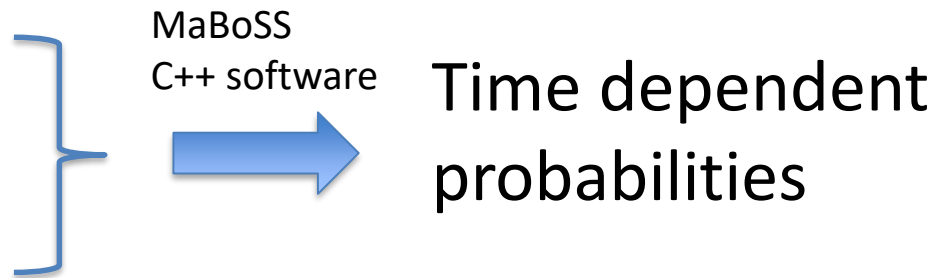
MaBoSS: $P(\vec{S}, t) \in [0, 1], \sum_{\vec{S} \in \Sigma} P(\vec{S}, t) = 1$

time dependent probabilities (over network states)

within Markov hypothesis, with asynchronous transitions
(single node transition)

Inputs:

- Transition rates
- Initial conditions



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

2. MaBoSS for probabilistic Boolean modelling

Two files that define the model: a **bnd file** (*boolean network description*) and a **cfg file** (*configuration file*).

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

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;}  
}
```

In **cfg file**:

```
$Extern_variable = 30;
```

2) Initial condition:

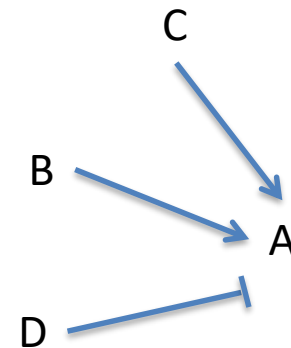
$p[(A,B,C,D)=(0,0,0,0)]=0.7$

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

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

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];
```



Modeling parameters in .cfg file: number of trajectories, maximum time, etc.

2. MaBoSS for probabilistic Boolean modelling

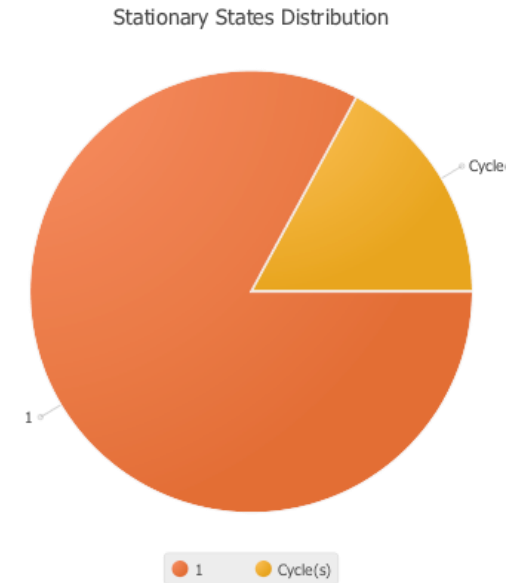
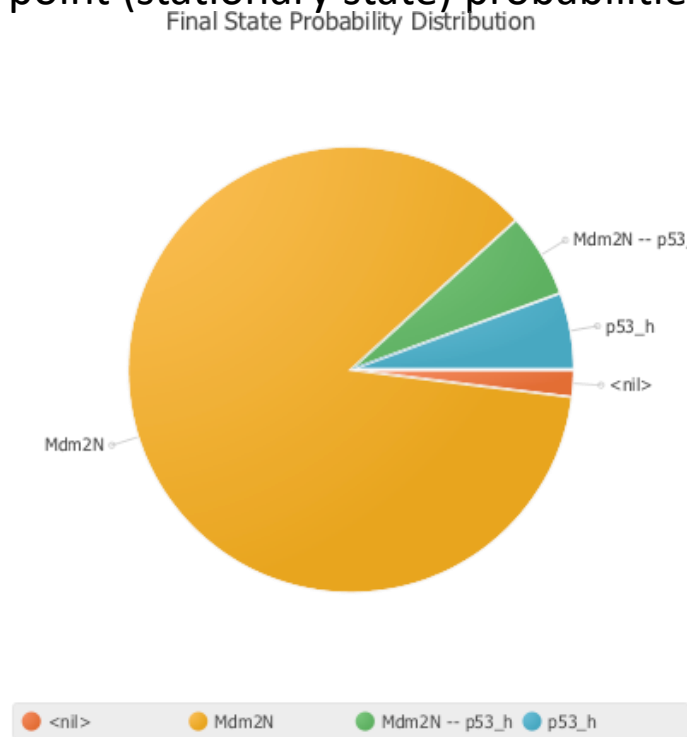
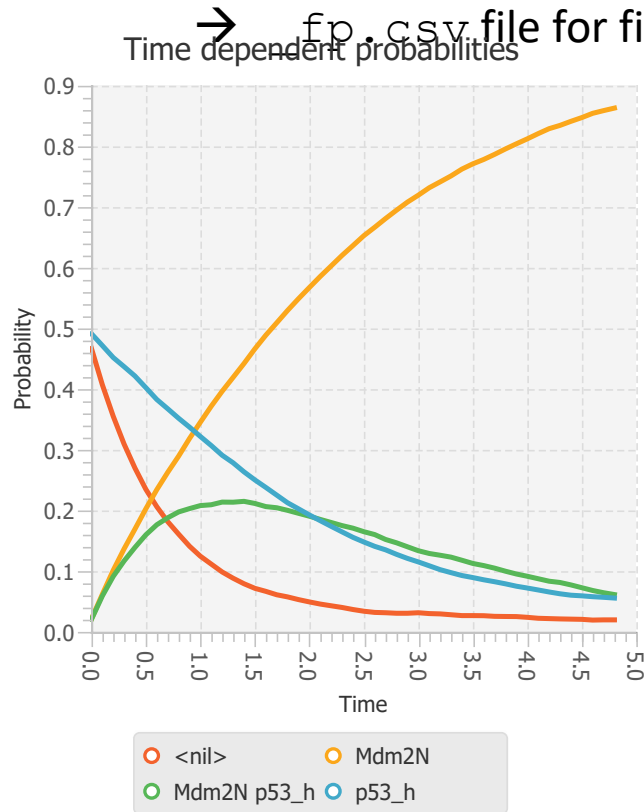
Core MaboSS c++ executable file (unix):

```
'MaBoSS -c configuration.cfg -o out_prefix network.bnd'
```

→ `_traj.csv` file for transient probabilities

→ `_statdist.csv` file for stationary distribution decomposition (not used a lot)

→ `_fp.csv` file for fix point (stationary state) probabilities



Careful: final states is a combination of stationary states and limit cycles.

2. MaBoSS for probabilistic Boolean modelling

Tool Box (MaBoSS-env, download from <https://maboss.curie.fr>):

- MBSS_FormatTable.pl for producing tables for spreadsheet
- MBSS_PieChart.py for pie chart of final distribution
- MBSS_TrajectoryFig.py for plot of probability trajectories
- MBSS_MutBndCfg.pl for handling mutation: add external variable in .bnd file that are controlled in .cfg file.
- MBSS_DrugSim.py for modeling drug effect

Tutorials and examples at <https://maboss.curie.fr>

3. UpPMaBoSS for population modeling

A Boolean model is initially based on signalling pathways inside each cell. 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 \rightarrow receptor for instance).

Possibility to model mutant effect on population dynamics for instance, because it is based on a signalling pathway.

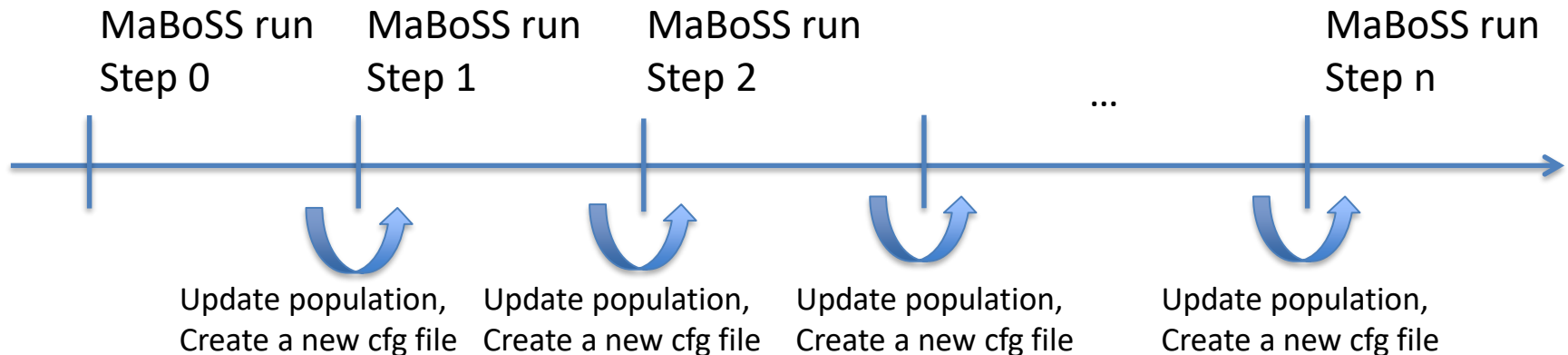
Outputs:

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

UpPMaBoSS, “Update Population MaBoSS”

Perl script that runs MaBoSS several times (“steps”). For each MaBoSS run:

1. Extract the final network state probability distribution.
2. Create a new **cfg file** (configuration file) for the new step.



- New cfg file: take initial condition from final probability distribution of previous step, remove cell death state, double division state, renormalize probability distribution, 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 (updated according to probability renormalization factor)

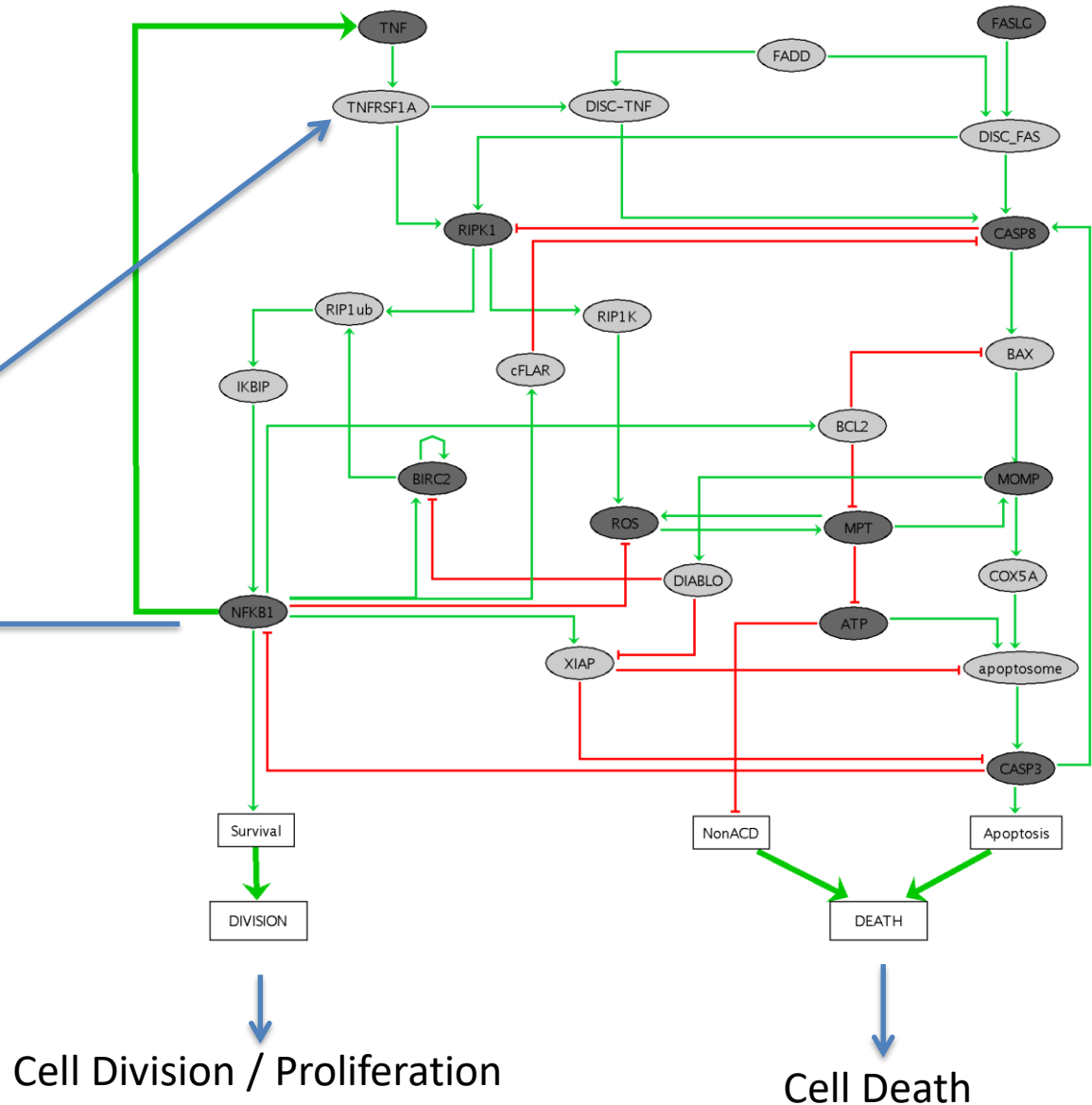
The “population ratio” is a number that represents the relative size of the cell population compared to the initial population size. Therefore:

- 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 MaBoSS web page, <https://maboss.curie.fr>

Production of TNF induced by NFκB



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 (pulse of TNF) 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
(shot of TNF)

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 induced by initial condition,
but this external variable will be
updated (TNF permanent)

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).
-

```
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
```

Node for cell death

Node for cell division

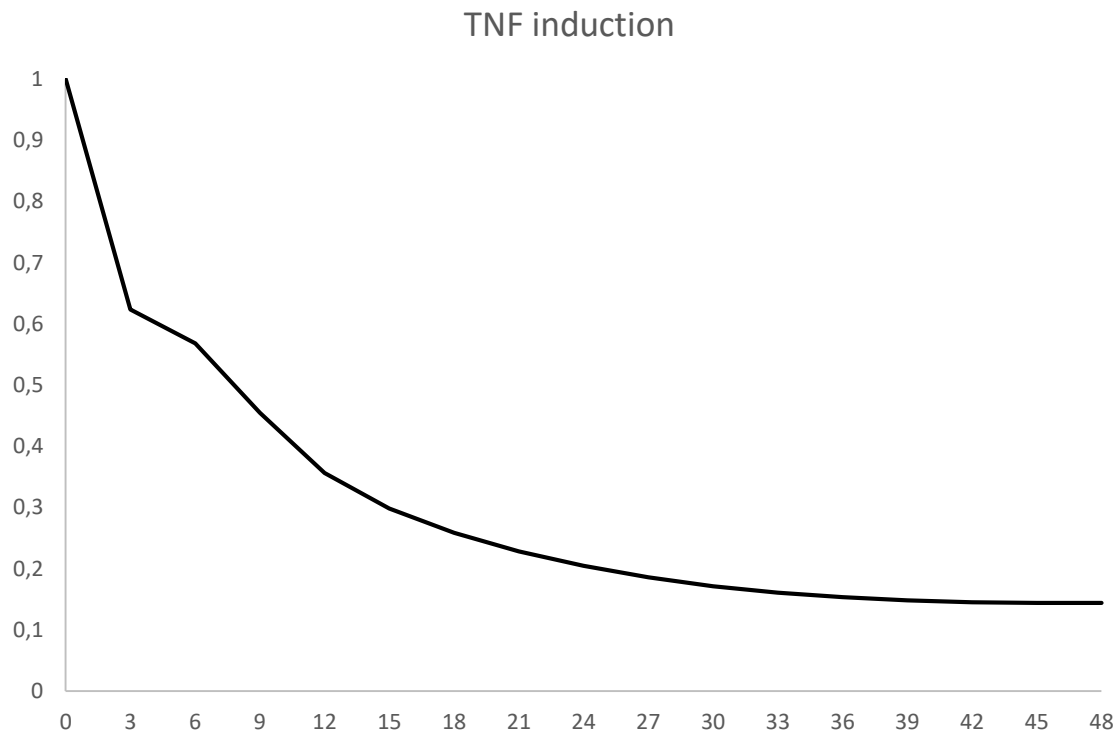
External variable \$TNF_induc is updated according to state probabilities at the end of each MaBoSS run

Update operator

Number of steps → total time = 16*3 hours

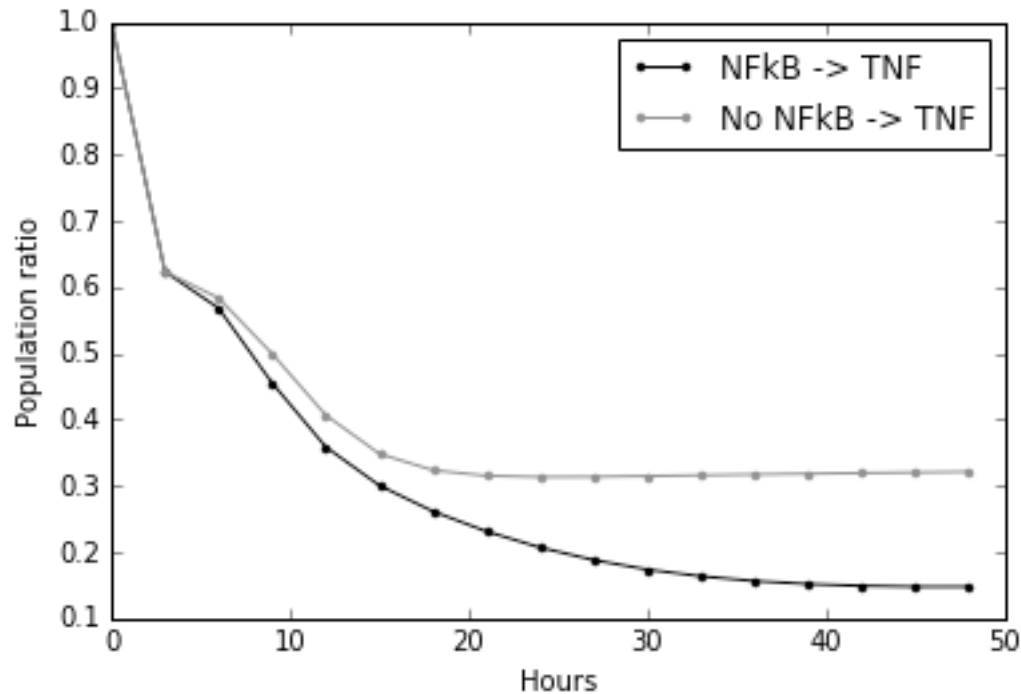
MaBoSS executable name

Population ratio of TNF induction model



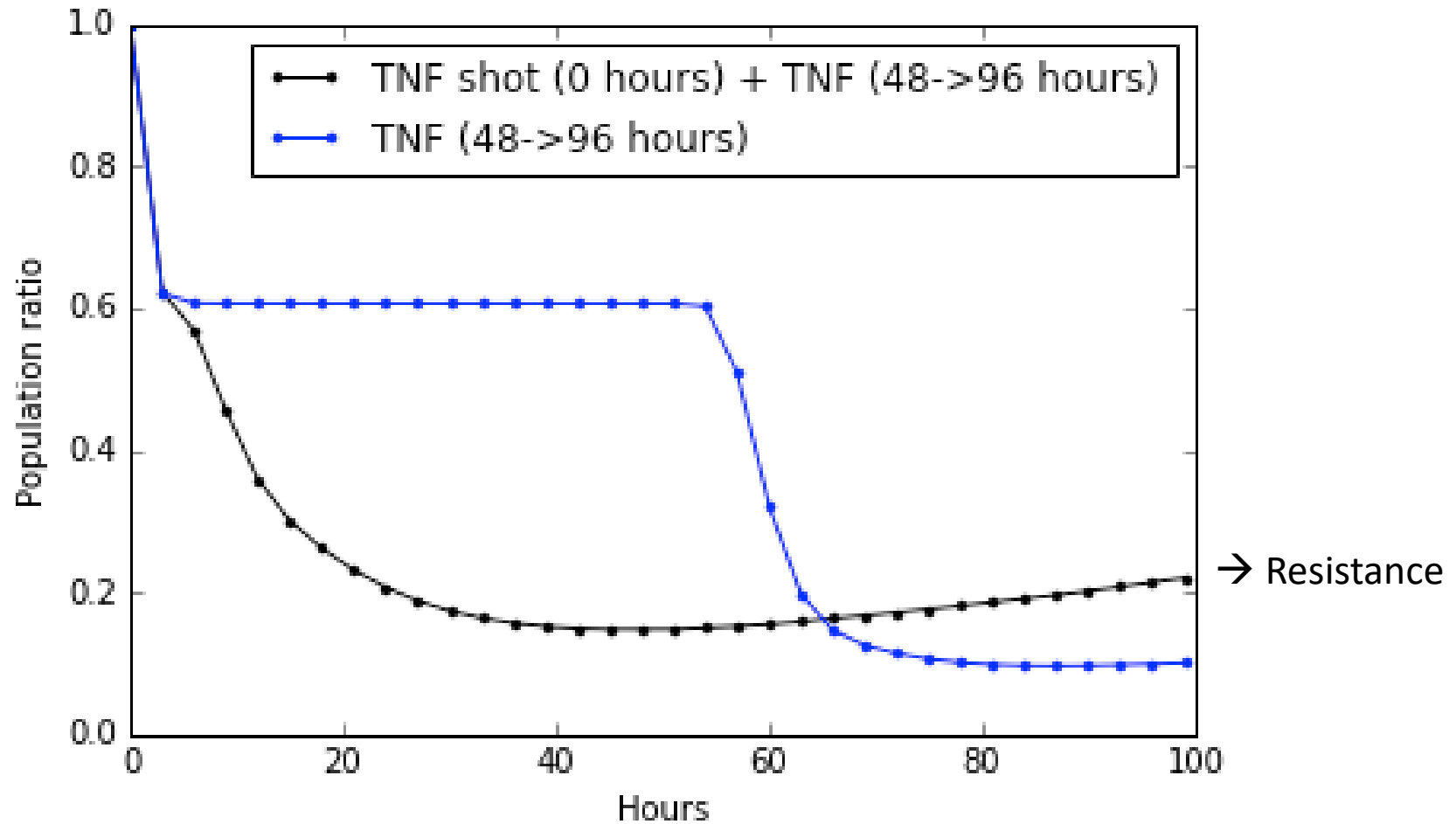
```
$ProdTNF_NFkB = 1;  
$TNF_induc = 0;  
[TNF].istate = 1 [1] , 0 [0];
```

Population ratio with/without paracrine loop



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

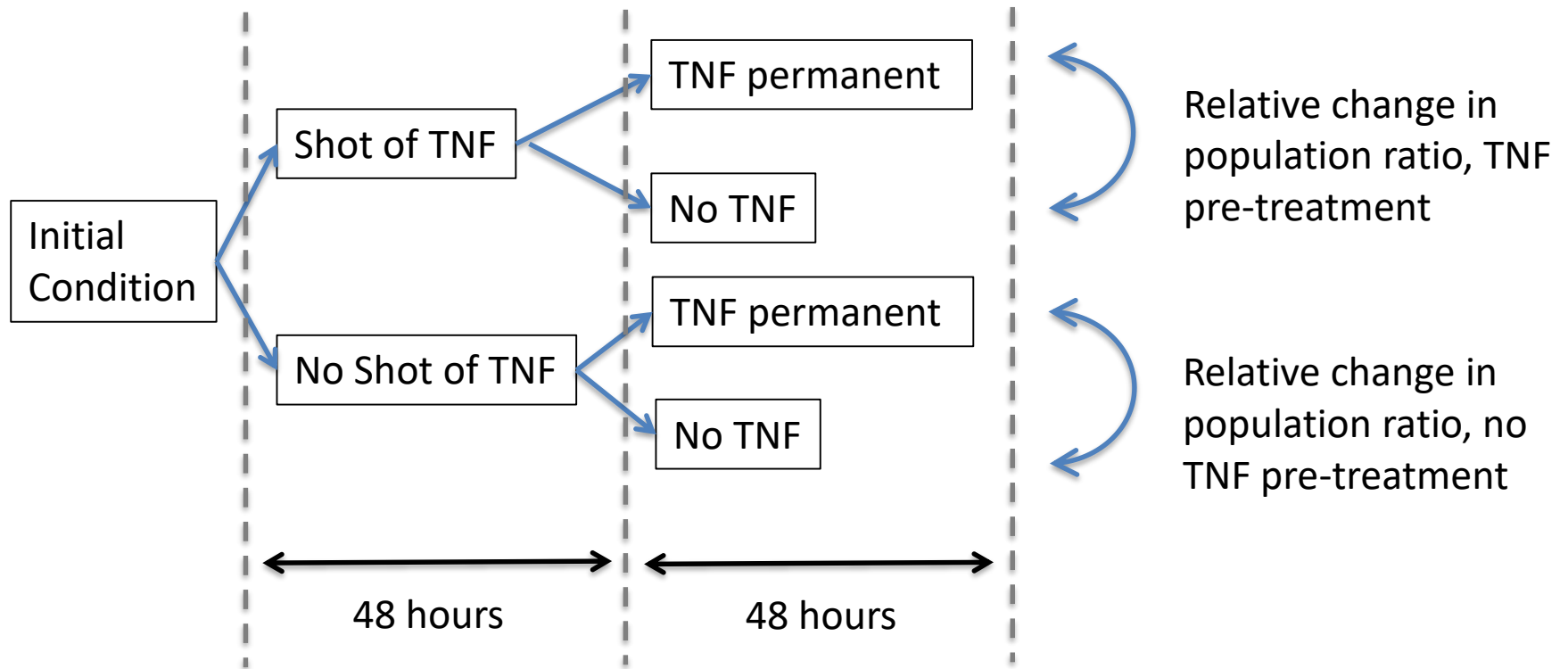
Population ratio upon different TNF treatment protocols



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

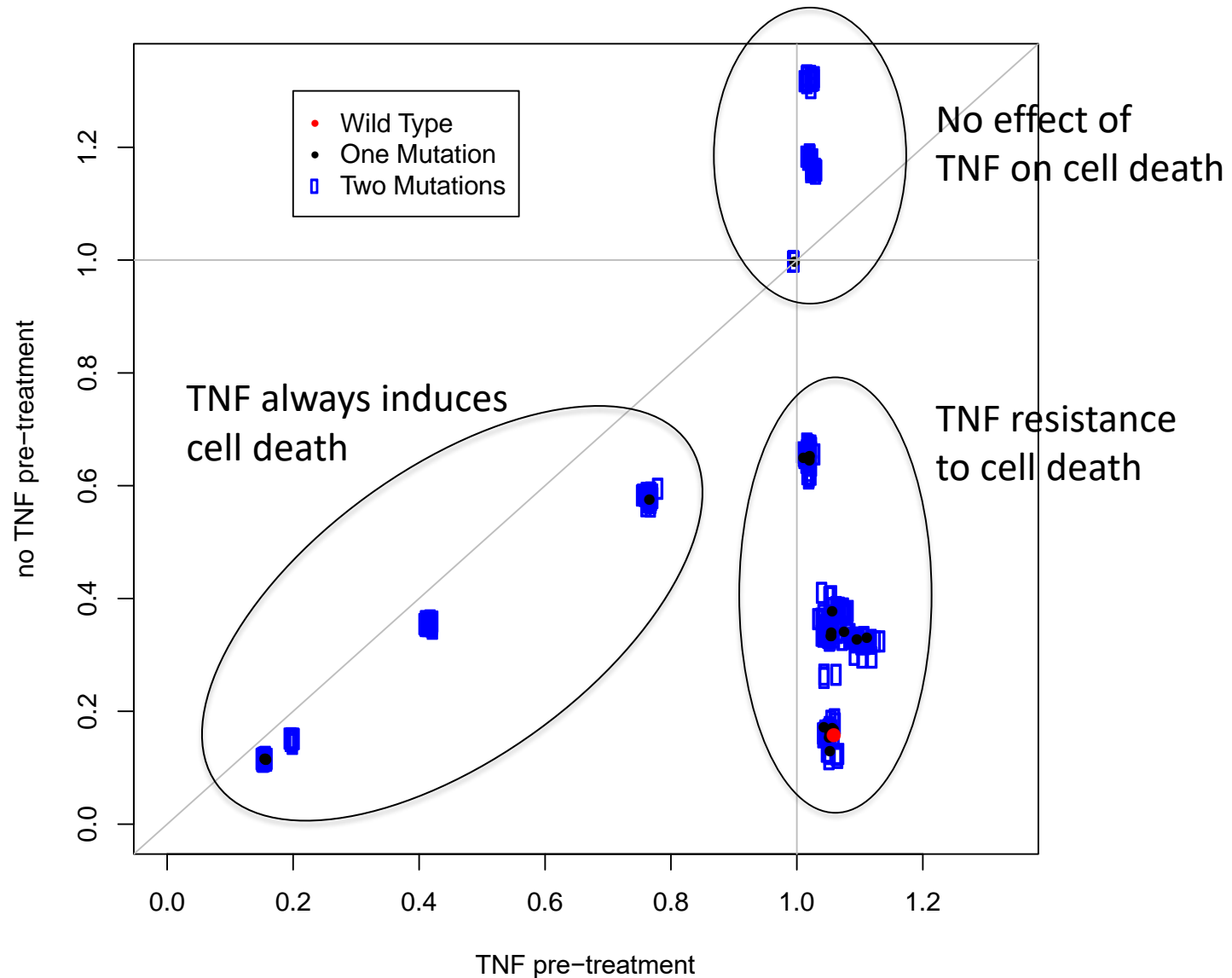
```
$ProdTNF_NFkB = 1;  
$TNF_induc = 1;  
[TNF].istate = 1 [1] , 0 [0];
```

In silico “Experimental” protocol for testing TNF resistance



Simulation for every single and double mutations

Change in population ratio upon TNF treatment, single and double mutants.



Perspective

- Study time step size dependence
- Develop ligand → receptor non-linear models
- Apply UpPMaBoSS.pl to Immunogenic Cell Death
- Asynchronous population update (PopMaBoSS)?

Acknowledgements

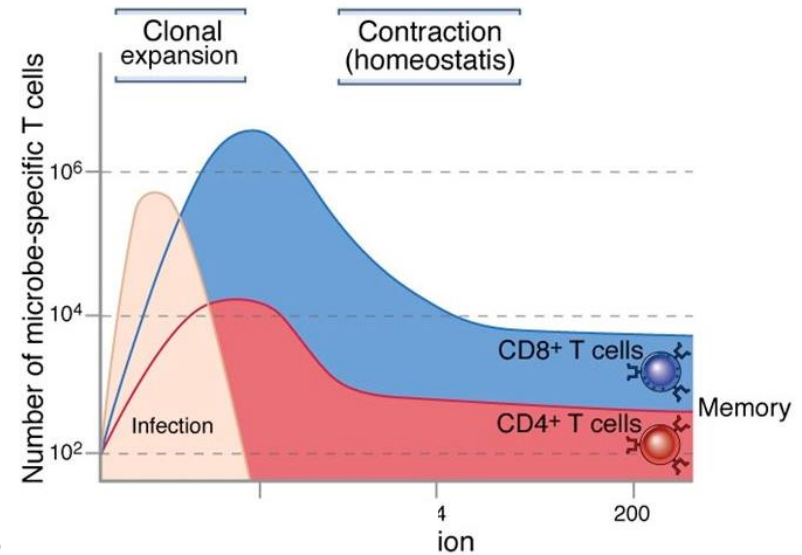
- D. Thieffry and C. Hernandez (ENS, Paris)
- E. Viara (MaBoSS developer)
- L. Calzone (Institut Curie Paris)

5. Example with T-cell activation/differentiation

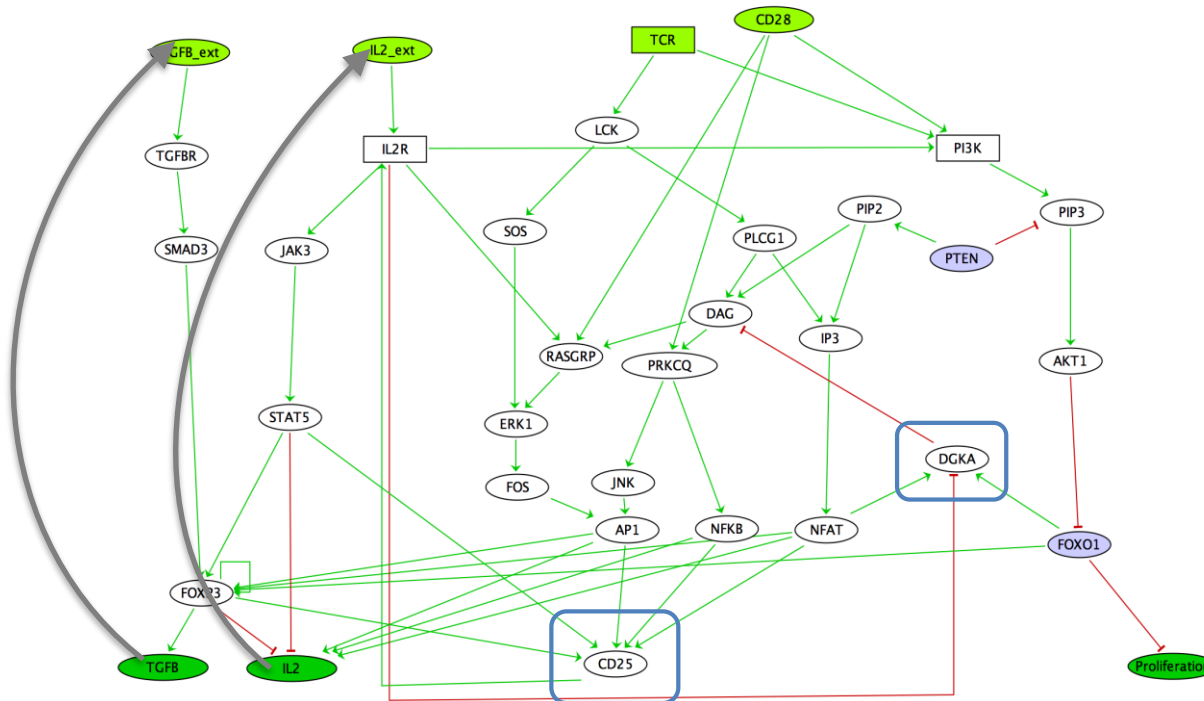
Céline Hernandez (ENS, D. Thieffry group) model.

Observed phenotype (clonal expansion):

- A transient activation of TCR (T-cell receptor, induced by infection) induces T-cells proliferation, that is amplified by IL2 (Interleukin-2).
- TGFbeta induces T-cell differentiation → ~ 10% of Treg (FOXP3+)



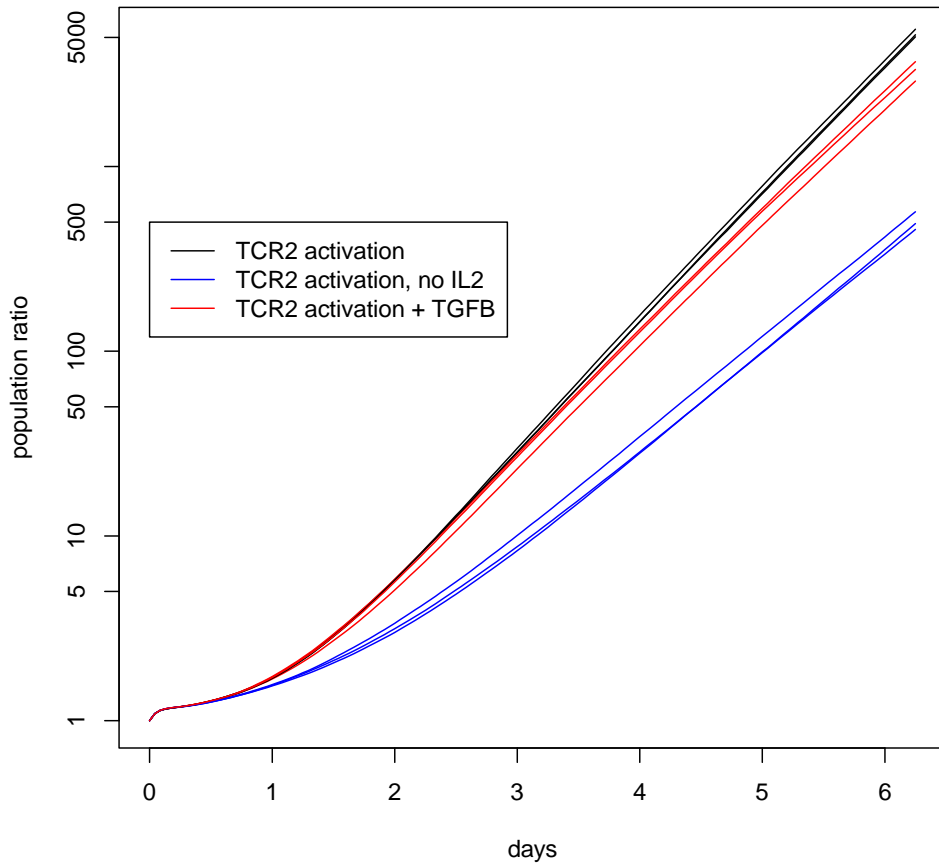
Model:



Three conditions of transient activation of TCR:

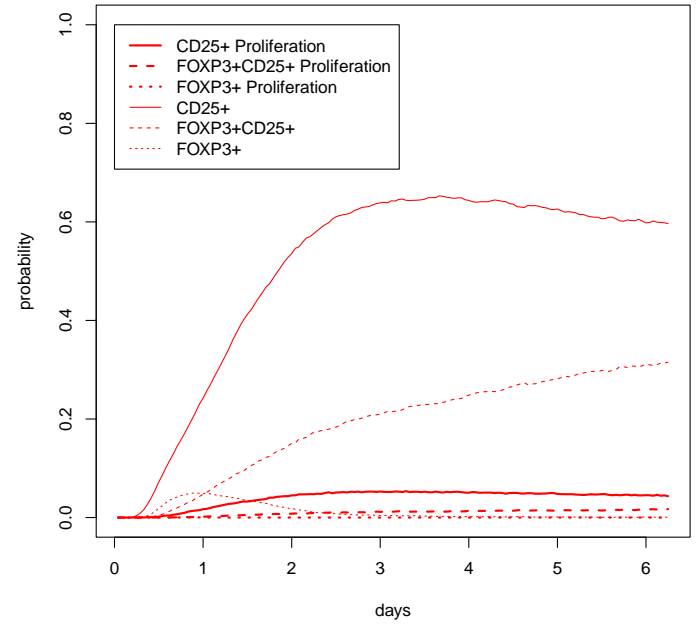
- TCR activation
- TCR activation with no IL2 loop
- TCR activation with TFGB (\rightarrow Treg)

Population ratio



Main difficulty: setting up the non-linear effect ligand \rightarrow receptor

Cell populations (TCR activation + TGFB)



Ligands

