

RULE-BASED MODELLING WITH BIONETGEN

Thank you for attending our workshop. This is our hand-out outlining the syntax you need to start creating BioNetGen models.

BioNetGen files can be generated and edited in any text editor software. The file then needs to be saved with an extension ‘.bngl’ Later the file may be opened with Rule Bender, BioNetGen command line or Virtual Cell.

BIONETGEN SYNTAX

The code in BioNetGen is usually split up in the following blocks:

1. `parameters` (rate constants, these define things such as rates of reactions)
2. `molecule types` (defines species – molecules involved your reactions)
3. `seed species` (starting concentrations of species)
4. `observables` (a list of things to be measured on the output graph)
5. `reaction rules` (write the reactions to take place here)

Each block is enclosed by ‘begin x’ and ‘end x’ e.g. ‘begin parameters’ and ‘end parameters’. As the model becomes more complex other blocks can be introduced, such as ‘functions and ‘compartments’. ‘Compartments’ would be inserted before molecule types and ‘functions’ after ‘observables’ and before the ‘reaction rules’.

Changing the order of blocks may be required as you progress to complex models but the parameters and parameter order above will be fine to get started.

To run the simulation the following commands most also be included after your list of blocks:

`generate_network({overwrite=>1})` – This step generates a network of reactions each time you run a simulation.

`simulate_ode({t_end=>1000,n_steps=>10000})` – produces a simulation output using Ordinary Differential Equations (ODE)

`simulate_ssa({t_end=>1000,n_steps=>10000})` – produces a simulation output using Stochastic Simulation Algorithm (SSA)

Example of the code is below. The # annotations describe what each line means.

FIGURE 1 ANNOTATED EXAMPLE OF BIONETGEN CODE

```

1 #BioNetGen syntax
2
3 begin parameters
4 k1 10 #rate constant and its value
5 k2 100
6 k_1 1
7 end parameters
8
9 begin molecule types
10 A(x) #In brackets is the binding site. Brackets are always necessary.
11 B(y) #X and Y are binding sites. Name isn't important
12 C() #Still needs brackets to indicate species
13 P(a,ps~U~P) #Molecule with multiple binding sites
14 end molecule types
15
16 begin seed species
17 A(x) 1e-4 #Starting concentrations
18 B(y) 1e-6
19 C() 0
20 end seed species
21
22 begin observables #Determines the output species
23 Molecules A A(x)
24 Molecules B B(y)
25 Molecules C C()
26 Molecules Complex A(x!1).B(y!1)
27 Molecules Unphosphorylated P(ps~U) #only unphosphorylated
28 Molecules Phosphorylated P(ps~P) #only phosphorylated
29 end observables
30
31 # the actual reactions
32 begin reaction rules
33 A(x)+B(y) <-> A(x!1).B(y!1) k1,k_1 #DOT means 'bound', using binding sites x and y
34 A(x!1).B(y!1) -> C() k2
35 C() -> 0 k2 #Degradation
36 0 -> C() k1 #Creation
37 end reaction rules
38
39
40
41 ## actions ##
42 generate_network({overwrite=>1})
43
44 # Equilibration
45 simulate_ode({t_end=>1000,n_steps=>1000})

```

EXERCISE 1

Modelling Michaelis-Menten kinetics.

The Michaelis-Menten reaction describes a simple reaction in which a substrate is converted to a single product after binding to an enzyme, as shown by the equation below:



Use RuleBender to model this as an ODE, using these parameters to start with:

$$k_{+1} = 1e6$$

$$k_{-1} = 1e-5$$

$$k_{+2} = 0.1$$

and these initial concentrations:

$$S = 30000$$

$$E = 12000$$

$$P = 0$$

Now convert your model into an SSA by converting concentrations to number of molecules using the following formula:

$$y = C \times N \times V$$

Where y is number of molecules, C is concentration, N is the Avogadro's constant (6.22×10^{23}) and V is cell volume (6.22×10^{23})

The rates of reactions would need to be converted from concentration/sec to molecules/ sec using the following formula:

$$\text{Rate new} = \text{rate old} \times N \times V$$

N is the Avogadro's constant (6.22×10^{23}) and V is cell volume (6.22×10^{23}) as above.

Rate constants in ODEs are usually denoted by a K , as above. You should replace this K with a C for your SSA model (to follow convention).

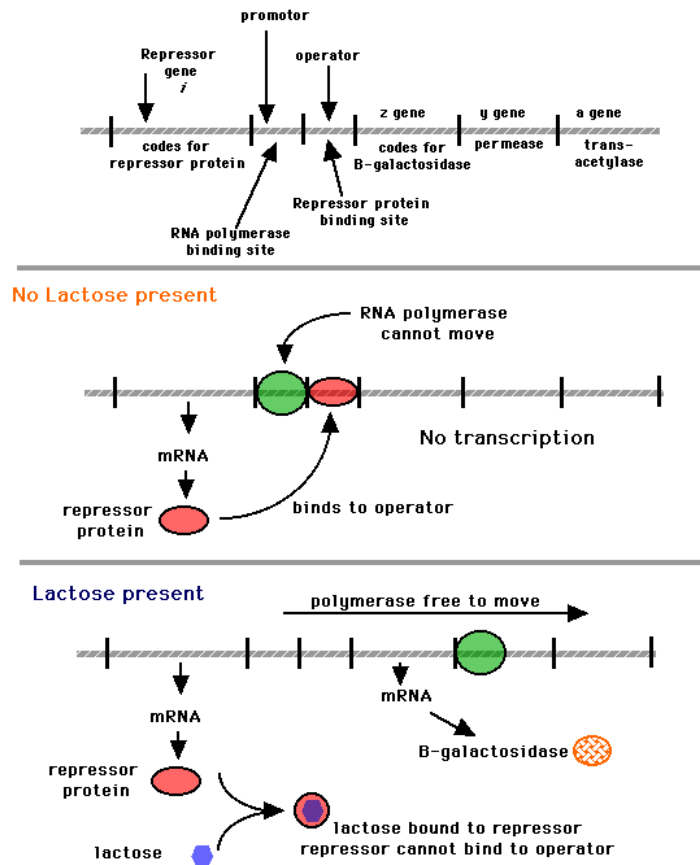
The ODE and SSA models you have produced so far should give you very similar graphs. Now try using a smaller number of molecules like 300 instead. What do you notice?

You can also try to change the parameters to see how they would influence the model.

EXERCISE 2

As an iGEM team we often wish to regulate the expression of the construct in some way. One common method of regulation uses an IPTG- inducible promoter which is switched ON only in the presence of IPTG. IPTG binds to and

The lac Operon



removes the repressor molecule from the promoter allowing the expression of the gene. Shown on Figure 2.

Design a model where protein expression is switched ON only in the presence of IPTG.

Some of the parameters include:

<i>VE. coli</i>	$8 \cdot 10^{-16}$	L	E. coli volume
OT	1	(copy number)	operator content ($[O]T \approx 2.08 \text{ nM}$)
<i>ksMR</i>	0.23	$\text{nM} \cdot \text{min}^{-1}$	lacI transcription rate [†]
<i>ksR</i>	15	min^{-1}	LacI monomer translation rate constant [†]
<i>kr</i>	960	$\text{nM}^{-1} \cdot \text{min}^{-1}$	association rate constant for repression [§]
<i>k-r</i>	2.4	min^{-1}	dissociation rate constant for repression [§]
<i>ksIMY</i>	0.5	min^{-1}	lacY transcription rate constant [†]
<i>ksOMY</i>	0.01	min^{-1}	leak lacY transcription rate constant [†]
<i>ksY</i>	30	min^{-1}	lacY translation rate constant [†]
<i>kp</i>	0.12	$\text{nM}^{-1} \cdot \text{min}^{-1}$	LacY-IPTGex association rate constant
<i>k-p</i>	0.1	min^{-1}	LacY-IPTGex dissociation rate constant
λMR	0.462	min^{-1}	lacI mRNA degradation constant [†]
λMY	0.462	min^{-1}	lacY mRNA degradation constant [†]
$\lambda R2$	0.2	min^{-1}	LacY degradation constant [†]
λY	0.2	min^{-1}	LacY-inducer degradation constant [†]
Comparison of Deterministic and Stochastic Models of the lac Operon Genetic Network Michail Stamatakis and Nikos V. Mantzaris Department of Chemical and Biomolecular Engineering, Rice University, Houston, Texas 77005			

We would like to thank you for coming to our workshop on rule-based modelling!

If you have any questions or want help creating a model **please do not hesitate to e-mail** us at y.dem'yanenko@ncl.ac.uk or g.a.pettitt@ncl.ac.uk . We would love to help you!

If you found this workshop useful, or it has inspired you to create your own models using BioNetGen, we would really appreciate it if you could let us know and site Team Newcastle on your wiki.

Have fun modelling!

Newcastle University iGEM Team

To find out more about our research visit our wiki at:

<http://2013.igem.org/Team:Newcastle>

To find out more about BioNetGen please visit:

<http://bionetgen.org/index.php/Documentation>

REFERENCES

<http://www.michaelsharris.com/12ubio/text/moleculargenetics/lacoperongeneregulation.htm> for the lac pic