**P53/MDM2 model**
**Example of MaBoSS use**

**https://maboss.curie.fr/**

We propose here an example of all the functions available in MaBoSS environment.

MaBoSS environment should be installed from https://maboss.curie.fr → Installation
GINsim should be installed from http://ginsim.org/dev/
The following files are necessary (they can be downloaded from https://maboss.curie.fr → How To):
- Model in GINsim format: example.zginml
- Configuration file for MBSS_DrugSim.py: Example_DrugSim.txt
- Profile file for MBSS_MultipleSim.py: ExampleProfile.txt
- Preparation file for MBSS_PrepareProjectFile.sh: Example.pmbss

We consider a published model of p53 response to DNA damage [1]. p53 interacts with Mdm2, which appears in two forms, cytoplasmic and nuclear. On one hand, p53 upregulates the level of cytoplasmic Mdm2 which is then transported into the nucleus and inhibits the export of nuclear Mdm2. On the other hand, Mdm2 facilitates the degradation of p53 through ubiquitination. In the model, stress regulates the level of DNA damage, which in turn participates in the degradation process of Mdm2. p53 inhibits the DNA damage signal by promoting DNA repair.

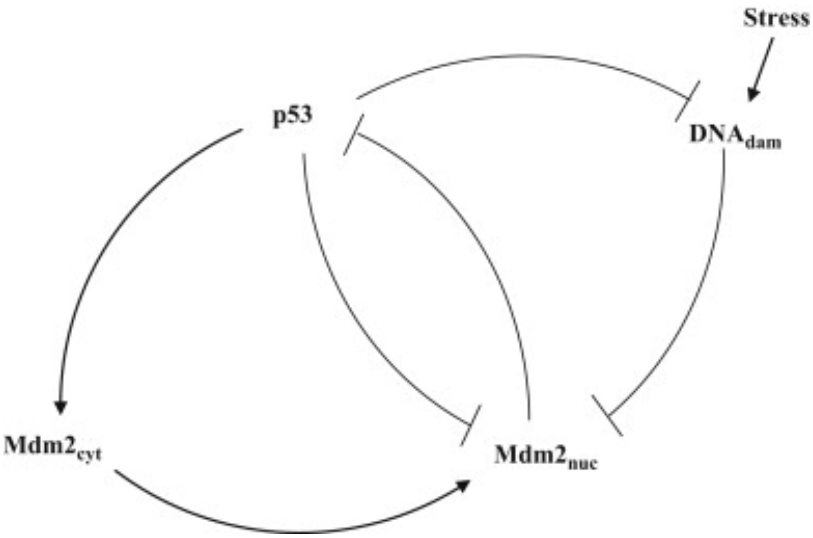
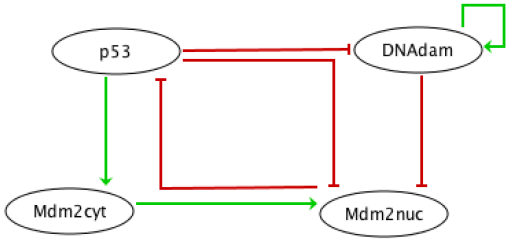


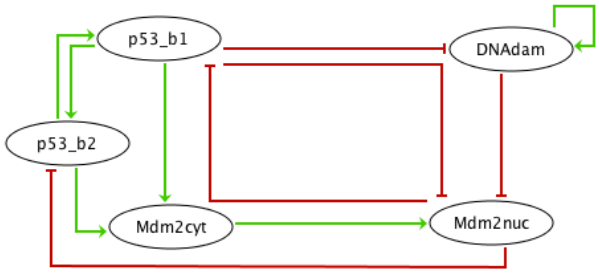
Figure 1. Model is presented in details in [1].




The model is multi-valued. Using GINsim, we booleanize the model.

```
$ java –jar GINsim–2.9.4–SNAPSHOT–jar–with–dependencies.jar ––dev
```

This version is in development and is the one that supports the “booleanization” of a multi-valued model.




The model is exported from GINsim in MaBoSS format, two files are created:
example.bnd
example.bnd.cfg (rename it as example.cfg)

The bnd file is as follows:

```
Node p53_b1 {
  logic = (!p53_b2 & !Mdm2nuc) | (p53_b2);
  rate_up = @logic ? $u_p53_b1 : 0;
  rate_down = @logic ? 0 : $d_p53_b1;
}
Node p53_b2 {
  logic = (p53_b1 & !Mdm2nuc);
  rate_up = @logic ? $u_p53_b2 : 0;
  rate_down = @logic ? 0 : $d_p53_b2;
}

Node Mdm2cyt {
  logic = (p53_b1 & p53_b2);
  rate_up = @logic ? $u_Mdm2cyt : 0;
  rate_down = @logic ? 0 : $d_Mdm2cyt;
}

Node Mdm2nuc {
  logic = (!p53_b1 & !Mdm2cyt & !DNAdam) | (!p53_b1 & Mdm2cyt) |
(p53_b1 & Mdm2cyt);
  rate_up = @logic ? $u_Mdm2nuc : 0;
```

```
  rate_down = @logic ? 0 : $d_Mdm2nuc;
}

Node DNAdam {
  logic = (!p53_b1 & DNAdam);
  rate_up = @logic ? $u_DNAdam : 0;
  rate_down = @logic ? 0 : $d_DNAdam;
}
```

And the corresponding cfg file : example.bnd.cfg, that you should rename : example.cfg

```
$u_p53_b1 = 1;
$d_p53_b1 = 1;
$u_p53_b2 = 1;
$d_p53_b2 = 1;
$u_Mdm2cyt = 1;
$d_Mdm2cyt = 1;
$u_Mdm2nuc = 1;
$d_Mdm2nuc = 1;
$u_DNAdam = 1;
$d_DNAdam = 1;

[p53_b1,p53_b2].istate = 1 [0,0], 0 [1,0], 0 [1,1];
[Mdm2cyt].istate = 1 [0] , 0 [1];
[Mdm2nuc].istate = 1 [0] , 0 [1];
[DNAdam].istate = 1 [0] , 0 [1];

time_tick = 0.5;
max_time = 1000;
sample_count = 10000;
discrete_time = 0;
use_physrandgen = 1;
seed_pseudorandom = 0;
display_traj = 0;
statdist_traj_count = 0;
statdist_cluster_threshold = 1;
thread_count = 1;
statdist_similarity_cache_max_size = 20000;
```

Note that because p53 is multi-valued, a special initial condition is written. There is a forbidden state: [p53_b1,p53_b2]=[0,1].

Open in an editor the file example.cfg.
You can put probability for initial conditions:

```
 [DNAdam].istate = .4 [0] , .6 [1];
```

For catching transient effects in the simulations of this example, you can modify the cfg file:

```
time_tick = 0.1
max_time=4
```

Note that the asymptotic solution has not been reached yet with max_time=4.

If you wish to set some variables as internal, you need to specify it and set the corresponding value to 1. Otherwise, they will be considered as external and will be part of the output:

```
Mdm2cyt.is_internal = 0;
Mdm2nuc.is_internal = 0;
DNAdam.is_internal = 1;
p53_b1.is_internal = 0;
p53_b2.is_internal = 0;
```

1. **Run MaBoSS simulations**

$ MBSS_FormatTable.pl example.bnd example.cfg

A folder is created with the name of the cfg file.

2. **Visualize the results**
    a. Fixed points

    Open example/example_fp.csv

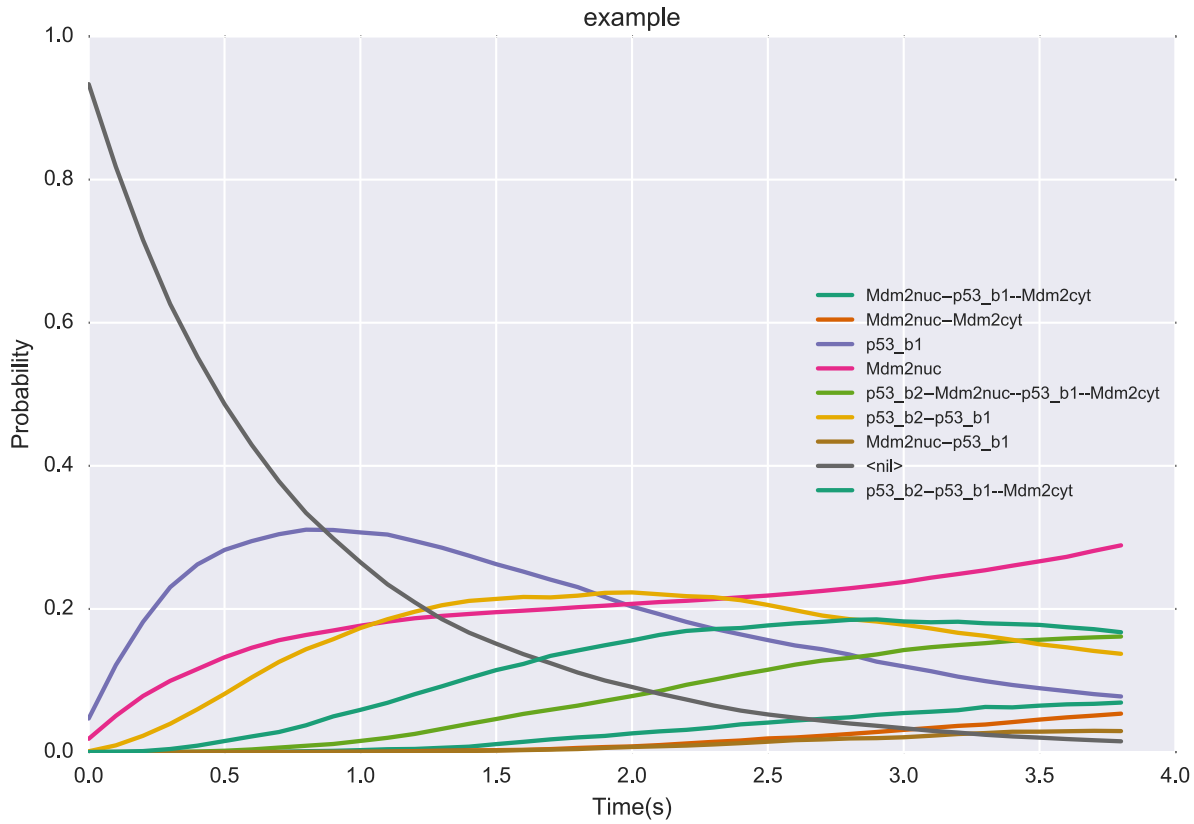
| FP | Proba | State | p53_b2 | Mdm2nuc | p53_b1 | Mdm2cyt | DNAdam |
|---|---|---|---|---|---|---|---|
| #1 | .2972 | Mdm2nuc | 0 | 1 | 0 | 0 | 0 |

The stable states can be compared to those of GINsim. In that case, the probability of the fixed point is not 1 because the simulation time has not reached the stable behavior yet.

    b. Trajectories

$ MBSS_TrajectoryFig.py example

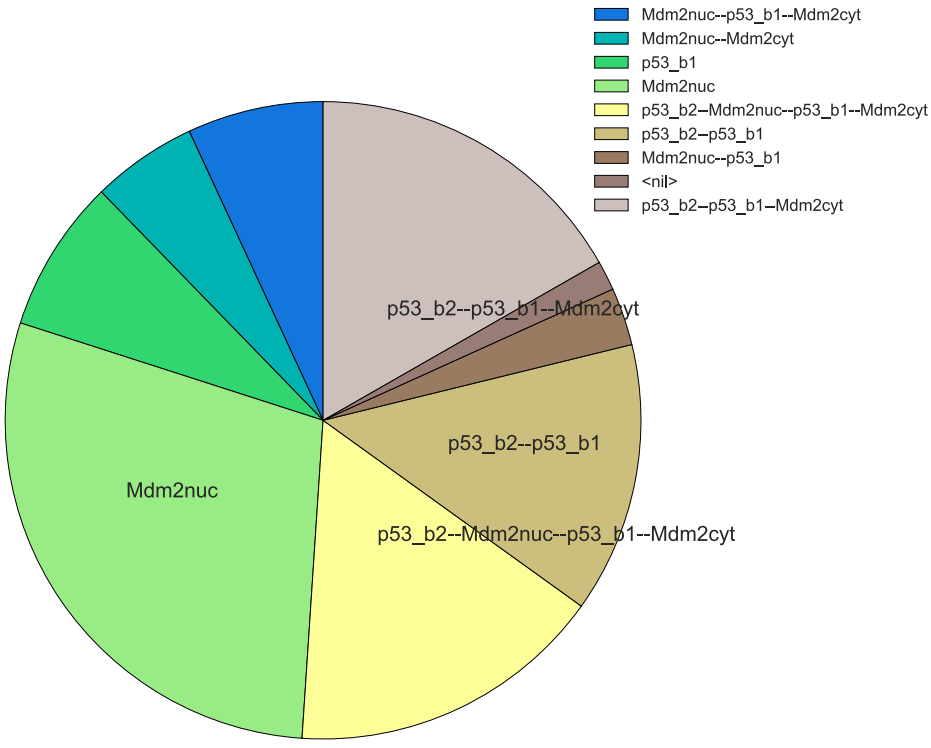
Open example_traj.pdf




c. Pie charts

$ MBSS_PieChart.py example

Open example_pie.pdf



### 3. Generate mutant network file

$ MBSS_MutBndCfg.pl example.bnd example.cfg 'Mdm2nuc Mdm2cyt p53_b1'

The nodes that were found in the bnd file are specified:

```
Catch node p53_b1
Catch node Mdm2cyt
Catch node Mdm2nuc
```

This is a verification that the nodes were properly written and modified accordingly in the newly created bnd file.

2 files are created :
example_mut.bnd
example_mut.cfg

In the cfg file, you can modify the parameters corresponding to the mutant simulations you wish to perform. Here Mdm2 nuclear is forced to be always active: open example_mut.cfg and set

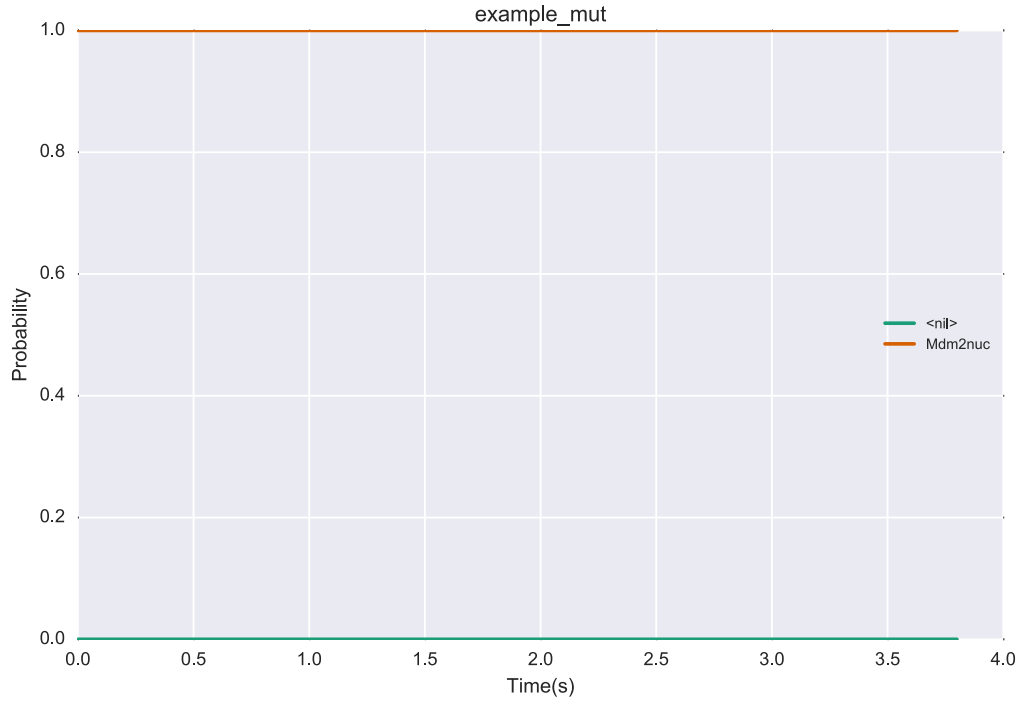
```
High_Mdm2nuc = 1;
```

Launch the simulation of the mutant and repeat previous steps:

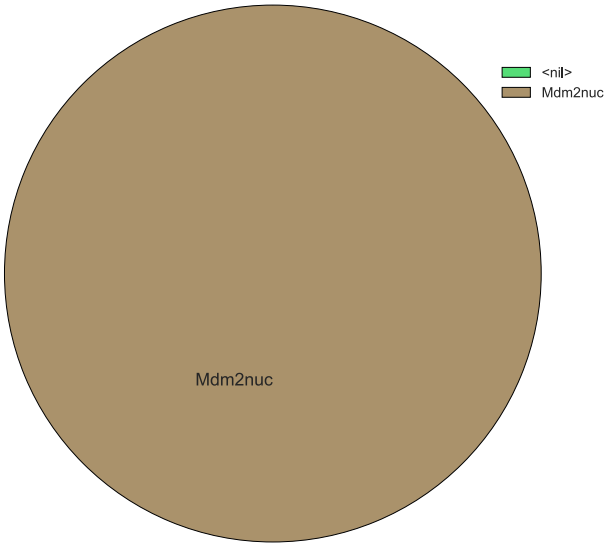
$ MBSS_FormatTable.pl example_mut.bnd example_mut.cfg

The results can be visualized using previous scripts:

$ MBSS_TrajectoryFig.py example_mut



$ MBSS_PieChart.py example_mut




4. **DrugSim**

DrugSim.py is helpful to investigate potential combination of two drugs having a null, intermediate or high inhibitory effect, by generating the corresponding bnd and cfg files for all the double-drug combination possibilities with all the inhibitory levels specified by the user in a .txt file. An example of the content of the Example_DrugSim.txt file can be found on the webpage and is as follows:

```
example
DNAdam,Mdm2cyt
0.0, 0.2, 0.4
6
```

Then type
$ MBSS_DrugSim.py Example_DrugSim.txt

First, the script will alter the bnd and cfg files to add the inhibitor nodes and their logical rules (adding “new” to the file names) and use these new files to create all the possible combinations of inhibition levels for two drugs (in a new folder “drug_simulations”).
Then, it will create bash files containing the commands to run the simulations and launch the simulations. Bash files are conserved in order to allow the user to (re)launch the simulations.

5. **Perform a parameter sensitivity analysis**

One variation is done at a time.
Type:
$ MBSS_SensitivityAnalysis.pl example.bnd example.cfg +10

All cfg files are prepared.
A folder is created. In it, a shell script allows to launch the simulations of all the possible cfg files. To run it, type:
$ cd Sensitivity_example/
$ ./Sensitivity_example.sh

A folder is created for all the cfg files. Each folder corresponds to a simulation and takes the name of the cfg file.

### 6. Use the chosen condition from a previous simulation as an initial condition

Modify the file example.cfg by setting all nodes as external:

```
Mdm2cyt.is_internal = 0;
Mdm2nuc.is_internal = 0;
DNAdam.is_internal = 0;
p53_b1.is_internal = 0;
p53_b2.is_internal = 0;
```

Re-run the simulation with MBSS_FormatTable.pl:
$ MBSS_FormatTable.pl example.bnd example.cfg

MBSS_FormatTable.pl creates a new folder: example_1

Create a new cfg file (example_Out2InitCond.cfg) without any initial condition; you can do it automatically with the following command:
$ cat example.cfg | perl -pe 's/.*istate.*//;' > example_Out2InitCond.cfg

Then type:
$ MBSS_InitCondFromTrajectory.pl example.bnd example_1/example_1_probtraj.csv 39 >> example_Out2InitCond.cfg

You have selected the bnd file, the trajectory file, the number of the line you wish to use as an initial condition and the config file in which you report the values of each node from the selected simulation as the initial condition.

In the cfg file (example_Out2InitCond.cfg), a new line was added:

```
[p53_b1,p53_b2,Mdm2cyt,Mdm2nuc,DNAdam].istate =0.000200 [0,0,0,0,0]
, 0.292177 [0,0,0,1,0] , 0.002908 [0,0,1,1,1] , 0.064079 [1,0,0,0,0]
, 0.122505 [1,1,0,0,0] , 0.010029 [1,1,1,0,1] , 0.032888 [1,0,0,1,0]
, 0.002688 [1,0,1,1,1] , 0.006566 [1,1,1,1,1] , 0.044666 [0,0,1,1,0]
, 0.153183 [1,1,1,0,0] , 0.074452 [1,0,1,1,0] , 0.146740 [1,1,1,1,0]
, 0.013477 [0,0,0,0,1] , 0.003046 [0,0,0,1,1] , 0.016049 [1,0,0,0,1]
, 0.012223 [1,1,0,0,1] , 0.002122 [1,0,0,1,1] ;
```

This cfg file can be used for a new MaBoSS run.

### 7. MultipleSim.py

MultipleSim.py prepares cfg files for performing several simulations.

Remove the initial conditions in the file example.cfg:

```
//[p53_b1,p53_b2].istate = 1 [0,0], 0 [1,0], 0 [1,1];
//[Mdm2cyt].istate = 1 [0] , 0 [1];
//[Mdm2nuc].istate = 1 [0] , 0 [1];
//[DNAdam].istate = 1 [0] , 0 [1];
```

We prepared an example of a txt file that performs two simulations:

```
TestProf1
p53_b2
DNAdam
p53_b1:u:30
Mdm2nuc:0.005
Mdm2nuc,Mdm2cyt
TestProf2
p53_b1,DNAdam

p53_b1:u:300
Mdm2nuc:0.005
Mdm2nuc,Mdm2cyt
```

A description of the format of the file can be found in the README_MultiSim file.

Type:
$ MBSS_MultipleSim.py example.cfg ExampleProfile.txt

Type "0" for using existing simulations, then type "1,2" for selecting both simulations (TestProf1 and TestProf2), then type "n" for not lauching the simulations on a cluster. MBSS_FormatTable.pl is launched twice, using example_TestProf1.cfg (where TestProf1 is the name of the first simulation in the ExampleProfile.txt file) and example_TestProf2.cfg (where TestProf2 is the name of the first simulation in the ExampleProfile.txt file) as configuration files.

### 8. Prepare a series of simulations

We created as an example a file Example.pmbss where we perform several tasks, all compacted into one single file.

```
# ---PrepareProjectFile File for preparing MaBoSS, to be executed
with the command "MBSS_PrepareProjectFile.sh File.pmbss"
# ---A directory "File" will be created
# ---Go to the directory: "cd File/", then run MaBoSS and construct
tables within the dedicated shell script: "source File.sh"
#----Avoid spaces in file names
#
#Executable MaBoSS name
MABOSS = MaBoSS;
#
#cfg file
CFG = example.cfg;
#
#BND file
BND = example.bnd;
```

```
#
#Construct an initial condition from a previous simulation, by
specifying the line number of a probtraj file to set as the initial
condition
INIT_COND=[example_mut.bnd,example_mut/example_mut_probtraj.csv,40];
#
# Build a cfg file for each single mutation of the specified nodes.
MUT=Mdm2nuc Mdm2cyt;
#
# Extend the set of cfg files by considering mutation combinations
up to the specified value
COMB_MUT=2;
#
# Build the cfg file for each parameter variation, specified by an
external variable and a suffix to add to the external variable
definition
VAR_SENS=[$u_Mdm2nuc , -.5] [$u_Mdm2cyt,* 0.01] [$d_Mdm2nuc,+ 0.03];
#
#Extend the set of cfg files by considering parameter variation
combinations up to the specified value
COMB_VAR_SENS=2;
#
#Compute table of probability trajectories? Yes, no or fig (for
producing pie and trajectory figures)
TRAJ_TABLE=fig;
#
#Compute table of stationary distribution decomposition? yes (with
probability cutoff) or no
#STAT_TABLE=[yes,0.004];
```

Type:
$ MBSS_PrepareProjectFile.sh Example.pmbss

Go to the newly created folder Example and run MaBoSS simulations:
$ cd Example
$ ./Example.sh

All cfg files are run with MaBoSS, the formatted trajectory tables are provided for each of the simulations, trajectory figures and pie charts are constructed.

**References**

[1] Abou-Jaoudé W, Ouattara DA, Kaufman M. From structure to dynamics: frequency tuning in the p53-Mdm2 network I. Logical approach. J Theor Biol. 2009 Jun 21;258(4):561-77.