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)
2. **molecule types** (defines species - molecules involved)
3. **seed species** (starting concentrations of species)
4. **observables** (output species are defined here)
5. **reaction rules**

Each block is enclosed by ‘begin x’ and ‘end x’. As the model becomes more complex other blocks can be introduced, such as ‘functions and ‘compartments’. These blocks need to be inserted at a specific place in the code. ‘Compartments’ will be before molecule types and ‘functions’ will be after ‘observables’ and before the ‘reaction rules’.

To run the simulation the following commands are used:

```bash
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 on the next page with annotations using a #.

The code shown in figure 1 is very simple and doesn’t show full power of BioNetGen. Our team has used it to model more complicated synthetic pathways such as subtilin two-component system and cell wall biosynthesis pathway (where murE is involved). These models are all single cell models, as at the moment it is not yet possible to use BioNetGen to model the changes in the population over time, and hence produce hybrid models. However we found that with the use of functions it is possible to mimic hybrid behaviour. To find out more about our research visit our wiki at:


To find out more about BioNetGen please visit:
# BioNetGen syntax

begin parameters
k1 10 # rate constant and its value
k2 100
k_1 1
end parameters

begin molecule types
A(x) # In brackets is the binding site. Brackets are always necessary.
B(y) # X and Y are binding sites. Name isn't important
C() # Still needs brackets to indicate species
P(a,ps-U-P) # Molecule with multiple binding sites
end molecule types

begin seed species
A(x) 1e-4 # Starting concentrations
B(y) 1e-6
C() 0
end seed species

begin observables # Determines the output species
Molecules A A(x)
Molecules B B(y)
Molecules C C()
Molecules Complex A(x!1).B(y!1)
Molecules Unphosphorylated P(ps-U) # only unphosphorylated
Molecules Phosphorylated P(ps-P) # only phosphorylated
end observables

# the actual reactions
begin reaction rules
A(x)+B(y) <-> A(x!1).B(y!1) k_1, k_1 # DOT means 'bound', using binding sites x and y
A(x!1).B(y!1) -> C() k_2
C() -> 0 k_2 # Degradation
0 -> C() k_1 # Creation
end reaction rules

## actions ##
generate_network(overwrite=>1)

# Equilibration
simulate_ode(t_end=>1000, n_steps=>1000)
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:

\[ S + E \xrightarrow[k_{+1}]{k_{-1}} ES \xrightarrow[k_{+2}]{k_{-2}} P + E \]

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

- \( k_{+1} = 1 \times 10^6 \)
- \( k_{-1} = 1 \times 10^{-4} \)
- \( k_{+2} = 0.1 \)
- Avogadro’s constant = \( 6.22 \times 10^{23} \)
- Cell volume = \( 1.2 \times 10^{-15} \text{ m}^3 \)

and these initial concentrations:

- \( S = 1 \times 10^{-4} \)
- \( E = 1 \times 10^{-6} \)
- \( P = 0 \)

Model as both an ODE and a SSA. For the SSA convert concentration to molecule number by using the formula:

\[ n = C \times V \times N \]

Where \( n \) is number of molecules, \( C \) is concentration, \( N \) is the Avogadro’s constant and \( V \) is cell volume.

The rates of reactions would need to be converted also:

\[ c = k \times V \times N \]

Where \( c \) is the rate constant for SSA, \( k \) is the rate constant for ODE, \( N \) is the Avogadro’s constant and \( V \) is cell volume.

This should give you very similar graphs. Now try using a smaller number of molecules like 300 instead.

You can also try to change the parameters to see how they would influence the model.
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 (isopropylthio-β-galactoside) induces the activity of β-galactosidase in bacteria. It binds to and removes the repressor molecule from the promoter allowing the expression of the gene. This results into a protein being made.

Design a model where protein expression is switched ON only in the presence of IPTG, and switched OFF when IPTG is absent.

### Table 1: Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V. E. coli$</td>
<td>$8 \times 10^{-16}$</td>
</tr>
<tr>
<td>$OT$</td>
<td>1 (copy number) operator content ($[O]T = 2.08 \text{ nM}$)</td>
</tr>
<tr>
<td>$k_{sMR}$</td>
<td>0.23 nM·min$^{-1}$ lacI transcription rate†</td>
</tr>
<tr>
<td>$k_sR$</td>
<td>15 min$^{-1}$ LacI monomer translation rate constant†</td>
</tr>
<tr>
<td>$k_r$</td>
<td>960 nM$^{-1}$·min$^{-1}$ association rate constant for repression§</td>
</tr>
<tr>
<td>$k_r^{-1}$</td>
<td>2.4 min$^{-1}$ dissociation rate constant for repression§</td>
</tr>
<tr>
<td>$k_{s1MY}$</td>
<td>0.5 min$^{-1}$ lacY transcription rate constant†</td>
</tr>
<tr>
<td>$k_{s0MY}$</td>
<td>0.01 min$^{-1}$ leak lacY transcription rate constant†</td>
</tr>
<tr>
<td>$k_sY$</td>
<td>30 min$^{-1}$ lacY translation rate constant†</td>
</tr>
<tr>
<td>$k_p$</td>
<td>0.12 nM$^{-1}$·min$^{-1}$ LacY-IPTGex association rate constant</td>
</tr>
<tr>
<td>$k_p^{-1}$</td>
<td>0.1 min$^{-1}$ LacY-IPTGex dissociation rate constant</td>
</tr>
<tr>
<td>$\lambda_{MR}$</td>
<td>0.462 min$^{-1}$ lacI mRNA degradation constant†</td>
</tr>
<tr>
<td>$\lambda_{MY}$</td>
<td>0.462 min$^{-1}$ lacY mRNA degradation constant†</td>
</tr>
<tr>
<td>$\lambda_R$</td>
<td>0.2 min$^{-1}$ LacY degradation constant†</td>
</tr>
<tr>
<td>$\lambda_Y$</td>
<td>0.2 min$^{-1}$ LacY-inducer degradation constant†</td>
</tr>
<tr>
<td>$k_{ar1}$</td>
<td>$3 \times 10^{-7}$ min$^{-1}$ association rate constant for IPTG-repressor binding</td>
</tr>
<tr>
<td>$k_{-ar}$</td>
<td>12 min$^{-1}$ dissociation rate constant for IPTG-repressor dissociation</td>
</tr>
</tbody>
</table>

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 cite and link to 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

For this task you may wish to use functions such as ‘logical if’. The syntax may be found in BioNetGen Bible at the link:

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