BEL2ABM manual and user guide

= and documentation of implemented behaviour =

V1.0

= Calambrone version =

Author: Michaela Gündel, Fraunhofer SCAI Bio, St. Augustin, Germany

michaela.guendel@scai-extern.fraunhofer.de

**Contents**

[This BEL2ABM manual / guide 3](#_Toc463947355)

[BEL2ABM 3](#_Toc463947356)

[Conversion of BEL to ABM behaviour – description 4](#_Toc463947357)

[BEL functions 4](#_Toc463947358)

[Abundance functions 4](#_Toc463947359)

[Modifications 5](#_Toc463947360)

[Activities 5](#_Toc463947361)

[Processes 6](#_Toc463947362)

[Transformations 6](#_Toc463947363)

[BEL relationships 7](#_Toc463947364)

[Causal relationships 7](#_Toc463947365)

[Correlative relationships 7](#_Toc463947366)

[Genomic relationships 7](#_Toc463947367)

[Other relationships 8](#_Toc463947368)

[Agent activity 8](#_Toc463947369)

[Agent location 8](#_Toc463947370)

[Homeostatic mimicking 9](#_Toc463947371)

[death with homeostatic value 9](#_Toc463947372)

[death with maximum upper limit 9](#_Toc463947373)

[reproduce 10](#_Toc463947374)

[Ontology leveraging 10](#_Toc463947375)

[Ontology format 11](#_Toc463947376)

[Biological behaviour 11](#_Toc463947377)

[Enzymes 11](#_Toc463947378)

[Allosteric enzymes 11](#_Toc463947379)

[External files 12](#_Toc463947380)

[Homeostatic value file 12](#_Toc463947381)

[Settings 12](#_Toc463947382)

[Arguments passed to the Java program 12](#_Toc463947383)

[Settings .ini file 12](#_Toc463947384)

[References 15](#_Toc463947385)

# This BEL2ABM manual / guide

This manual describes the currently implemented behaviours of the BEL2ABM tool. For every BEL language entity, we explain how it is converted into agent behaviour. The document details which of the BEL components have been implemented in the tool and which are parts of the future work list.

# BEL2ABM

BEL2ABM is a method and tool to automatically convert BEL [1] essentials to an Agent-Based Model (ABM). Please read our technical application note [] for details.

BEL2ABM uses an ontology (see Ontology leveraging), an external homeostatic value file (see Homeostatic mimicking and Homeostatic value file) and a settings file (see Settings .ini file) for conversion of BEL code to an ABM. The ABM is written in NetLogo [2] and run be run in the NetLogo suite in a “world” consisting of 51x51 patches. The user has various options of interacting with the simulation (Figure 1): via initial numbers, agent numbers for agent injection during runtime (however, agent numbers to be injected need to be changed directly in the code), binding options of complex agents and their binding strengths, movement speeds, reaction distance, homeostatic mimicking and lifespans.



Figure 1: Simulation window in NetLogo. (1) initial numbers (2) binding options of complex agents (3) binding strengths of complex agents (4) movement speeds (5) reaction distance (6) homeostatic mimicking switch (7) lifespans

The output of the conversion is a NetLogo (.netlogo) file together with a locations.txt file that contains all (x,y) positions of all agents at all time points together with their breed name and breed ID.

# Conversion of BEL to ABM behaviour – description

## BEL functions

### Abundance functions

In general, any abundance found in the BEL code is translated to be an agent in the ABM code. Its abilities and behaviours are inferred both from the BEL code and from the ontology.

#### abundance(), a()

A non-specified abundance is converted into a default agent via the -defaultAbundance setting (see Settings .ini file).

#### complexAbundance(), complex()

This is treated as a molecular complex and can currently have 2 members (for instance, a complex of a beta secretase bound with an APP). Whenever agents of these 2 kinds meet, they can, depending on their activity value probability, form a complex agent (the two single agents will die with their energies preserved, and a new complex agent will be born). Depending on the binding strength set by the user via a slider, this complex will remain bound or will dissolve, resulting in two new single agents with old energies minus energy used during the lifetime of the complex, with the complex dying.

#### compositeAbundance(), composite()

Two agents can work together to exert a certain effect. This is represented in BEL via composite abundances. In BEL2ABM, whenever one of each agent types get inside the reaction distance set by the user, they form a composite agent (with the two single agents dying, but their energy lists are preserved) that can then exert its effect in the simulation. In contract to complex agents, at every time step a check is done whether its members moved away from each other (they are not retained via binding strength) too far and out of reaction distance. If they moved too far, the composite agent dies and 2 single agents will be created according to their old energies minus energy used as a composite agent.

#### geneAbundance(), g()

Currently converted into a default agent if no direct link to the ontology is available in the BEL code.

#### microRNAAbundance(), m()

Currently converted into a default agent if no direct link to the ontology is available in the BEL code.

#### proteinAbundance(), p()

All protein abundances are connected to the protein ontology class set by the user (see Settings .ini file) and will thus have all protein qualities set in the ontology.

#### rnaAbundance(), r()

Currently converted into a default agent if no direct link to the ontology is available in the BEL code.

### Modifications

#### proteinModification(), pmod()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### substitution(), sub()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### truncation(), trunc()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### fusion(), fus()

Currently not implemented. Corresponding BEL code is currently disregarded.

### Activities

#### catalyticActivity(), cat()

A catalytic activity will be treated as a molecular activity of an enzyme. The normal “Agent activity” of the agent is used.

#### chaperoneActivity(), chap()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### gtpBoundActivity(), gtp()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### kinaseActivity(), kin()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### molecularActivity(), act()

This activity corresponds to the activity value assigned to an agent (see Agent activity section). If something in the BEL code increases or decreases the molecular activity of an entity X, the corresponding agent X’s activity will be increased or decreases accordingly in the simulation.

#### peptidaseActivity(), pep()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### phosphataseActivity(), phos()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### ribosylationActivity(), ribo()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### transcriptionalActivity(), tscript()

Currently not implemented. Corresponding BEL code is currently disregarded.

#### transportActivity(), tport()

tport() only has a partial implementation thus far, for composite and complex agents and increases only.

In cases of tport (ag) -> complex(ag1, ag2), the agent ag will grab ag1 and ag2 agents if they are within the reaction distance and the same region and get them closer to each other and the calling complex / composite agent (with 1.5 times their normal movement speed).

### Processes

#### biologicalProcess(), bp()

A biological process is converted into a NetLogo procedure. A procedure can be called during simulation runtime either by an agent or by another process. Such process procedures can have the effect to increase or decrease another agent (let it die under certain probability if within reaction distance and inside the same region, or create a new one under certain probability) or another process. A process procedure is only executed inside the region in which it is valid.

#### pathology(), path()

Pathologies are currently treated the same as processes.

### Transformations

#### translocation(), tloc()

A translocation moves an agent from one region to a different region under a certain probability. A translocation can have a further effect (tloc(…) -> or -| a() , bp() or act(a())).

#### cellSecretion(), sec()

Currently not implemented. Relative code will be ignored.

#### cellSurfaceExpression(), surf()

Currently not implemented. Relative code will be ignored.

#### degradation(), deg()

Currently not implemented. Relative code will be ignored.

#### reaction(), rxn()

Reactions have an input and an output. As a general rule, a check is done whether all input reactants are within reaction distance and inside the same region. According to a certain probability corresponding to the calling agent’s activity value, the reaction is carried out: the input reactants die and the output products are newly created. Enzymes are a special case of reactants and are retained after the reaction if specified in the product output list. In case of allosteric enzymes, one bound molecule gets split off and the enzyme’s activity value is reduced accordingly (compare Allosteric enzymes section).

A reaction can also decrease or increase an agent, a bioprocess (no implementation yet for decreases bp) or an agent’s activity (reaction(reactants(), products()) -> or -| …) after successful execution of the reaction.

## BEL relationships

### Causal relationships

#### decreases, -|

For occurrences of a(A1) -| a(A2) in the BEL code, at every time step, agent A1 is asked to check whether A2 is within reaction distance inside the same region. If so, A2 will die according to A1’s activity probability value. The same holds if a process calls -| a(A2), only that a random probability [0..100] is applied. Occurrences of … -| bp() are currently ignored.

#### directlyDecreases, =|

Same implementation as decreases.

#### increases, ->

For occurrences of a(A1) -> a(A2), the activity probability of A1 value is applied to determine whether a new A2 will be created (random [0.100] probability in the case of processes).

#### directlyIncreases, =>

Same implementation as increases.

#### causesNoChange

This relationship is not translated into ABM code.

### Correlative relationships

#### negativeCorrelation

Not implemented yet.

#### positiveCorrelation

Not implemented yet.

#### association, --

No translation to ABM code.

### Genomic relationships

#### Analogous

Not implemented yet.

#### Orthologous

Not implemented yet.

#### transcribedTo, :>

Not implemented yet.

#### translatedTo, >>

Not implemented yet.

### Other relationships

#### biomarkerFor

If an abundance is a biomarker for a process, the corresponding agent will under a certain probability random [0..100] increase the process.

#### hasMember

Complex agents (complexAbundance(), complex()) carry a list of members with them. A new member is created for every occurrence of hasMember in the BEL code.

#### hasMembers

The member list of complex agents will be increased with the new members coming from this triplet.

#### hasComponent

Composite agents (compositeAbundance(), composite()) have a list of components attached to them. A new component is added for every occurrence of hasComponent in the BEL code.

#### hasComponents

The component list of composite agents will be increased with the new components coming from this triplet.

#### isA

isA statements found in the BEL code (both for abundances and processes) are used to establish the connection to the ontology.

#### prognosticBiomarkerFor

Not implemented yet.

#### rateLimitingStepOf

Not implemented yet.

#### subProcessOf

Not implemented yet.

# Agent activity

An agent that is active (see –activeProperty in Settings .ini file section) will have an activity value (random 100) and will participate in the simulation according to this probability. The activity value can change during the simulation depending on the BEL code (eg., bp(…) -> act(someEntity).

# Agent location

Agents that have -locatedIn and/or -producedIn axioms inside the ontology or spatial annotations in the BEL code (eg. via Anatomy) (cf. Settings .ini file section) will only be allowed to move in space within these regions in the simulation. At setup, agents with -producedIn will be located only in these regions. In order to cross regional boundaries, a tloc() triplet is necessary in the BEL code to make them move to the corresponding regions.

# Homeostatic mimicking

The following is valid only for agents that follow homeostasis. This is determined according to whether the agent’s ontology class has a -isBodilyDevelopmentalProcess link, see Settings .ini file.

If homeostasis\_mimicking is switched on in the simulation, agent reproduction and death will be more or less likely the farer away the current entity count is from this homeostatic value. Based on the ontology hierarchy, missing values are inferred in unambiguous cases or maximum upper limits are used cases in which this is not possible.

Probabilities will be calculated as follows:

## death with homeostatic value

;;homeostasis mimicking: die when there are too many of your kind

let current count breed

let h homeostatic-[value\_of\_agent]

let minimum h / 3

let maximum h \* 3

let dev\_cur\_from\_homeo current – h ;; deviation of current number from homeostatic value

let ran random-normal-in-bounds h (h / 20) minimum maximum

let dev\_ran\_from\_homeo abs h – ran ;; deviation of random normal number from homeostatic value

if dev\_cur\_from\_homeo > 0 and ( random-float 1 >= abs ( dev\_ran\_from\_homeo / dev\_cur\_from\_homeo) )

;; the greater the deviation, the higher the probability to die

 [

 if random 100 < 50

 [ die ]

 ]

## reproduce

let current count breed

let minimum maxhomeostatic-Teff\_naive / 3

let maximum maxhomeostatic-Teff\_naive \* 3

let dev\_cur\_from\_homeo current - maxhomeostatic-Teff\_naive ;; deviation of current number from homeostatic value

let ran random-normal-in-bounds maxhomeostatic-Teff\_naive (maxhomeostatic-Teff\_naive / 20) minimum maximum

let dev\_ran\_from\_homeo abs maxhomeostatic-Teff\_naive - ran ;; deviation of random normal number from homeostatic value

if dev\_cur\_from\_homeo > 0 and ( random-float 1 <= abs ( dev\_ran\_from\_homeo / dev\_cur\_from\_homeo ) ) ;; the greater the deviation from maxvalue, the lower the probability to reproduce

 [

 if random 100 < dupli-rate-[…] and energy > 0 [

 set activity (activity / 2) ;; divide activity between parent and offspring

 hatch-[agent] times [ lt random 90 set energy random (2 \* lifespan-[…]) set color […] set size […] ] ;; don't move forward to prevent leaving the region

 ]

## Reproduce with upper limit

;; agent has an upper limit of

;; if its number gets as high or higher than this, let its youngest agents die

let cur\_no count breed

let youngest one-of breed ;; just to initialize

repeat cur\_no - upper-lim-myelin - 1

[

 set youngest max-one-of breed [energy]

 ask youngest [ die ]

]

# Ontology leveraging

All agents have either a direct (via namespace or isA triplet in the BEL code) or asserted (via defaults) connection to the ontology and will be treated as such in the simulation. Proteins will be treated as proteins and will thus have a lifespan, but for instance cannot reproduce), genes will be genes, cells will be cells (and thus have a lifespan AND can reproduce), processes will be processes etc. For all possible links to the ontology and behaviours/characteristics usable for the simulation please consult the Settings .ini file section.

## Ontology format

The ontology needs to be in RDF/XML format. BEL2ABM does not perform any reasoning on the ontology, so make sure that you use an inferred version of your ontology (1 single file) if you need reasoning.

## Biological behaviour

Both agents and procedures get part of their behaviour from a) the BEL code and b) the ontology. BEL2ABM uses the hierarchical structure of the ontology (all assertions made for a class are also valid for all subclasses) and the axioms attached to the classes via the relationships listed in the Settings .ini file section. Thus, whenever a general upper class agent performs certain behaviour in the simulations, all its subclass agents (if contained in the BEL code) will show the same behaviour. The same holds for processes, whenever a general upper class process is called, its subclass processes (if contained in the BEL code) show this same behaviour.

Agents that are linked either directly to a subclass of -enzyme or -allostericEnzyme are treated as such inside reactions.

### Enzymes

(currently no particular implementation)

### Allosteric enzymes

Allosteric enzymes can have more than 1 binding site. The user can set the number of molecules that can bind to the allosteric enzyme via a chooser in the simulation window (1:n or n:1, depending on the molecular complex name) and can freely set the number of molecules that can bind to the enzyme (Figure 2). Whenever a new molecule binds to the allosteric enzyme, the enzyme’s activity will rise according to

set activity activity + ( 50 / ( APP.APP.Beta\_secretase.Beta\_secretase\_maxn - 1) ,

and whenever it loses one bound molecule, its activity will decrease according to

set activity activity - ( 50 / ( APP.APP.Beta\_secretase.Beta\_secretase\_maxn - 1) .

This way, the more molecules are bound to the enzyme, the higher the possibility that the enzyme complex will participate in a reaction.



Figure 2: Chooser for number of binding sites. The setting shown says that n APP.APP (dimer) molecules can bind to 1 beta\_secretase.beta\_secretase (dimer) allosteric enzyme. The lower part specifies n (“APP.APP.Beta\_secretase.Beta\_secretase\_maxn ”) to 2 (2 binding sites for APP dimers).

# External files

## Homeostatic value file

The homeostatic value file is read during runtime. It needs to be set using the .ini file, see Settings .ini file. If no homeostatic value is available, a maximum value may be set that must not be exceeded during the simulation.

The external homeostatic value file needs to follow the following format (tab separated):

OntoID→label→homeo\_value→max\_level→comment→unit→source

Example:

http://scai.fraunhofer.de/MSOntology#T\_Reg→Regulatory T cells→20→→rare→microliter→"Cellular and Molecular Immunology, 8th edition, Abbas, Lichtman and Pillai."

# Settings

## NetLogo sliders, choosers etc.

### duplicate-rate-…

The probability in percent with which the agent will reproduce.

### ini-no-…

Initial number of agent at setup.

### upper-lim-…

This takes effect only on the agent’s reproduce procedure. The agent will stop to reproduce once the upper limit threshold has been reached.

### …-move-speed

The speed with which the agent moves in the world. If set to 0, the agent is immobile.

### [member number choosers, maxn]

For complex and composite agents, the user can choose the number of agents than can maximally interact with each other. If the agent’s name is agent1.agent2, then the meaning is the following:

n:1 n agent1’s can interact with 1 agent2 (ie, agent2 has n binding sites for agent1)

1:n n agent2’s can interact with 1 agent1 (ie, agent1 has n binding site for agent2)

1:1 agent1 and agent2 can only interact 1 with 1

The n can be set in the chooser directly below.

### bind-str-…

The binding strength of complex agents. It corresponds to the agent’s probability to remain bound or dissolve into 2 separate agents.

### reaction-distance

Distance used to evaluate agents’ distance for any kind of reaction.

### homeostasis-mimicking

See Homeostatic mimicking.

### lifespan-…

This corresponds to the energy value (lifetime) of an agent in terms of ticks. Every agent at setup or agent creation time gets an energy value of random 2 \* lifespan-…. Thus, at most after 2 \* lifespan-…, the agent will die of age.

## Arguments passed to the Java program

| ***Argument*** | ***Description*** | ***Example value*** |
| --- | --- | --- |
| -l  | Lists the KAMs in the KAM store. OpenBEL method. |  |
| -k | The KAM to be used | APP\_SORLA |
| -ABMCode | The output file to be created | output.nlogo |
| -v | Verbous output in resulting .netlogo file (includes provenance of the code) |  |

## Settings .ini file

| ***Argument*** | ***Description*** | ***Example values*** |
| --- | --- | --- |
| -agent | The BEL abundances that will be used for display in the NetLogo simulation. Note: This is just for display. Internally, all abundances are transformed into agents. Use long names of BEL terms (eg. complexAbundance() instead of complex()) | complexAbundance(proteinAbundance(MSO:"Alpha secretase"),proteinAbundance(MSO:"Alpha secretase"))proteinAbundance("sappalpha\_d") |
| -BELTermAnnotationProperty | The annotation property used in the ontology to connect it to BEL. | http://scai.fraunhofer.de/HuPSON#BELtermontology triple “allosteric enzyme” example:http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000340 http://scai.fraunhofer.de/HuPSON#BELtermabundance(HUPSON:"allosteric enzyme") |
| -onto | The ontology to be used. | C:/Users/ontologies/HuPSON\_inferred.owl |
| -agentrelation | A check is done whether the abundance can be used as an agent. If the ontology class has an axiom attached to it (via -agentrelation) that points to the –agentclass, it means the abundance can. | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00001036 eg[http://some\_class] http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00001036 http://scai.fraunhofer.de/HuPSON#agent  |
| -agentclass | A check is done whether the abundance can be used as an agent. If the ontology class has an axiom attached to it (via -agentrelation) that points to the –agentclass, it means the abundance can. | http://scai.fraunhofer.de/HuPSON#agent eg[http://some\_class] http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00001036 http://scai.fraunhofer.de/HuPSON#agent  |
| -defaultAbundance | If the abundance is not connected to the ontology, this default is assumed. | http://www.ifomis.org/bfo/1.1/snap#MaterialEntity |
| -defaultProcess | If the process is not connected to the ontology, this default is assumed. | http://www.ifomis.org/bfo/1.1/span#Process |
| -complexAbundance | If the abundance is not connected to the ontology, this default is assumed. | http://purl.obolibrary.org/obo/CHEBI\_36080 |
| -compositeAbundance | If the abundance is not connected to the ontology, this default is assumed. | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00001152 |
| -proteinAbundance | If the abundance is not connected to the ontology, this default is assumed. | http://purl.obolibrary.org/obo/CHEBI\_36080 |
| -enzyme | All abundances that have this superclass are considered as enzymes and treated as such. | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00001449 |
| -allostericEnzyme | All abundances that have this superclass are considered as allosteric enzymes and treated as such. | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000340 |
| -locatedInAnnotationName | Sets the terminology used in the BEL code to specify the location of an abundance. | AnatomyNervousSystem |
| -locatedIn | Checks the ontology for this URI to establish where an abundance may be located. | http://purl.org/obo/owl/ro#located\_in |
| -producedIn | Checks the ontology for this URI to establish where an abundance is produced. | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000302 |
| -qualProp | URI in the ontology that points to qualitative properties.  | http://purl.obofoundry.org/obo/OBI\_0000298 has\_qualityeg: protein has\_quality some life\_span |
| -mathmlProp | URI used as annotation property in the ontology to connect a class to its MathML code. | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_71497513 eg hasContentMathML <”math… /> |
| -agentreproducealgorithm | Used for agent introduction. Variable values in order of appearance inside MathML string, tab separated | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000015 20 365here: stochastic pulse trains |
| -agentreproducealgorithm\_default | If no –agentreproducealgorithm is specifically set, agents are introduced randomly into the system | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000032eg random agent reproduce |
| -homeostatic\_concentrations | A tab separated external file that specifies homeostatic values of entities. See Homeostatic mimicking section. | C:\Users\latitude\_user\workspace\BEL2ABM\homeostatic\_values\_peripheralblood.txt |
| -homeostatic\_concentrations\_default | If homeostatic mimicking is switched on, this is the default value for all entities whose homeostatic concentration is not contained in the external file. | 1000 |
| -isBodilyDevelopmentalProcess | refers to the axiom attached to a class whose agent will be periodically introduced into the model because it is the output of some bodily development function that steadily occurs over time in the organism | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000039 http://purl.org/obo/owl/GO#GO\_0032502here: is\_output\_of some hematopoiesis |
| -increases | The relation in the ontology used to connect a class to another class that it increases the number/occurrence of. | http://scai.fraunhofer.de/HuPSON#increases |
| -increasedby | The inverse relation of –increases. | http://scai.fraunhofer.de/HuPSON#increased\_by |
| -decreases | The relation in the ontology used to connect a class to another class that it decreases the number/occurrence of. | http://scai.fraunhofer.de/HuPSON#decreases |
| -decreasedby | The inverse relation of –decreases. | http://scai.fraunhofer.de/HuPSON#decreased\_by |
| -processURI | process class inside the ontology, for look-up; to connect processes disconnected to the ontology. | http://www.ifomis.org/bfo/1.1/span#Process |
| -reproduce | An agent that can reproduce will have a link to this ontology class. | http://purl.org/obo/owl/PATO#PATO\_0001434 |
| -inactiveProperty | An agent that is inactive will have a link to this ontology class. The agent will have no activity value in the ABM and will thus participate in the simulation without any dependency on activity. | http://purl.org/obo/owl/PATO#PATO\_0001706 |
| -activeProperty | An agent that is active will have a link to this ontology class. The agent will have an activity value (random 100) and will participate in the simulation according to this probability. See Agent activity section. | http://purl.org/obo/owl/PATO#PATO\_0001707 |
| -noHomeostasis | indicates that an agent isn't controlled by homeostasis: in HuPSON ''number controlled by homeostasis' some false' | http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000157 http://scai.fraunhofer.de/HuPSON#SCAIVPH\_00000086 |
| -reactionDistance | An agent can interact with other agents that are within a distance of [0..-reactionDistance]. | 3 |

# To do list

* (go through the code and check TODO entries)
* Dependency of a reaction and all kinds of relationships needs to become also dependent on the concentration of agents in the reaction distance. So far, a random choice is made whenever an agent can participate in more than 1 reaction/relationship (eg, whenever it can participate in n actions, the action to be performed is chosen by “random n”). Only the one chosen is then performed, without looking at concentrations.

# References

|  |  |
| --- | --- |
| [1]  | “OpenBEL: Biological Expression Language (BEL),” [Online]. Available: http://www.openbel.org/bel-expression-language.. [Accessed 25 09 2015]. |
| [2]  | U. Wilensky, “NetLogo.,” Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL., 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo. [Accessed 01 10 2015]. |