



Grupo de Dinâmica Não-Linear

IST, Dep. de Física

Av. Rovisco Pais

1049-001 Lisboa

FAX: (351)-218419123; URL: <http://sd.ist.utl.pt>

GeneticNetworks

A Mathematica© Package for the modeling and simulation of
Genetic Regulatory Networks

2009

by

Rui Dilão and Daniele Muraro

In the framework of the GENNETEC project
(FP6 STREP IST 034952)

<https://sd.ist.utl.pt/>

NonLinear Dynamics Group, Instituto Superior Técnico

Lisboa, 31 July 2009

GeneticNetworks is a package that simulates the dynamics of the concentrations of genes and proteins of a genetic regulatory network. The starting point is the double-graph of activations and repressions. The input are two strings, *activations* and *repressions*, that record the network of interactions. The same symbol is used to denote both a gene and the protein that it produces. For example,

$$\begin{aligned} \text{activations} &= \{A \rightarrow B, A \rightarrow R\} \\ \text{repressions} &= \{R \rightarrow B\} \end{aligned} \tag{1}$$

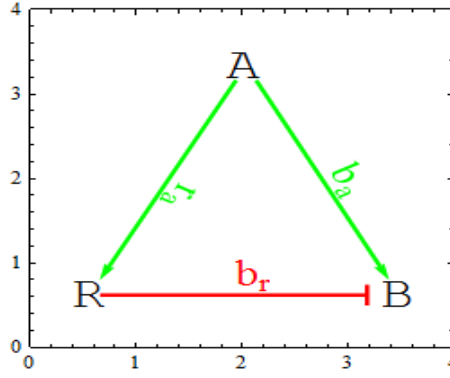


Figure 1: Graph of interactions for the network (1).

In this double-graph, protein A activates the genes B and R, and protein R represses gene B (see figure 1). The proteins whose corresponding gene is lacking, can be considered as constant proteins, adding the symbol \$ to their name (e.g. in order to consider protein A as a constant protein during time, it is necessary to write A\$).

The following functions are defined in the *GeneticNetworks* package:

- NetworkGraph, ManipulateGraph
- Reactions, ReactionsOneSite, ReactionGraph
- SubstanceNames, SubstanceVariables, SubstanceInitialConditions
- ParameterNames, ParameterInput
- Equations
- ConservationLaws
- EmptyGenes, EmptyGenesPosition, NotEmptyGenes

- WriteDataC, CEquations

NetworkGraph[activations, repressions]

returns the graph of the activations (green arrows) and repressions (red edges). The structure of the first graph is chosen by default by the Mathematica function *GraphPlot* and can not be changed.

ManipulateGraph[activations, repressions, dimension of the graph]

returns the graph of the activations and repressions. The graph can be manipulated to the desired form.

In Fig. 2, we represent the graph of the activations in the gap-genes network of *Drosophila* using both functions.

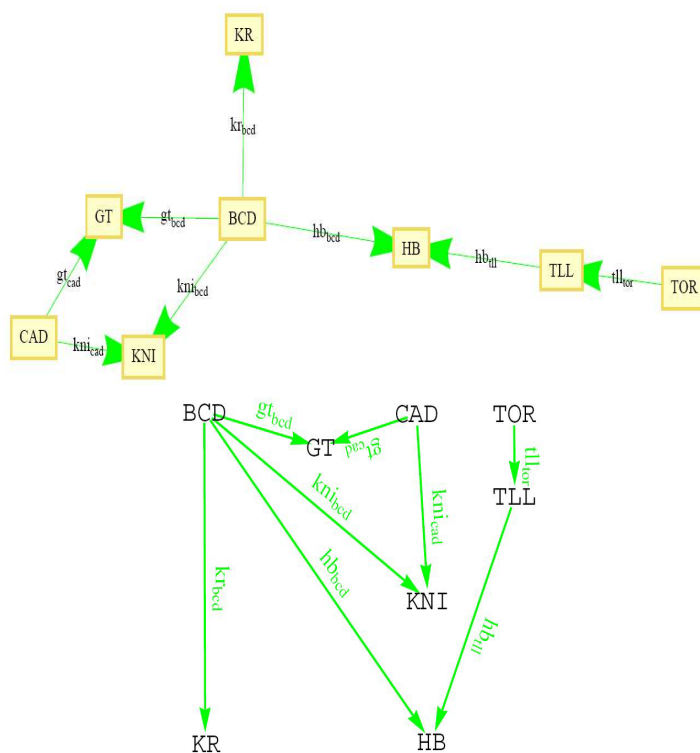


Figure 2: Graph of the activations in *Drosophila* gap-genes network. On the top we see the output of *NetworkGraph*, on the bottom we have the same graph obtained by *ManipulateGraph*.

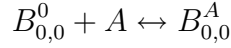
Reactions[activations, repressions]

prints the names of the proteins of the graph; the number of activations, repressions and total sites of the corresponding gene; the name of the proteins which

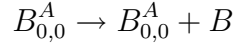
have no binding sites; and the proteins that have been chosen to be considered constant. This function also returns two tables, *reactions* and *parameters*, containing all the reactions and the parameters of the network. Genes are denoted by the name of the protein they produce and with subscripts and overscripts indicating their binding sites. The subscripts are reserved to proteins which repress the gene, while the overscripts to proteins which activate it. The interactions can be divided into three different species, each stored in a global table:

ReactionGeneProtein, *ReactionProduction* and *ReactionDegradation*.

The first table holds the double bimolecular reactions of the binding, e.g.



The second table holds the productions of proteins, e.g.



The third table holds the degradations of proteins (if any), e.g.



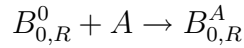
Correspondingly, the following global tables for the parameters are defined:

ParameterFw, *ParameterBw*, *ParameterProduction*, *ParameterDegradation*.

ParameterFw and *ParameterBw* hold the parameters related to *ReactionGeneProtein*. The parameters are denoted by the small letter of the gene involved in the reaction and, in the indexes, the binding sites of the gene and the small letter of the protein reacting with the gene. For the backward reaction the index is preceded by a minus sign.

For example,

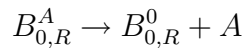
for the forward reaction



the corresponding parameter is

$$(b_{0,R}^0)_a,$$

for the backward reaction

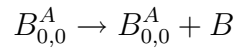


the corresponding parameter is

$$(b_{0,R}^0)_{-a}.$$

ParameterProduction holds the parameters related to *ReactionProduction* denoted by the letter p and, in the indexes, the binding sites of the gene, and the name of the protein involved in the reaction. For example,

for the reaction



the corresponding parameter is

$$(p_{0,0}^A)_B$$

.

ParameterDegradation holds the parameters related to *ReactionDegradation* denoted by the letter d and, in the index, the name of the protein involved in the reaction. For example,

for the reaction



the corresponding parameter is

$$d_B$$

.

ReactionsOneSite[*activations*, *repressions*] returns the same prints and tables as *Reactions*, but counting only one binding site per gene. The proteins have to compete in order to bind to the gene.

ReactionGraph[*reactions*, *parameters*] displays the reactions with the corresponding parameters. An example of the kinetic mechanisms associated to the input

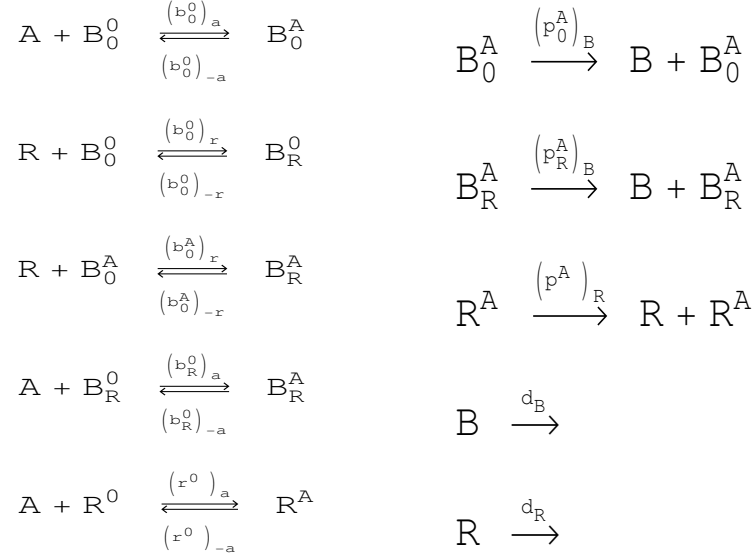


Figure 3: Reactions of the genetic network defined in (1)

in (1) is shown in figure 3.

SubstanceNames[reactions], *SubstanceVariable[reactions]*,
SubstanceInitialConditions[reactions]

return, respectively, a table with the name of the variables in *reactions*, the same table with the explicit temporal dependence (if any), the table with the substance variables taken at time $t=0$ and a set of global variables in order to assign externally initial conditions.

ParameterNames[parameters], *ParameterInput[parameters]*

compute, respectively, a table with the name of the parameters in *reactions* independently of the graph of the activations and repressions, and the same table with a set of global variables in order to assign parameters values.

Equations[reactions, parameters]

calculates the differential equations describing the time evolution of the concentrations of the chemical substances involved in the set of reactions. The differential equations are obtained by the mass action law, and returned in a table. An example for input (1) is shown in figure 4.

ConservationLaws[reactions]

returns a table with the conservation laws of the full set of reactions. An example for input (1) is shown in figure 5.

$$\begin{aligned}
A'[t] &= -A[t] \left(b_0^0 \right)_a B_0^0[t] + \left(b_0^0 \right)_{-a} B_0^A[t] - A[t] \left(b_R^0 \right)_a B_R^0[t] + \left(b_R^0 \right)_{-a} B_R^A[t] - A[t] \left(r^0 \right)_a R^0[t] + \left(r^0 \right)_{-a} R^A[t] \\
B'[t] &= -B[t] d_B + (p_0^A)_B B_0^A[t] + (p_R^A)_B B_R^A[t] \\
R'[t] &= -R[t] d_R - R[t] \left(b_0^0 \right)_r B_0^0[t] - R[t] \left(b_0^A \right)_r B_0^A[t] + \left(b_0^0 \right)_{-r} B_R^0[t] + \left(b_0^A \right)_{-r} B_R^A[t] + (p^A)_R R^A[t] \\
\left(B_0^0 \right)'[t] &= -A[t] \left(b_0^0 \right)_a B_0^0[t] - R[t] \left(b_0^0 \right)_r B_0^0[t] + \left(b_0^0 \right)_{-a} B_0^A[t] + \left(b_0^0 \right)_{-r} B_R^0[t] \\
\left(B_0^A \right)'[t] &= A[t] \left(b_0^0 \right)_a B_0^0[t] - \left(b_0^0 \right)_{-a} B_0^A[t] - R[t] \left(b_0^A \right)_r B_0^A[t] + \left(b_0^A \right)_{-r} B_R^A[t] \\
\left(B_R^0 \right)'[t] &= R[t] \left(b_0^0 \right)_r B_0^0[t] - \left(b_0^0 \right)_{-r} B_R^0[t] - A[t] \left(b_R^0 \right)_a B_R^0[t] + \left(b_R^0 \right)_{-a} B_R^A[t] \\
\left(B_R^A \right)'[t] &= R[t] \left(b_0^A \right)_r B_0^A[t] + A[t] \left(b_R^0 \right)_a B_R^0[t] - \left(b_0^A \right)_{-r} B_R^A[t] - \left(b_R^0 \right)_{-a} B_R^A[t] \\
(G_1)'[t] &= B[t] d_B \\
(G_2)'[t] &= R[t] d_R \\
\left(R^0 \right)'[t] &= -A[t] \left(r^0 \right)_a R^0[t] + \left(r^0 \right)_{-a} R^A[t] \\
\left(R^A \right)'[t] &= A[t] \left(r^0 \right)_a R^0[t] - \left(r^0 \right)_{-a} R^A[t]
\end{aligned}$$

Figure 4: Equations of the genetic network defined in (1).

$$\begin{aligned}
&R^0[t] + R^A[t] \\
&B_0^0[t] + B_0^A[t] + B_R^0[t] + B_R^A[t] \\
&A
\end{aligned}$$

Figure 5: Conservation laws of the genetic network defined in (1)

EmptyGenes[*real value*], *NotEmptyGenes*[*real value*]

assign the value in input to the initial conditions of every empty, or not empty, gene.

EmptyGenesPosition[*substanceInitialConditions*, *emptyGenes*]

returns the positions of the empty genes in the table *substanceInitialConditions*.

WriteDataC[]

generates the output file *data.txt* with the parameters and the initial conditions used in the current network.

CEquations[*equations*, *substanceVariables*, *parameters*]

generates the output file *equations.h* where the equations of the system are translated into a C function.

The functions listed above have been taken from a previous package, *Kinetics*, and adapted to evaluate the input of a genetic network. This package has been subsequently introduced in *GeneticNetworks*. Its aim is to compute the differential equations and the conservation laws of a general table of reactions, not necessarily related to a genetic network. In order to distinguish its functions from the ones described above, we gave them a name ending in *Kinetics*:

- ReactionGraphKinetics
- SubstanceNamesKinetics, SubstanceVariablesKinetics,
SubstanceInitialConditionsKinetics
- ParameterNamesKinetics, ParameterInputKinetics
- EquationsKinetics
- ConservationLawsKinetics

The input is a table holding all the reactions of the analyzed mechanism, and the parameters are automatically computed without a relation with a graph. Their structure is very similar to the ones previously described, and is accounted in the help of the package.

Bibliography

- [Alves and Dilão, 2005] F. Alves, R. Dilão (2005) A simple framework to describe the regulation of gene expression in prokaryotes, *C. R. Biologies*, **328**, 429-444.
- [Alves and Dilão, 2006] F. Alves, R. Dilão (2006) Modeling segmental patterning in *Drosophila*: Maternal and gap genes., *Journal of Theoretical Biology*, **241**, 342-359.
- [Alves and Dilão, 2006] F. Alves, R. Dilão (2006) A software tool to model genetic regulatory networks: applications to segmental patterning in *Drosophila*. *BIOMAT 2005, International Symposium on Mathematical and Computational Biology.*, R. P. Mondaini, R. Dilão, pp. 71-88, World Scientific, Singapore, 2006.
- [Dilão and Muraro, 2006] R. Dilão and D. Muraro (2009) Emergent thresholds in genetic regulatory networks: Protein patterning in *Drosophila* morphogenesis, pre-print.

GeneticNetworks

Commands

```
<< GeneticNetworks.m
```

```
? GeneticNetworks
```

GeneticNetworks.m is a package that models a gene regulatory network.

The following functions are defined in the GeneticNetworks package:

NetworkGraph

Reactions

SubstanceNames

SubstanceVariables

SubstanceInitialConditions

ParameterNames

ParameterInput

ReactionGraph

Equations

ConservationLaws

The first functions to run are NetworkGraph[activation, repression] and Reactions[activation, repression].

The input are two tables that indicate the activations and repressions of proteins on genes.

For example,

activation={A → B, B → C}

repression={B → A}.

This means that the protein A activates the gene B, the protein

B activates the gene C, and the protein B represses the gene A (please note that the same symbol is used to denote both a gene and the corresponding protein).

NetworkGraph[activation, repression] returns the graph of the activations (green arrow) and repressions (red edge).

Reactions[activation, repression] prints the names of the

proteins of the graph; the number of activation, repression and total sites of the corresponding gene; the name of the proteins which have no sites and that, by default, will be considered constant. This function also returns two tables, reactions and parameters,

containing all the reactions and the parameters of the graph accounting of the potential production and degradation of the proteins.

Genes are denoted by the name of the protein they produce

and with subscripts and superscripts indicating their regulator binding sites.

The subscripts are reserved to proteins which repress the gene, while the superscripts to proteins which activate it.

SubstanceNames[reactions], SubstanceVariable[reactions],

SubstanceInitialConditions[reactions] return, respectively, a table with the name of the variables in the table reactions, the same table with the explicit temporal dependence (if any), the table with the substance variables

taken at time $t=0$ and a set of global variables in order to assign externally initial conditions.

ParameterNames[reactions], ParameterInput[parameters]

compute, respectively, a table with the name of the parameters in the reactions independently of the graph of activations and repressions, and the same table with a set of global variables in order to assign parameter values.

ReactionGraph[reactions, parameters] displays the reactions with the corresponding parameters.

Equations[reactions, parameters] calculates the differential

equations describing the time evolution of the concentrations of genes and proteins of the genetic network.

The differential equations are obtained by the mass action

law, and the command Equations[reactions, parameters] returns a table with the differential equations.

ConservationLaws[reactions] returns a table with the conservation laws of the full set of reactions.

? NetworkGraph

NetworkGraph[activation, repression] returns the graph of a genetic regulatory network. The input are two tables that indicate the activations and repressions of proteins on genes.

For example,

activation={A → B, B → C}

repression={B → A}.

This means that the protein A activates the gene B, the protein

B activates the gene C, and the protein B represses the gene A (please note that the same symbol is used to denote both a gene and the corresponding protein).

The activation are drawn with a green arrow, while the repressions are drawn with a red edge.

? ManipulateGraph

ManipulateGraph[activation, repression] returns the graph of a genetic regulatory network in a form that can be manipulated. The input is the same of the function NetworkGraph

? Reactions

Reactions[activation, repression] returns the reactions

and the parameters involved in a genetic network. It also prints the names of the proteins and the numbers of the regulatory sites of the genes in the network. The input are two tables that indicate the activations and repressions of proteins on genes. For example,

activation={A \rightarrow B, B \rightarrow C}

repression={B \rightarrow A}.

This means that the protein A activates the gene B, the protein

B activates the gene C, and the protein B represses the gene A (please note that the same symbol is used to denote both a gene and the corresponding protein).

Reactions prints the names of the proteins of the graph; the

number of activation, of repression and of the total sites of the corresponding gene; the name of the proteins which have no sites and that, by default, will be considered constant.

This function then returns two tables, reactions and parameters,

containing all the reactions and the parameters of the graph accounting of the potential production and degradation of the proteins.

Genes are denoted by the name of the protein they produce

and with subscripts and superscripts indicating their regulator binding sites.

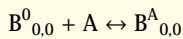
The subscripts are reserved to proteins which repress the gene, the superscripts to proteins which activate it.

The network of interactions can be divided into three different species, each stored in a global table:

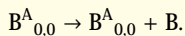
ReactionGeneProtein, ReactionProduction, ReactionDegradation.

The first table holds the double bimolecular

reactions between a gene and a protein activating or repressing it, i.e.



The second table holds the reactions of production of a protein by an activated gene, i.e.



The third table holds the degradations of proteins (if any), i.e. $B \rightarrow .$

Correspondingly, the following global tables for parameters are

defined: ParameterFw, ParameterBw, ParameterProduction, ParameterDegradation.

ParameterFw and ParameterBw hold the parameters related to ReactionGeneProtein.

The parameters are denoted by the small letter of the gene involved in the reaction and, in the index, the small letter of the protein reacting with the gene.

For the backward reaction the index is preceded by a minus sign. For example,

for the forward reaction $B_{0,0}^0 + A \rightarrow B_{0,0}^A$ the corresponding parameter is b_a ,

for the backward reaction $B_{0,0}^A \rightarrow B_{0,0}^0 + A$ the corresponding parameter is b_{-a} .

ParameterProduction holds the parameters related to

ReactionProduction denoted by the letter p and, in the index, the name of the protein involved in the reaction. For example,

for the reaction $B_{0,0}^A \rightarrow B_{0,0}^A + B$ the corresponding parameter is p_B .

ParameterDegradation holds the parameters related to

ReactionDegradation denoted by the letter d and, in the index, the name of the protein involved in the reaction. For example,

for the reaction $B \rightarrow .$ the corresponding parameter is d_B .

? ReactionsOneSite

ReactionOneSite[activation, repression] returns
 the same prints and tables of Reactions, but counting only one binding
 site per gene. The proteins have to compete to bind the gene.

? ReactionGraph

ReactionGraph[reactions] displays the reactions with the corresponding rate constants.

? SubstanceNames

SubstanceNames[reactions] returns a table with the name of the variables in the table reactions.

? SubstanceVariables

SubstanceVariables[reactions] returns a table with the name of the variables in the table reactions, with the explicit
 temporal dependence --- Name[t].

? SubstanceInitialConditions

SubstanceInitialConditions[reactions] returns a table with the substance variables taken at time t=0, and a
 set of global variables in order to assign externally initial conditions. The global variables are InitCond[i].

? ParameterInput

ParameterInput[parameters] returns a table with
 the name of the parameters in the reactions, and a set of global variables
 in order to assign parameter values. The global variables are ParInputVar[i].

? Equations

Equations[reactions, parameters] calculates the

differential equations describing the time evolution of the concentrations of the chemical substances involved in the set of chemical reactions of a genetic network.

The differential equations are obtained by the mass action

law, and the command Equations[reactions, parameters] returns a table with the differential equations.

The input to Equations[reactions, parameters] are the tables

of the reactions and of the corresponding parameters obtained by the function Reactions[activation, repression].

The proteins with a constant concentration in the output of

Reactions will be considered, by default, constant. Thus, for such proteins, there will be no differential equation.

Another feature of this package is the introduction of 'garbage' variables, denoted by $G_i (i=1, \dots, n)$, to compute also the degradations with the mass action law. For example,

the reaction $B \rightarrow$ will be substituted by $B \rightarrow$

G_1 . The equations with these garbage variables must not be considered.

? ConservationLaws

ConservationLaws[reactions] returns a table with the conservation laws of the full set of reactions.

The proteins with a constant concentration in the output of Reactions will be considered, by default, constant.

Thus, such proteins should appear among the conservation laws.

Another feature of this package is the introduction of 'garbage' variables, denoted by $G_i (i=1, \dots, n)$, to compute also the degradations with the mass action law. For example,

the reaction $B \rightarrow$ will be substituted by $B \rightarrow G_1$. The conservation laws with these garbage variables must not be considered.

? EmptyGenes

EmptyGenes[value] assigns the value in input to the initial conditions of every empty gene.

? NotEmptyGenes

NotEmptyGenes[value] assigns the value in input to the initial conditions of every not empty gene.

? EmptyGenesPosition

EmptyGenesPosition[substanceInitialConditions, emptyGenes]

returns the positions of the empty genes in the table 'substanceInitialConditions'

? WriteDataC

WriteDataC[] generates the output file 'data.txt'

with the parameters and the initial conditions used in the current network.

? CEquations

CEquations[equations, substanceVariables, parameters] generates the output file 'equations.h' where the equations of the system are translated into a C function.

? EulerIntegration

EulerIntegration[equations_, parameters_, initialConditions_, variables_, iterations_, step_] returns a table with the temporal evolution of the equations in input.

? EulerPoints

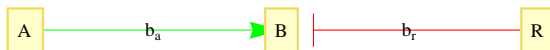
EulerPoints[equations_, parameters_, initialConditions_, variables_, iterations_, step_] returns the final points of the variables after the evolution of the equations in input.

Network 1 : one activator and one repressor

```

activations = {A → B};
repressions = {R → B};
NetworkGraph[activations, repressions]

```



```

Reactions[activations, repressions];

Print["\n\nGraph of reactions"];
ReactionGraph[reactions, parameters];

Print["\n\nSubstances"];
names = SubstanceNames[reactions];
variables = SubstanceVariables[reactions];
initialConditions = SubstanceInitialConditions[reactions];
Print[names]; Print[variables]; Print[initialConditions];

Print["\n\nParameters"];
parametersInput = ParameterInput[parameters];
Print[parametersInput];

Print["\n\nTime evolution equations"];
eqs = Equations[reactions, parameters];
Print[eqs // Column]

Print["\n\nConservation laws"];
cons = ConservationLaws[reactions];
Print[cons // Column];

```

Proteins = {A, B, R}

Activation Sites per Gene = {0, 1, 0}

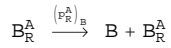
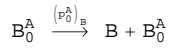
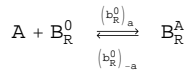
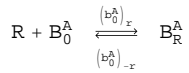
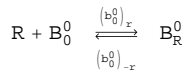
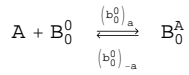
Repression Sites per Gene = {0, 1, 0}

Total Sites per Gene = {0, 2, 0}

Proteins Without Genes = {A, R}

Constant Proteins = {}

Graph of reactions



Substances

$\{A, B, R, B_0^0, B_0^A, B_R^0, B_R^A\}$

$\{A[t], B[t], R[t], B_0^0[t], B_0^A[t], B_R^0[t], B_R^A[t]\}$

$\{A[0] == \text{InitCond}[1], B[0] == \text{InitCond}[2], R[0] == \text{InitCond}[3],$
 $B_0^0[0] == \text{InitCond}[4], B_0^A[0] == \text{InitCond}[5], B_R^0[0] == \text{InitCond}[6], B_R^A[0] == \text{InitCond}[7]\}$

Parameters

$\{(b_0^0)_a \rightarrow \text{ParInputVar}[1], (b_0^0)_r \rightarrow \text{ParInputVar}[2], (b_0^A)_r \rightarrow \text{ParInputVar}[3], (b_R^0)_a \rightarrow \text{ParInputVar}[4],$
 $(p_0^A)_B \rightarrow \text{ParInputVar}[5], (p_R^A)_B \rightarrow \text{ParInputVar}[6], d_B \rightarrow \text{ParInputVar}[7], (b_0^0)_{-a} \rightarrow \text{ParInputVar}[8],$
 $(b_0^0)_{-r} \rightarrow \text{ParInputVar}[9], (b_0^A)_{-r} \rightarrow \text{ParInputVar}[10], (b_R^0)_{-a} \rightarrow \text{ParInputVar}[11]\}$

Time evolution equations

$$\begin{aligned}
A'[t] &= -A[t] \left(b_0^0 \right)_a B_0^0[t] + \left(b_0^0 \right)_{-a} B_0^A[t] - A[t] \left(b_R^0 \right)_a B_R^0[t] + \left(b_R^0 \right)_{-a} B_R^A[t] \\
B'[t] &= -B[t] d_B + \left(p_0^A \right)_B B_0^A[t] + \left(p_R^A \right)_B B_R^A[t] \\
R'[t] &= -R[t] \left(b_0^0 \right)_x B_0^0[t] - R[t] \left(b_0^A \right)_x B_0^A[t] + \left(b_0^0 \right)_{-x} B_R^0[t] + \left(b_0^A \right)_{-x} B_R^A[t] \\
\left(B_0^0 \right)'[t] &= -A[t] \left(b_0^0 \right)_a B_0^0[t] - R[t] \left(b_0^0 \right)_x B_0^0[t] + \left(b_0^0 \right)_{-a} B_0^A[t] + \left(b_0^0 \right)_{-x} B_R^0[t] \\
\left(B_0^A \right)'[t] &= A[t] \left(b_0^0 \right)_a B_0^0[t] - \left(b_0^0 \right)_{-a} B_0^A[t] - R[t] \left(b_0^A \right)_x B_0^A[t] + \left(b_0^A \right)_{-x} B_R^A[t] \\
\left(B_R^0 \right)'[t] &= R[t] \left(b_0^0 \right)_x B_0^0[t] - \left(b_0^0 \right)_{-x} B_R^0[t] - A[t] \left(b_R^0 \right)_a B_R^0[t] + \left(b_R^0 \right)_{-a} B_R^A[t] \\
\left(B_R^A \right)'[t] &= R[t] \left(b_0^A \right)_x B_0^A[t] + A[t] \left(b_R^0 \right)_a B_R^0[t] - \left(b_0^A \right)_{-x} B_R^A[t] - \left(b_R^0 \right)_{-a} B_R^A[t] \\
\left(G_1 \right)'[t] &= B[t] d_B
\end{aligned}$$

Conservation laws

$$\begin{aligned}
&B_0^0[t] + B_0^A[t] + B_R^0[t] + B_R^A[t] \\
&R \\
&A
\end{aligned}$$

```

(** PARAMETER INPUT **)

(* Forward *)
parametersFwValue = Table[1., {i, Length@ParametersFw}];
(*  $(b_0^0)_a$  *) ParInputVar[1] = parametersFwValue[[1]];
(*  $(b_0^0)_r$  *) ParInputVar[2] = parametersFwValue[[2]];
(*  $(b_0^A)_r$  *) ParInputVar[3] = parametersFwValue[[3]];
(*  $(b_R^0)_a$  *) ParInputVar[4] = parametersFwValue[[4]];

(** Productions **)
parametersProductionValue = Table[1., {i, Length@ParametersProduction}];
(*  $(p_0^A)_B$  *) ParInputVar[5] = parametersProductionValue[[1]];
(*  $(p_R^A)_B$  *) ParInputVar[6] = parametersProductionValue[[2]];

(** Degradations **)
parametersDegradationValue = Table[1., {i, Length@ParametersDegradation}];
(*  $d_B$  *) ParInputVar[7] = parametersDegradationValue[[1]];

(* Backward *)
parametersBwValue = Table[0.1, {i, Length@ParametersBw}];
(*  $(b_0^0)_{-a}$  *) ParInputVar[8] = parametersBwValue[[1]];
(*  $(b_0^0)_{-r}$  *) ParInputVar[9] = parametersBwValue[[2]];
(*  $(b_0^A)_{-r}$  *) ParInputVar[10] = parametersBwValue[[3]];
(*  $(b_R^0)_{-a}$  *) ParInputVar[11] = parametersBwValue[[4]];

(** INITIAL CONDITIONS INPUT **)
(* A *) InitCond[1] = 1;
(* B *) InitCond[2] = 0;
(* R *) InitCond[3] = 1;

EmptyGenesPosition[substanceInitialConditions, emptyGenes];
EmptyGenes[1.]
NotEmptyGenes[0.]

Print["Initial conditions"];
Print[initialConditions];
Print["\n\nParameters"];
Print[parametersInput];

```

Initial conditions

$$\{A[0] = 1, B[0] = 0., R[0] = 0., B_0^0[0] = 1., B_0^A[0] = 0., B_R^0[0] = 0., B_R^A[0] = 0.\}$$

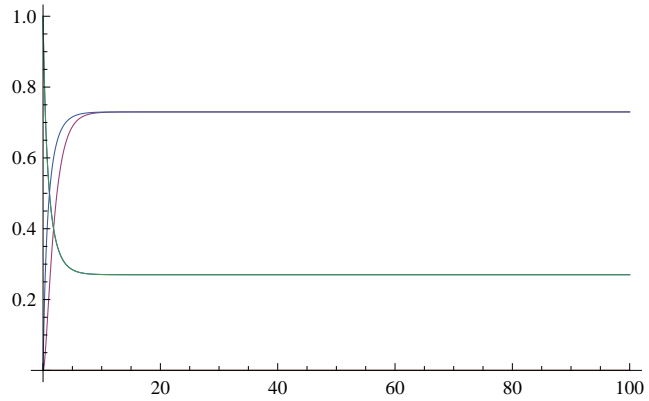
Parameters

$$\{(b_0^0)_a \rightarrow 1., (b_0^0)_r \rightarrow 1., (b_0^A)_r \rightarrow 1., (b_R^0)_a \rightarrow 1., (p_0^A)_B \rightarrow 1., (p_R^A)_B \rightarrow 1, d_B \rightarrow 1., (b_0^0)_{-a} \rightarrow 0.1, (b_0^0)_{-r} \rightarrow 0.1, (b_0^A)_{-r} \rightarrow 0.1, (b_R^0)_{-a} \rightarrow 0.1\}$$

```

Time = 100;
equations = Drop[equations, {8, 8}];
sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],
  substanceNames, {t, 0, Time}];
g1 = Plot[Evaluate[substanceVariables /. sol], {t, 0, Time}, PlotRange -> All]

```



```

(* Proportionality Constants *)
parametersPropConst = Table[1., {i, Length@Complement[E1, ConstantProteins]}}
{1., 1., 1.}

```

```

(* Write data file for C program *)
WriteDataC[]

# Vectors Lengths Variables ParametersRNA ParametersEmptyGenes ParametersFw ParametersProduction
7 0 1 4 2 1 4 3

# a1 a2 a3 a4 L1 L2 L3 L4

# Empty Genes B Value Position
1.
4

# b_O_O_a b_O_O_r b_A_O_r b_O_R_a
1. 1. 1. 1.

# pB_A_O pB_A_R
1. 1.

# d_B
1.

# b_O_O_-a b_O_O_-r b_A_O_-r b_O_R_-a
0.1 0.1 0.1 0.1

# Proportionality Constants A B R
1. 1. 1.

# Time
100.

(* Write equations for C program *)
CEquations[equations, substanceVariables, parameters]

void Equations(double *v, double *p, double *F)
{
  F[0] = -(p[0]*v[0]*v[3]) + p[7]*v[4] - p[3]*v[0]*v[5] + p[10]*v[6];
  F[1] = -(p[6]*v[1]) + p[4]*v[4] + p[5]*v[6];
  F[2] = -(p[1]*v[2]*v[3]) - p[2]*v[2]*v[4] + p[8]*v[5] + p[9]*v[6];
  F[3] = -(p[0]*v[0]*v[3]) - p[1]*v[2]*v[3] + p[7]*v[4] + p[8]*v[5];
  F[4] = p[0]*v[0]*v[3] - p[7]*v[4] - p[2]*v[2]*v[4] + p[9]*v[6];
  F[5] = p[1]*v[2]*v[3] - p[8]*v[5] - p[3]*v[0]*v[5] + p[10]*v[6];
  F[6] = p[2]*v[2]*v[4] + p[3]*v[0]*v[5] - p[9]*v[6] - p[10]*v[6];
}

(* Quit Kernel *)

<< GeneticNetworksV280709/GeneticNetworks.m

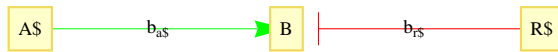
```

Network 2: one constant activator and one constant repressor

```

activations = {A$ -> B};
repressions = {R$ -> B};
NetworkGraph[activations, repressions]

```



```

Reactions[activations, repressions];

Print["\n\nGraph of reactions"];
ReactionGraph[reactions, parameters];

Print["\n\nSubstances"];
names = SubstanceNames[reactions];
variables = SubstanceVariables[reactions];
initialConditions = SubstanceInitialConditions[reactions];
Print[names]; Print[variables]; Print[initialConditions];

Print["\n\nParameters"];
parametersInput = ParameterInput[parameters];
Print[parametersInput];

Print["\n\nTime evolution equations"];
eqs = Equations[reactions, parameters];
Print[eqs // Column]

Print["\n\nConservation laws"];
cons = ConservationLaws[reactions];
Print[cons // Column];

```

```
Proteins = {A$, B, R$}
```

```
Activation Sites per Gene = {0, 1, 0}
```

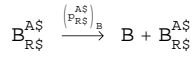
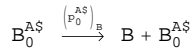
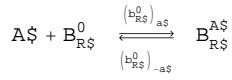
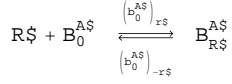
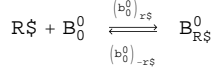
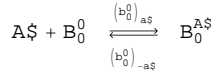
```
Repression Sites per Gene = {0, 1, 0}
```

```
Total Sites per Gene = {0, 2, 0}
```

```
Proteins Without Genes = {A$, R$}
```

```
Constant Proteins = {A$, R$}
```

Graph of reactions



Substances

$$\{B, B_0^0, B_0^{A\$}, B_{R\$}^0, B_{R\$}^{A\$}\}$$

$$\{B[t], B_0^0[t], B_0^{A\$}[t], B_{R\$}^0[t], B_{R\$}^{A\$}[t]\}$$

$$\{B[0] == \text{InitCond}[1], B_0^0[0] == \text{InitCond}[2], \\ B_0^{A\$}[0] == \text{InitCond}[3], B_{R\$}^0[0] == \text{InitCond}[4], B_{R\$}^{A\$}[0] == \text{InitCond}[5]\}$$

Parameters

$$\{(b_0^0)_{a\$} \rightarrow \text{ParInputVar}[1], (b_0^0)_{r\$} \rightarrow \text{ParInputVar}[2], (b_0^{A\$})_{r\$} \rightarrow \text{ParInputVar}[3], \\ (b_{R\$}^0)_{a\$} \rightarrow \text{ParInputVar}[4], (p_0^{A\$})_B \rightarrow \text{ParInputVar}[5], (p_{R\$}^{A\$})_B \rightarrow \text{ParInputVar}[6], d_B \rightarrow \text{ParInputVar}[7], \\ (b_0^0)_{-a\$} \rightarrow \text{ParInputVar}[8], (b_0^0)_{-r\$} \rightarrow \text{ParInputVar}[9], (b_0^{A\$})_{-r\$} \rightarrow \text{ParInputVar}[10], \\ (b_{R\$}^0)_{-a\$} \rightarrow \text{ParInputVar}[11], A\$ \rightarrow \text{ParInputVar}[12], R\$ \rightarrow \text{ParInputVar}[13]\}$$

Time evolution equations

$$\begin{aligned}
B'[t] &= -B[t] d_B + \left(p_{0\$}^{A\$}\right)_B B_0^{A\$}[t] + \left(p_{R\$}^{A\$}\right)_B B_{R\$}^{A\$}[t] \\
\left(B_0^0\right)'[t] &= -A\$ \left(b_0^0\right)_{a\$} B_0^0[t] - R\$ \left(b_0^0\right)_{r\$} B_0^0[t] + \left(b_0^0\right)_{-a\$} B_0^{A\$}[t] + \left(b_0^0\right)_{-r\$} B_{R\$}^0[t] \\
\left(B_0^{A\$}\right)'[t] &= A\$ \left(b_0^0\right)_{a\$} B_0^0[t] - \left(b_0^0\right)_{-a\$} B_0^{A\$}[t] - R\$ \left(b_0^{A\$}\right)_{r\$} B_0^{A\$}[t] + \left(b_0^{A\$}\right)_{-r\$} B_{R\$}^{A\$}[t] \\
\left(B_{R\$}^0\right)'[t] &= R\$ \left(b_0^0\right)_{r\$} B_0^0[t] - \left(b_0^0\right)_{-r\$} B_{R\$}^0[t] - A\$ \left(b_{R\$}^0\right)_{a\$} B_{R\$}^0[t] + \left(b_{R\$}^0\right)_{-a\$} B_{R\$}^{A\$}[t] \\
\left(B_{R\$}^{A\$}\right)'[t] &= R\$ \left(b_0^{A\$}\right)_{r\$} B_0^{A\$}[t] + A\$ \left(b_{R\$}^0\right)_{a\$} B_{R\$}^0[t] - \left(b_0^{A\$}\right)_{-r\$} B_{R\$}^{A\$}[t] - \left(b_{R\$}^0\right)_{-a\$} B_{R\$}^{A\$}[t] \\
\left(G_1\right)'[t] &= B[t] d_B
\end{aligned}$$

Conservation laws

$$\begin{aligned}
&B_0^0[t] + B_0^{A\$}[t] + B_{R\$}^0[t] + B_{R\$}^{A\$}[t] \\
&R\$ \\
&A\$
\end{aligned}$$

```

(** PARAMETER INPUT **)

(* Forward *)
parametersFwValue = Table[1., {i, Length@ParametersFw}];
(*  $(b_0^0)_a$  *) ParInputVar[1] = parametersFwValue[[1]];
(*  $(b_0^0)_r$  *) ParInputVar[2] = parametersFwValue[[2]];
(*  $(b_0^A)_r$  *) ParInputVar[3] = parametersFwValue[[3]];
(*  $(b_R^0)_a$  *) ParInputVar[4] = parametersFwValue[[4]];

(** Productions **)
parametersProductionValue = Table[1., {i, Length@ParametersProduction}];
(*  $(p_0^A)_B$  *) ParInputVar[5] = parametersProductionValue[[1]];
(*  $(p_R^A)_B$  *) ParInputVar[6] = parametersProductionValue[[2]];

(** Degradations **)
parametersDegradationValue = Table[1., {i, Length@ParametersDegradation}];
(*  $d_B$  *) ParInputVar[7] = parametersDegradationValue[[1]];

(* Backward *)
parametersBwValue = Table[0.1, {i, Length@ParametersBw}];
(*  $(b_0^0)_{-a}$  *) ParInputVar[8] = parametersBwValue[[1]];
(*  $(b_0^0)_{-r}$  *) ParInputVar[9] = parametersBwValue[[2]];
(*  $(b_0^A)_{-r}$  *) ParInputVar[10] = parametersBwValue[[3]];
(*  $(b_R^0)_{-a}$  *) ParInputVar[11] = parametersBwValue[[4]];

(** Constant Proteins **)
(* A *) ParInputVar[12] = 1;
(* R *) ParInputVar[13] = 1;

(* Initial Conditions *)
(* B *) InitCond[1] = 1;

EmptyGenesPosition[substanceInitialConditions, emptyGenes];
EmptyGenes[1.]
NotEmptyGenes[0.]

Print["Initial conditions"];
Print[initialConditions];
Print["\n\nParameters"];
Print[parametersInput];

```

Initial conditions

$$\{B[0] = 1, B_0^0[0] = 1., B_0^{AS}[0] = 0., B_{RS}^0[0] = 0., B_{RS}^{AS}[0] = 0.\}$$

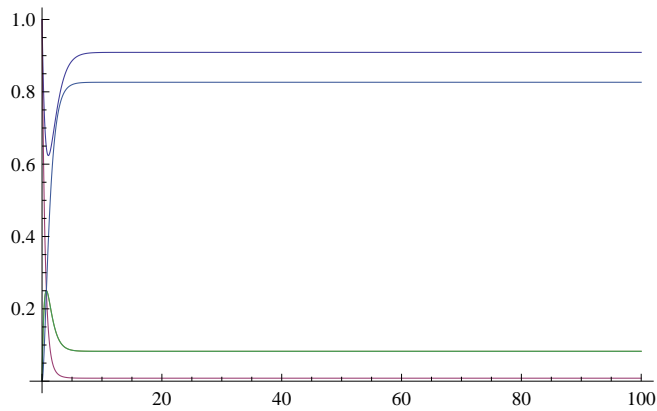
Parameters

$$\{(b_0^0)_{a\$} \rightarrow 1., (b_0^0)_{r\$} \rightarrow 1., (b_0^{AS})_{r\$} \rightarrow 1., (b_{RS}^0)_{a\$} \rightarrow 1., (p_0^{AS})_B \rightarrow 1, (p_{RS}^{AS})_B \rightarrow 1, \\ d_B \rightarrow 1., (b_0^0)_{-a\$} \rightarrow 0.1, (b_0^0)_{-r\$} \rightarrow 0.1, (b_0^{AS})_{-r\$} \rightarrow 0.1, (b_{RS}^0)_{-a\$} \rightarrow 0.1, A\$ \rightarrow 1, R\$ \rightarrow 1\}$$

```

Time = 100;
equations = Drop[equations, {6, 6}];
sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],
  substanceNames, {t, 0, Time}];
g1 = Plot[Evaluate[substanceVariables /. sol], {t, 0, Time}, PlotRange -> All]

```



```

(* Proportionality Constants *)
parametersPropConst = Table[1., {i, Length@Complement[E1, ConstantProteins]}}
{1.}

```

```

(* Write data file for C program *)
WriteDataC[]

# Vectors Lengths Variables ParametersRNA ParametersEmptyGenes ParametersFw ParametersProduction
5 0 1 4 2 1 4 1

# a1 a2 a3 a4 L1 L2 L3 L4

# Empty Genes B Value Position
1.
2

# b_O_O_a$ b_O_O_r$ b_A$_O_r$ b_O_R$_a$
1. 1. 1. 1.

# pB_A$_O pB_A$_R$
1. 1.

# d_B
1.

# b_O_O_-a$ b_O_O_-r$ b_A$_O_-r$ b_O_R$_-a$
0.1 0.1 0.1 0.1

# Proportionality Constants B
1.

# Time
100.

(* Write equations file for C program *)
CEquations[equations, substanceVariables, parameters]

void Equations(double *v, double *p, double *F)
{
  F[0] = -(p[6]*v[0]) + p[4]*v[2] + p[5]*v[4];
  F[1] = -(p[0]*p[11]*v[1]) - p[1]*p[12]*v[1] + p[7]*v[2] + p[8]*v[3];
  F[2] = p[0]*p[11]*v[1] - p[7]*v[2] - p[2]*p[12]*v[2] + p[9]*v[4];
  F[3] = p[1]*p[12]*v[1] - p[8]*v[3] - p[3]*p[11]*v[3] + p[10]*v[4];
  F[4] = p[2]*p[12]*v[2] + p[3]*p[11]*v[3] - p[9]*v[4] - p[10]*v[4];
}

(** All sites - one site: Spatial Comparison **)

(* Quit Kernel *)

<< GeneticNetworksV280709/GeneticNetworks.mks

```

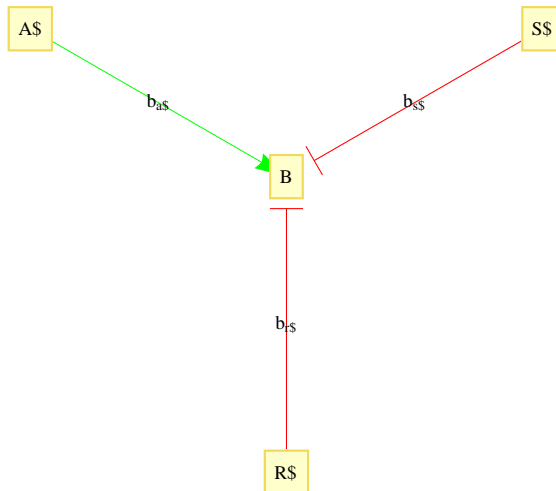
Network 3 :

spatial comparison for a network with and without competition of proteins

```

activations = {A$ -> B};
repressions = {R$ -> B, S$ -> B};
NetworkGraph[activations, repressions]

```



```

Reactions[activations, repressions];

Print["\n\nGraph of reactions"];
ReactionGraph[reactions, parameters];

Print["\n\nSubstances"];
names = SubstanceNames[reactions];
variables = SubstanceVariables[reactions];
initialConditions = SubstanceInitialConditions[reactions];
Print[names]; Print[variables]; Print[initialConditions];

Print["\n\nParameters"];
parametersInput = ParameterInput[parameters];
Print[parametersInput];

Print["\n\nTime evolution equations"];
eqs = Equations[reactions, parameters];
Print[eqs // Column]

Print["\n\nConservation laws"];
cons = ConservationLaws[reactions];
Print[cons // Column];

```

Proteins = {A\$, B, R\$, S\$}

Activation Sites per Gene = {0, 1, 0, 0}

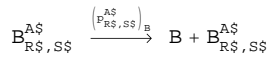
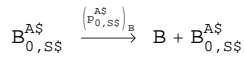
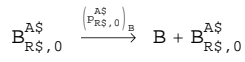
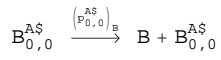
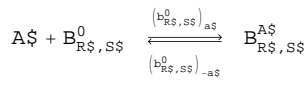
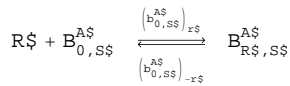
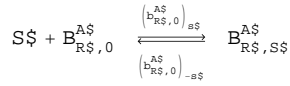
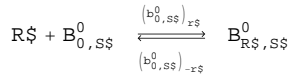
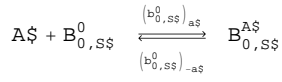
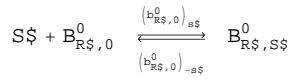
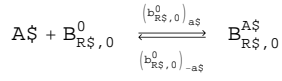
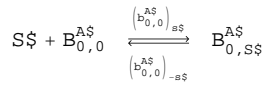
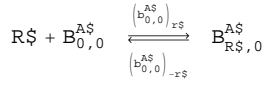
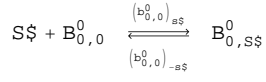
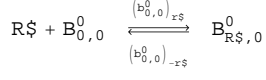
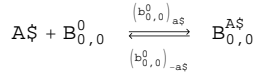
Repression Sites per Gene = {0, 2, 0, 0}

Total Sites per Gene = {0, 3, 0, 0}

Proteins Without Genes = {A\$, R\$, S\$}

Constant Proteins = {A\$, R\$, S\$}

Graph of reactions



Substances

$$\begin{aligned}
& \{B, B_{0,0}^0, B_{0,0}^{A\$}, B_{0,S\$}^0, B_{0,S\$}^{A\$}, B_{R\$,0}^0, B_{R\$,0}^{A\$}, B_{R\$,S\$}^0, B_{R\$,S\$}^{A\$}\} \\
& \{B[t], B_{0,0}^0[t], B_{0,0}^{A\$}[t], B_{0,S\$}^0[t], B_{0,S\$}^{A\$}[t], B_{R\$,0}^0[t], B_{R\$,0}^{A\$}[t], B_{R\$,S\$}^0[t], B_{R\$,S\$}^{A\$}[t]\} \\
& \{B[0] == \text{InitCond}[1], B_{0,0}^0[0] == \text{InitCond}[2], B_{0,0}^{A\$}[0] == \text{InitCond}[3], \\
& B_{0,S\$}^0[0] == \text{InitCond}[4], B_{0,S\$}^{A\$}[0] == \text{InitCond}[5], B_{R\$,0}^0[0] == \text{InitCond}[6], \\
& B_{R\$,0}^{A\$}[0] == \text{InitCond}[7], B_{R\$,S\$}^0[0] == \text{InitCond}[8], B_{R\$,S\$}^{A\$}[0] == \text{InitCond}[9]\}
\end{aligned}$$

Parameters

$$\begin{aligned}
& \{(b_{0,0}^0)_{a\$} \rightarrow \text{ParInputVar}[1], (b_{0,0}^0)_{r\$} \rightarrow \text{ParInputVar}[2], \\
& (b_{0,0}^0)_{s\$} \rightarrow \text{ParInputVar}[3], (b_{0,0}^{A\$})_{r\$} \rightarrow \text{ParInputVar}[4], (b_{0,0}^{A\$})_{s\$} \rightarrow \text{ParInputVar}[5], \\
& (b_{R\$,0}^0)_{a\$} \rightarrow \text{ParInputVar}[6], (b_{R\$,0}^0)_{s\$} \rightarrow \text{ParInputVar}[7], (b_{0,S\$}^0)_{a\$} \rightarrow \text{ParInputVar}[8], \\
& (b_{0,S\$}^0)_{r\$} \rightarrow \text{ParInputVar}[9], (b_{R\$,0}^{A\$})_{s\$} \rightarrow \text{ParInputVar}[10], (b_{0,S\$}^{A\$})_{r\$} \rightarrow \text{ParInputVar}[11], \\
& (b_{R\$,S\$}^0)_{a\$} \rightarrow \text{ParInputVar}[12], (p_{0,0}^{A\$})_B \rightarrow \text{ParInputVar}[13], (p_{R\$,0}^{A\$})_B \rightarrow \text{ParInputVar}[14], \\
& (p_{0,S\$}^{A\$})_B \rightarrow \text{ParInputVar}[15], (p_{R\$,S\$}^{A\$})_B \rightarrow \text{ParInputVar}[16], d_B \rightarrow \text{ParInputVar}[17], \\
& (b_{0,0}^0)_{-a\$} \rightarrow \text{ParInputVar}[18], (b_{0,0}^0)_{-r\$} \rightarrow \text{ParInputVar}[19], (b_{0,0}^0)_{-s\$} \rightarrow \text{ParInputVar}[20], \\
& (b_{0,0}^{A\$})_{-r\$} \rightarrow \text{ParInputVar}[21], (b_{0,0}^{A\$})_{-s\$} \rightarrow \text{ParInputVar}[22], (b_{R\$,0}^0)_{-a\$} \rightarrow \text{ParInputVar}[23], \\
& (b_{R\$,0}^0)_{-s\$} \rightarrow \text{ParInputVar}[24], (b_{0,S\$}^0)_{-a\$} \rightarrow \text{ParInputVar}[25], (b_{0,S\$}^0)_{-r\$} \rightarrow \text{ParInputVar}[26], \\
& (b_{R\$,0}^{A\$})_{-s\$} \rightarrow \text{ParInputVar}[27], (b_{0,S\$}^{A\$})_{-r\$} \rightarrow \text{ParInputVar}[28], (b_{R\$,S\$}^0)_{-a\$} \rightarrow \text{ParInputVar}[29], \\
& A\$ \rightarrow \text{ParInputVar}[30], R\$ \rightarrow \text{ParInputVar}[31], S\$ \rightarrow \text{ParInputVar}[32]\}
\end{aligned}$$

Time evolution equations

$$\begin{aligned}
B'[t] &= -B[t] d_B + \left(p_{0,0}^{AS}\right)_B B_{0,0}^{AS}[t] + \left(p_{0,S\$}^{AS}\right)_B B_{0,S\$}^{AS}[t] + \left(p_{R\$,0}^{AS}\right)_B B_{R\$,0}^{AS}[t] + \left(p_{R\$,S\$}^{AS}\right)_B B_{R\$,S\$}^{AS}[t] \\
(B_{0,0}^0)'[t] &= -A\$ (b_{0,0}^0)_{a\$} B_{0,0}^0[t] - R\$ (b_{0,0}^0)_{r\$} B_{0,0}^0[t] - \\
&\quad S\$ (b_{0,0}^0)_{s\$} B_{0,0}^0[t] + (b_{0,0}^0)_{-a\$} B_{0,0}^0[t] + (b_{0,0}^0)_{-s\$} B_{0,S\$}^0[t] + (b_{0,0}^0)_{-r\$} B_{R\$,0}^0[t] \\
(B_{0,S\$}^{AS})'[t] &= A\$ (b_{0,0}^0)_{a\$} B_{0,0}^0[t] - (b_{0,0}^0)_{-a\$} B_{0,0}^{AS}[t] - \\
&\quad R\$ (b_{0,0}^{AS})_{r\$} B_{0,0}^{AS}[t] - S\$ (b_{0,0}^{AS})_{s\$} B_{0,0}^{AS}[t] + (b_{0,0}^{AS})_{-s\$} B_{0,S\$}^{AS}[t] + (b_{0,0}^{AS})_{-r\$} B_{R\$,0}^{AS}[t] \\
(B_{0,S\$}^0)'[t] &= S\$ (b_{0,0}^0)_{s\$} B_{0,0}^0[t] - (b_{0,0}^0)_{-s\$} B_{0,S\$}^0[t] - \\
&\quad A\$ (b_{0,S\$}^0)_{a\$} B_{0,S\$}^0[t] - R\$ (b_{0,S\$}^0)_{r\$} B_{0,S\$}^0[t] + (b_{0,S\$}^0)_{-a\$} B_{0,S\$}^{AS}[t] + (b_{0,S\$}^0)_{-r\$} B_{R\$,S\$}^0[t] \\
(B_{0,S\$}^{AS})'[t] &= S\$ (b_{0,S\$}^{AS})_{s\$} B_{0,S\$}^{AS}[t] + A\$ (b_{0,S\$}^0)_{a\$} B_{0,S\$}^0[t] - \\
&\quad (b_{0,S\$}^{AS})_{-s\$} B_{0,S\$}^{AS}[t] - (b_{0,S\$}^0)_{-a\$} B_{0,S\$}^{AS}[t] - R\$ (b_{0,S\$}^{AS})_{r\$} B_{0,S\$}^{AS}[t] + (b_{0,S\$}^{AS})_{-r\$} B_{R\$,S\$}^{AS}[t] \\
(B_{R\$,0}^0)'[t] &= R\$ (b_{0,0}^0)_{r\$} B_{0,0}^0[t] - (b_{0,0}^0)_{-r\$} B_{R\$,0}^0[t] - \\
&\quad A\$ (b_{R\$,0}^0)_{a\$} B_{R\$,0}^0[t] - S\$ (b_{R\$,0}^0)_{s\$} B_{R\$,0}^0[t] + (b_{R\$,0}^0)_{-a\$} B_{R\$,0}^{AS}[t] + (b_{R\$,0}^0)_{-s\$} B_{R\$,S\$}^0[t] \\
(B_{R\$,0}^{AS})'[t] &= R\$ (b_{R\$,0}^{AS})_{r\$} B_{R\$,0}^{AS}[t] + A\$ (b_{R\$,0}^0)_{a\$} B_{R\$,0}^0[t] - \\
&\quad (b_{R\$,0}^{AS})_{-r\$} B_{R\$,0}^{AS}[t] - (b_{R\$,0}^0)_{-a\$} B_{R\$,0}^{AS}[t] - S\$ (b_{R\$,0}^{AS})_{s\$} B_{R\$,0}^{AS}[t] + (b_{R\$,0}^{AS})_{-s\$} B_{R\$,S\$}^{AS}[t] \\
(B_{R\$,S\$}^0)'[t] &= R\$ (b_{0,S\$}^0)_{r\$} B_{0,S\$}^0[t] + S\$ (b_{R\$,0}^0)_{s\$} B_{R\$,0}^0[t] - \\
&\quad (b_{0,S\$}^0)_{-r\$} B_{R\$,S\$}^0[t] - (b_{R\$,0}^0)_{-s\$} B_{R\$,S\$}^0[t] - A\$ (b_{R\$,S\$}^0)_{a\$} B_{R\$,S\$}^0[t] + (b_{R\$,S\$}^0)_{-a\$} B_{R\$,S\$}^{AS}[t] \\
(B_{R\$,S\$}^{AS})'[t] &= R\$ (b_{R\$,S\$}^{AS})_{r\$} B_{R\$,S\$}^{AS}[t] + S\$ (b_{R\$,0}^{AS})_{s\$} B_{R\$,0}^{AS}[t] + \\
&\quad A\$ (b_{R\$,S\$}^0)_{a\$} B_{R\$,S\$}^0[t] - (b_{R\$,S\$}^{AS})_{-r\$} B_{R\$,S\$}^{AS}[t] - (b_{R\$,0}^{AS})_{-s\$} B_{R\$,S\$}^{AS}[t] - (b_{R\$,S\$}^0)_{-a\$} B_{R\$,S\$}^{AS}[t] \\
(G_1)'[t] &= B[t] d_B
\end{aligned}$$

Conservation laws

$$\begin{aligned}
&B_{0,0}^0[t] + B_{0,0}^{AS}[t] + B_{0,S\$}^0[t] + B_{0,S\$}^{AS}[t] + B_{R\$,0}^0[t] + B_{R\$,0}^{AS}[t] + B_{R\$,S\$}^0[t] + B_{R\$,S\$}^{AS}[t] \\
&S\$ \\
&R\$ \\
&A\$
\end{aligned}$$

```

(* Parameters *)
Do[ParInputVar[i] = 1., {i, 1, 12}];
Do[ParInputVar[i] = 0.1, {i, 18, Length[parametersInput] - 3}];

ParInputVar[13] = 10.;
ParInputVar[14] = 0.;
ParInputVar[15] = 0.;
ParInputVar[16] = 0.;

ParInputVar[17] = 0.1;

ParInputVar[30] = 1.;
ParInputVar[31] = 0.1;
ParInputVar[32] = 0.1;

(* Initial Conditions *)
(* B[0] *) InitCond[1] = 0.;
EmptyGenes[1.]
NotEmptyGenes[0.]

Print["Initial conditions"];
Print[initialConditions];
Print["\n\nParameters"];
Print[parametersInput];

```

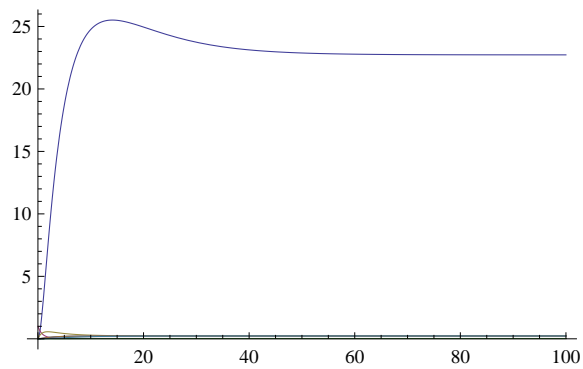
Initial conditions

$$\{B[0] = 0., B_{0,0}^0[0] = 1., B_{0,0}^{AS}[0] = 0., B_{0,S\$}[0] = 0., B_{0,S\$}^{AS}[0] = 0., B_{R\$,0}^0[0] = 0., B_{R\$,0}^{AS}[0] = 0., B_{R\$,S\$}^0[0] = 0., B_{R\$,S\$}^{AS}[0] = 0.\}$$

Parameters

$$\{ (b_{0,0}^0)_{a\$} \rightarrow 1., (b_{0,0}^0)_{r\$} \rightarrow 1., (b_{0,0}^0)_{s\$} \rightarrow 1., (b_{0,0}^{AS})_{r\$} \rightarrow 1., (b_{0,0}^{AS})_{s\$} \rightarrow 1., (b_{R\$,0}^0)_{a\$} \rightarrow 1., (b_{R\$,0}^0)_{s\$} \rightarrow 1., (b_{0,S\$}^0)_{a\$} \rightarrow 1., (b_{0,S\$}^0)_{r\$} \rightarrow 1., (b_{R\$,S\$}^0)_{a\$} \rightarrow 1., (p_{0,0}^{AS})_B \rightarrow 10., (p_{R\$,0}^{AS})_B \rightarrow 0., (p_{0,S\$}^{AS})_B \rightarrow 0., (p_{R\$,S\$}^{AS})_B \rightarrow 0., d_B \rightarrow 0.1, (b_{0,0}^0)_{-a\$} \rightarrow 0.1, (b_{0,0}^0)_{-r\$} \rightarrow 0.1, (b_{0,0}^0)_{-s\$} \rightarrow 0.1, (b_{0,0}^{AS})_{-r\$} \rightarrow 0.1, (b_{0,0}^{AS})_{-s\$} \rightarrow 0.1, (b_{R\$,0}^0)_{-a\$} \rightarrow 0.1, (b_{R\$,0}^0)_{-s\$} \rightarrow 0.1, (b_{0,S\$}^0)_{-a\$} \rightarrow 0.1, (b_{0,S\$}^0)_{-r\$} \rightarrow 0.1, (b_{R\$,S\$}^0)_{-s\$} \rightarrow 0.1, (b_{0,S\$}^{AS})_{-r\$} \rightarrow 0.1, (b_{0,S\$}^{AS})_{-s\$} \rightarrow 0.1, (b_{R\$,S\$}^0)_{-a\$} \rightarrow 0.1, A\$ \rightarrow 1., R\$ \rightarrow 0.1, S\$ \rightarrow 0.1 \}$$

```
Time = 100;  
equations = Drop[equations, -1];  
sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],  
  substanceNames, {t, 0, Time}];  
g1 = Plot[Evaluate[substanceVariables /. sol], {t, 0, Time}, PlotRange -> All]
```

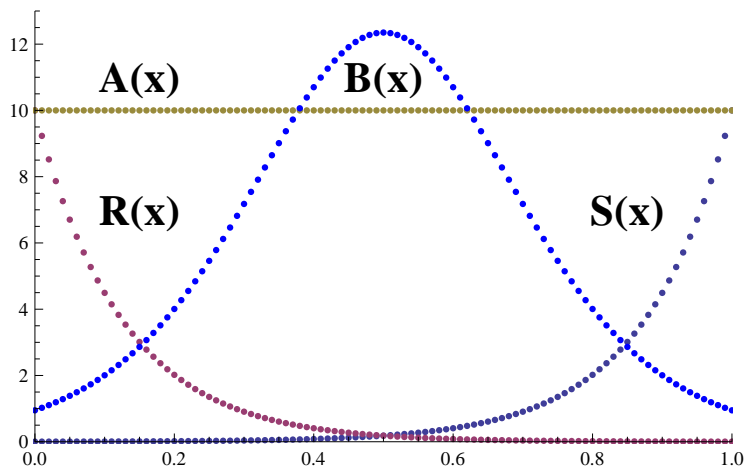


```

(* Spatial Integration *)

Tfin = 100; step = 0.01;
table = {}; table1 = {}; table2 = {}; table3 = {};
For[s1 = 0, s1 ≤ 1, s1 += step,
  (* A, R1, R2 *)
  ParInputVar[30] = 10.;
  ParInputVar[31] = 10 E(-8 s1);
  ParInputVar[32] = 10 E(-8 (1-s1));
  sol = NDSolve[Join[equations, substanceInitialConditions] /. parametersInput,
    substanceNames, {t, 0, Tfin}];
  table = Append[table, {s1, Evaluate[B[t] /. sol][[1]] /. t → Tfin}];
  table1 = Append[table1, {s1, ParInputVar[32]}];
  table2 = Append[table2, {s1, ParInputVar[31]}];
  table3 = Append[table3, {s1, ParInputVar[30]}];
];
ActRep = ListPlot[{table1, table2, table3}, PlotRange → All];
ListFit = table;
t = ListPlot[ListFit, PlotStyle → Blue];
(*g4=ListPlot[{table,table1,table2,table3},PlotRange→All]*)
g1 = Show[ActRep, t, Graphics[Text["A(x)", {0.15, 11},
  TextStyle -> {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold"}]],
Graphics[Text["R(x)", {0.15, 7}, TextStyle ->
  {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold"}]],
Graphics[Text["S(x)", {0.85, 7}, TextStyle ->
  {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold"}]],
Graphics[Text["B(x)", {0.5, 11}, TextStyle ->
  {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold"}]],
FrameLabel -> {"t", ""}, DisplayFunction -> $DisplayFunction,
ImageSize -> 500, PlotRange -> {{0, 1}, {0, 13}}]

```



```

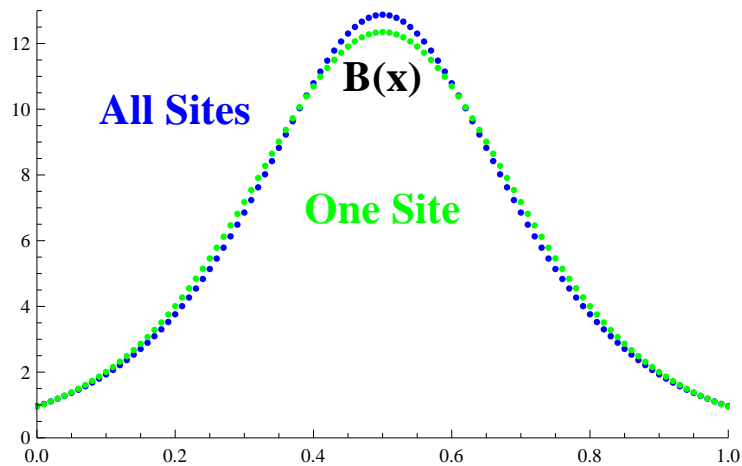
(* Steady state for the system with one binding site *)
A[x_] := Piecewise[{{a, 0 ≤ x ≤ 1.}}]
R[x_] := r E(-slope x)
S[x_] := r E(-slope (1-x))

Bsteady[x_] := 
$$\frac{c A[x]}{(1 + A[x] + R[x] + S[x])}$$
;

parInput2 = {a → ParProteins[1], r → ParProteins[2], slope → ParProteins[3], c → 100.};
(* Best parameters found by an optimization algorithm *)
BestFound = {0.825512, 83.8082, 7.19769};

```

```
(* Comparison of the spatial distributions *)
ListBestFound = {};
ParProteins[1] = BestFound[[1]];
ParProteins[2] = BestFound[[2]];
ParProteins[3] = BestFound[[3]];
AppendTo[ListBestFound, Table[{x, Bsteady[x] /. parInput2}, {x, 0, 1, step}]];
ListBestFound = Flatten[ListBestFound, 1];
fit = ListPlot[ListBestFound, PlotStyle -> Blue];
dataplot = ListPlot[ListFit, PlotStyle -> Green];
g2 = Show[fit, dataplot, Graphics[Text["One Site", {0.5, 7}, TextStyle ->
  {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Green}]],
Graphics[Text["All Sites", {0.2, 10}, TextStyle ->
  {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Blue}]],
Graphics[Text["B(x)", {0.5, 11}, TextStyle ->
  {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold"}]],
FrameLabel -> {"t", ""}, DisplayFunction -> $DisplayFunction,
ImageSize -> 500, PlotRange -> {{0, 1}, {0, 13}}]
```



```
(* Quit Kernel *)
```

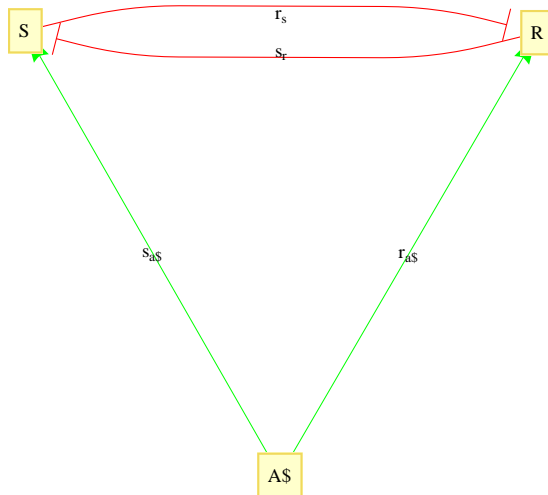
```
<< GeneticNetworksV280709/GeneticNetworks.m
```

Network 4: Threshold phenomenon

```

activations = {A$ -> R, A$ -> S};
repressions = {R -> S, S -> R};
NetworkGraph[activations, repressions]

```



```

Reactions[activations, repressions];

Print["\n\nGraph of reactions"];
ReactionGraph[reactions, parameters];

Print["\n\nSubstances"];
names = SubstanceNames[reactions];
variables = SubstanceVariables[reactions];
initialConditions = SubstanceInitialConditions[reactions];
Print[names]; Print[variables]; Print[initialConditions];

Print["\n\nParameters"];
parametersInput = ParameterInput[parameters];
Print[parametersInput];

Print["\n\nTime evolution equations"];
eqs = Equations[reactions, parameters];
Print[eqs // Column]

Print["\n\nConservation laws"];
cons = ConservationLaws[reactions];
Print[cons // Column];

```

Proteins = {A\$, R, S}

Activation Sites per Gene = {0, 1, 1}

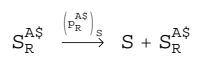
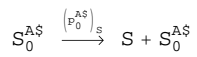
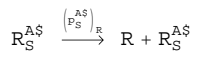
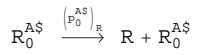
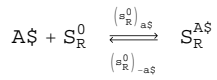
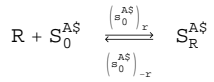
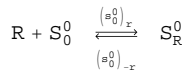
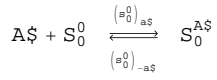
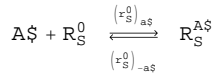
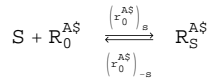
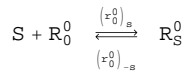
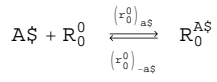
Repression Sites per Gene = {0, 1, 1}

Total Sites per Gene = {0, 2, 2}

Proteins Without Genes = {A\$}

Constant Proteins = {A\$}

Graph of reactions



Substances

$$\begin{aligned} & \{R, S, R_0^0, R_0^{AS}, R_S^0, R_S^{AS}, S_0^0, S_0^{AS}, S_R^0, S_R^{AS}\} \\ & \{R[t], S[t], R_0^0[t], R_0^{AS}[t], R_S^0[t], R_S^{AS}[t], S_0^0[t], S_0^{AS}[t], S_R^0[t], S_R^{AS}[t]\} \\ & \{R[0] == \text{InitCond}[1], S[0] == \text{InitCond}[2], R_0^0[0] == \text{InitCond}[3], \\ & R_0^{AS}[0] == \text{InitCond}[4], R_S^0[0] == \text{InitCond}[5], R_S^{AS}[0] == \text{InitCond}[6], S_0^0[0] == \text{InitCond}[7], \\ & S_0^{AS}[0] == \text{InitCond}[8], S_R^0[0] == \text{InitCond}[9], S_R^{AS}[0] == \text{InitCond}[10]\} \end{aligned}$$

Parameters

$$\begin{aligned} & \{(r_0^0)_{a\$} \rightarrow \text{ParInputVar}[1], (r_0^0)_s \rightarrow \text{ParInputVar}[2], (r_0^{AS})_s \rightarrow \text{ParInputVar}[3], (r_s^0)_{a\$} \rightarrow \text{ParInputVar}[4], \\ & (s_0^0)_{a\$} \rightarrow \text{ParInputVar}[5], (s_0^0)_r \rightarrow \text{ParInputVar}[6], (s_0^{AS})_r \rightarrow \text{ParInputVar}[7], \\ & (s_R^0)_{a\$} \rightarrow \text{ParInputVar}[8], (p_0^{AS})_R \rightarrow \text{ParInputVar}[9], (p_s^{AS})_R \rightarrow \text{ParInputVar}[10], \\ & (p_0^{AS})_s \rightarrow \text{ParInputVar}[11], (p_R^{AS})_s \rightarrow \text{ParInputVar}[12], d_R \rightarrow \text{ParInputVar}[13], d_S \rightarrow \text{ParInputVar}[14], \\ & (r_0^0)_{-a\$} \rightarrow \text{ParInputVar}[15], (r_0^0)_{-s} \rightarrow \text{ParInputVar}[16], (r_0^{AS})_{-s} \rightarrow \text{ParInputVar}[17], \\ & (r_s^0)_{-a\$} \rightarrow \text{ParInputVar}[18], (s_0^0)_{-a\$} \rightarrow \text{ParInputVar}[19], (s_0^0)_{-r} \rightarrow \text{ParInputVar}[20], \\ & (s_0^{AS})_{-r} \rightarrow \text{ParInputVar}[21], (s_R^0)_{-a\$} \rightarrow \text{ParInputVar}[22], A\$ \rightarrow \text{ParInputVar}[23]\} \end{aligned}$$

Time evolution equations

$$\begin{aligned} R'[t] &= -R[t] d_R + (p_0^{AS})_R R_0^{AS}[t] + (p_s^{AS})_R R_S^{AS}[t] - R[t] (s_0^0)_r S_0^0[t] - R[t] (s_0^{AS})_r S_0^{AS}[t] + (s_0^0)_{-r} S_R^0[t] + (s_0^{AS})_{-r} S_R^{AS}[t] \\ S'[t] &= -S[t] d_S - S[t] (r_0^0)_s R_0^0[t] - S[t] (r_0^{AS})_s R_0^{AS}[t] + (r_0^0)_{-s} R_S^0[t] + (r_0^{AS})_{-s} R_S^{AS}[t] + (p_0^{AS})_s S_0^{AS}[t] + (p_R^{AS})_s S_R^{AS}[t] \\ (G_1)'[t] &= R[t] d_R \\ (G_2)'[t] &= S[t] d_S \\ (R_0^0)'[t] &= -A\$ (r_0^0)_{a\$} R_0^0[t] - S[t] (r_0^0)_s R_0^0[t] + (r_0^0)_{-a\$} R_0^{AS}[t] + (r_0^0)_{-s} R_S^0[t] \\ (R_0^{AS})'[t] &= A\$ (r_0^0)_{a\$} R_0^0[t] - (r_0^0)_{-a\$} R_0^{AS}[t] - S[t] (r_0^{AS})_s R_0^{AS}[t] + (r_0^{AS})_{-s} R_S^{AS}[t] \\ (R_S^0)'[t] &= S[t] (r_0^0)_s R_0^0[t] - (r_0^0)_{-s} R_S^0[t] - A\$ (r_s^0)_{a\$} R_S^0[t] + (r_s^0)_{-a\$} R_S^{AS}[t] \\ (R_S^{AS})'[t] &= S[t] (r_0^{AS})_s R_0^{AS}[t] + A\$ (r_s^0)_{a\$} R_S^0[t] - (r_0^{AS})_{-s} R_S^{AS}[t] - (r_s^0)_{-a\$} R_S^{AS}[t] \\ (S_0^0)'[t] &= -A\$ (s_0^0)_{a\$} S_0^0[t] - R[t] (s_0^0)_r S_0^0[t] + (s_0^0)_{-a\$} S_0^{AS}[t] + (s_0^0)_{-r} S_R^0[t] \\ (S_0^{AS})'[t] &= A\$ (s_0^0)_{a\$} S_0^0[t] - (s_0^0)_{-a\$} S_0^{AS}[t] - R[t] (s_0^{AS})_r S_0^{AS}[t] + (s_0^{AS})_{-r} S_R^{AS}[t] \\ (S_R^0)'[t] &= R[t] (s_0^0)_r S_0^0[t] - (s_0^0)_{-r} S_R^0[t] - A\$ (s_R^0)_{a\$} S_R^0[t] + (s_R^0)_{-a\$} S_R^{AS}[t] \\ (S_R^{AS})'[t] &= R[t] (s_0^{AS})_r S_0^{AS}[t] + A\$ (s_R^0)_{a\$} S_R^0[t] - (s_0^{AS})_{-r} S_R^{AS}[t] - (s_R^0)_{-a\$} S_R^{AS}[t] \end{aligned}$$

Conservation laws

$$\begin{aligned} & S_0^0[t] + S_0^{AS}[t] + S_R^0[t] + S_R^{AS}[t] \\ & R_0^0[t] + R_0^{AS}[t] + R_S^0[t] + R_S^{AS}[t] \\ & A\$ \end{aligned}$$


```

(** PARAMETER INPUT **)

(** Reactions Genes-Proteins **)
(* Forward *)
parametersFwValue = Table[1., {i, Length@ParametersFw}];
ParInputVar[1] = parametersFwValue[[1]];
ParInputVar[2] = parametersFwValue[[2]];
ParInputVar[3] = parametersFwValue[[3]];
ParInputVar[4] = parametersFwValue[[4]];
ParInputVar[5] = parametersFwValue[[5]];
ParInputVar[6] = parametersFwValue[[6]];
ParInputVar[7] = parametersFwValue[[7]];
ParInputVar[8] = parametersFwValue[[8]];

(** Productions **)
parametersProductionValue = Table[0., {i, Length@ParametersProduction}];
ParInputVar[9] = parametersProductionValue[[1]] = 1.;
ParInputVar[10] = parametersProductionValue[[2]] = 0.;
ParInputVar[11] = parametersProductionValue[[3]] = 2.3;
ParInputVar[12] = parametersProductionValue[[4]] = 0.;

(** Degradations **)
parametersDegradationValue = Table[1., {i, Length@ParametersDegradation}];
ParInputVar[13] = parametersDegradationValue[[1]];
ParInputVar[14] = parametersDegradationValue[[2]];

(* Backward *)
parametersBwValue = Table[0., {i, Length@ParametersBw}];
ParInputVar[15] = parametersBwValue[[1]];
ParInputVar[16] = parametersBwValue[[2]];
ParInputVar[17] = parametersBwValue[[3]];
ParInputVar[18] = parametersBwValue[[4]];
ParInputVar[19] = parametersBwValue[[5]];
ParInputVar[20] = parametersBwValue[[6]];
ParInputVar[21] = parametersBwValue[[7]];
ParInputVar[22] = parametersBwValue[[8]];

(** Constant Proteins **)
s1 = 0.1;
(* A *) ParInputVar[23] = 70.;

(** INITIAL CONDITIONS INPUT **)
(** Not Constant Proteins **)
RInitialDistribution[x_] := 70 - 70 x;
(* R *) InitCond[1] = RInitialDistribution[s1];
(* S *) InitCond[2] = 0.;
EmptyGenes[100.]
NotEmptyGenes[0.]

Print["Initial conditions"];
Print[initialConditions];
Print["\n\nParameters"];
Print[parametersInput];

```

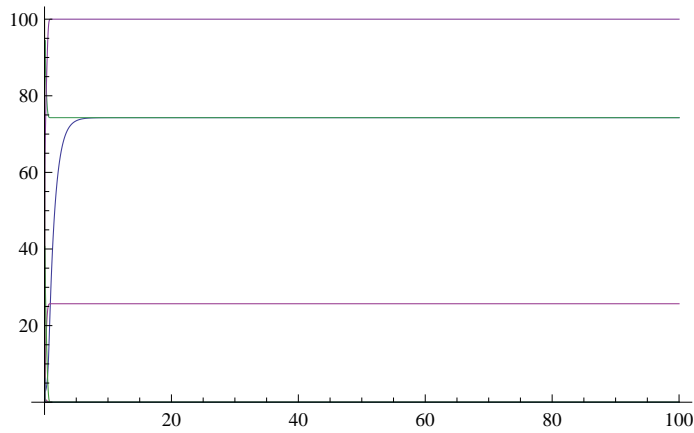
Initial conditions

$$\{R[0] = 63., S[0] = 0., R_0^0[0] = 100., R_0^{AS}[0] = 0., R_S^0[0] = 0., \\ R_S^{AS}[0] = 0., S_0^0[0] = 100., S_0^{AS}[0] = 0., S_R^0[0] = 0., S_R^{AS}[0] = 0.\}$$

Parameters

$$\{(r_0^0)_{a\$} \rightarrow 1., (r_0^0)_s \rightarrow 1., (r_0^{AS})_s \rightarrow 1., (r_s^0)_{a\$} \rightarrow 1., (s_0^0)_{a\$} \rightarrow 1., (s_0^0)_r \rightarrow 1., (s_0^{AS})_r \rightarrow 1., (s_R^0)_{a\$} \rightarrow 1., \\ (p_0^{AS})_R \rightarrow 1., (p_s^{AS})_R \rightarrow 0., (p_0^{AS})_s \rightarrow 2.3, (p_R^{AS})_s \rightarrow 0., d_R \rightarrow 1., d_s \rightarrow 1., (r_0^0)_{-a\$} \rightarrow 0., (r_0^0)_{-s} \rightarrow 0., \\ (r_0^{AS})_{-s} \rightarrow 0., (r_s^0)_{-a\$} \rightarrow 0., (s_0^0)_{-a\$} \rightarrow 0., (s_0^0)_{-r} \rightarrow 0., (s_0^{AS})_{-r} \rightarrow 0., (s_R^0)_{-a\$} \rightarrow 0., AS \rightarrow 70.\}$$

```
Time = 100;
equations = Drop[equations, {3, 4}];
sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],
  substanceNames, {t, 0, Time}];
g1 = Plot[Evaluate[substanceVariables /. sol], {t, 0, Time}, PlotRange -> All]
```



```
(* Threshold effect *)

APropConst = 1;
RPropConst = 1;
SPropConst = 0.5;
Time = 100;
table1 = {}; table2 = {}; table3 = {}; table4 = {};
For[s1 = 0, s1 <= 1, s1 += 0.01,

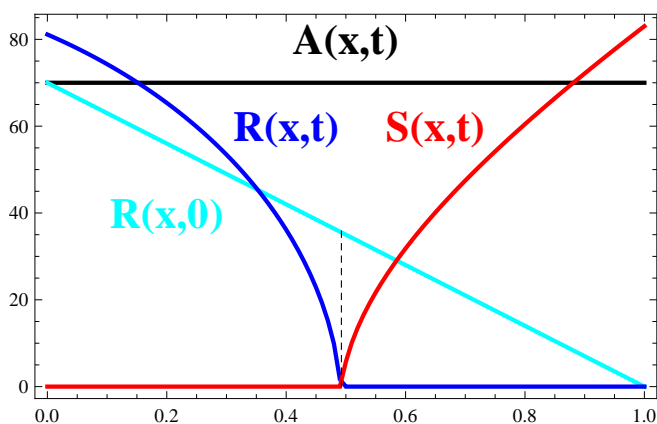
  (* BCD *) ParInputVar[23] = 70.;
  (* HB *) InitCond[1] = RInitialDistribution[s1];

  sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],
    substanceNames, {t, 0, Time}];

  table1 = Append[table1, {s1, APropConst ParInputVar[23]}];
  table2 = Append[table2, {s1, RPropConst InitCond[1]}];
  table3 = Append[table3, {s1, RPropConst Evaluate[R[t] /. sol] /. t -> Time}];
  table4 = Append[table4, {s1, SPropConst Evaluate[S[t] /. sol] /. t -> Time}];
];
Agraph =
  ListPlot[table1, Joined -> True, PlotStyle -> {Black, Thick}, PlotRange -> All];
Ringraph = ListPlot[table2, Joined -> True,
  PlotStyle -> {Cyan, Thick}, PlotRange -> All];
Rgraph = ListPlot[table3, Joined -> True, PlotStyle -> {Blue, Thick}, PlotRange -> All];
Sgraph = ListPlot[table4, Joined -> True, PlotStyle -> {Red, Thick}, PlotRange -> All];

dash = Graphics[{Dashing[{0.01}], Line[{0.4925, 0}, {0.4925, 36}]}];

threshold = Show[Agraph, Ringraph, Rgraph, Sgraph, dash,
  Graphics[Text["A(x,t)", {0.5, 80}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Black}]],
  Graphics[Text["R(x,t)", {0.4, 60}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Blue}]],
  Graphics[Text["S(x,t)", {0.65, 60}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Red}]],
  Graphics[Text["R(x,0)", {0.2, 40}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Cyan}]],
  Graphics[Text["R(x,0)", {0.2, 40}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 20, FontWeight -> "Bold", Cyan}]],
  Frame -> True]
```



```
(* Quit Kernel *)
```

```
<< GeneticNetworksV280709/GeneticNetworks.m
```

Network 5: Fit of Hunchback and Knirps

```
Needs["ErrorBarPlots`"]

BCDCompleteData = ReadList["C:\\Users\\Daniele
Muraro\\Documents\\GeneticNetworksV280709\\input\\gap_data\\bcd.txt",
{Number, Real, Real, Real}];

BCDx = BCDCompleteData[[All, 2]];
BCDmean = BCDCompleteData[[All, 3]];
BCDstddev = BCDCompleteData[[All, 4]];

BCDgraph = ErrorListPlot[
Table[{BCDx[[i]], BCDmean[[i]], ErrorBar[BCDstddev[[i]]}], {i, Length@BCDx}],
PlotStyle -> {PointSize[0.01], RGBColor[0, 0, 0]}, PlotRange -> All,
Joined -> False];

CADCompleteData = ReadList["C:\\Users\\Daniele
Muraro\\Documents\\Manual\\hb_kni_tllexp\\input\\gap_data\\cad.txt",
{Number, Real, Real, Real}];

CADx = CADCompleteData[[All, 2]];
CADmean = CADCompleteData[[All, 3]];
CADstddev = CADCompleteData[[All, 4]];

CADgraph = ErrorListPlot[
Table[{CADx[[i]], CADmean[[i]], ErrorBar[CADstddev[[i]]}], {i, Length@CADx}],
PlotStyle -> {PointSize[0.01], RGBColor[0, 0, 0]}, PlotRange -> All,
Joined -> False];

BCDCADFit = ReadList["C:\\Users\\Daniele
Muraro\\Documents\\GeneticNetworksV280709\\input\\initial_distributions\\bcd_cad
_fit.txt", {Real, Real, Real}];

BCDFit = BCDCADFit[[All, 1 ;; 2 ;; 1]];
CADFit = BCDCADFit[[All, 1 ;; 3 ;; 2]];

BCDFitgraph = ListPlot[BCDFit, Joined -> True, PlotStyle -> {Purple, Thick}];
CADFitgraph = ListPlot[CADFit, Joined -> True, PlotStyle -> {Green, Thick}];

HB13CycleCompleteData = ReadList["C:\\Users\\Daniele
Muraro\\Documents\\GeneticNetworksV280709\\input\\gap_data\\hb_13.txt",
{Number, Number, Real, Real, Real}];
```

```

HB13Cyclex = HB13CycleCompleteData[[All, 3]];
HB13Cyclemean = HB13CycleCompleteData[[All, 4]];
HB13Cyclestdev = HB13CycleCompleteData[[All, 5]];

HB13Cyclegraph = ErrorListPlot[
  Table[{{HB13Cyclex[[i]], HB13Cyclemean[[i]]}, ErrorBar[HB13Cyclestdev[[i]]]},
    {i, Length@HB13Cyclex}], PlotStyle -> {PointSize[0.01], RGBColor[0, 0, 0]},
  PlotRange -> All, Joined -> False];

HB13CycleFit = ReadList["C:\\Users\\Daniele
  Muraro\\Documents\\GeneticNetworksV280709\\input\\initial_distributions\\hb_fit.
  txt", {Real, Real}];

HB13CycleFitgraph = ListPlot[HB13CycleFit, Joined -> True, PlotStyle -> {Blue, Thick}];

TLLCompleteData = ReadList["C:\\Users\\Daniele
  Muraro\\Documents\\GeneticNetworksV280709\\input\\gap_data\\tll.txt",
  {Number, Number, Real, Real, Real}];

TLLx = TLLCompleteData[[All, 3]];
TLLmean = TLLCompleteData[[All, 4]];
TLLstdev = TLLCompleteData[[All, 5]];

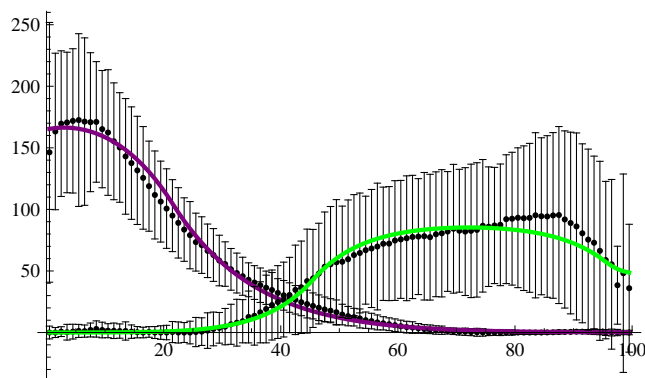
TLLgraph = ErrorListPlot[
  Table[{{TLLx[[i]], TLLmean[[i]]}, ErrorBar[TLLstdev[[i]]]}, {i, Length@TLLx}],
  PlotStyle -> {PointSize[0.01], Black}, PlotRange -> All, Joined -> False];

TLLFit = ReadList["C:\\Users\\Daniele
  Muraro\\Documents\\GeneticNetworksV280709\\input\\initial_distributions\\tll_fit
  .txt", {Real, Real}];

TLLFitgraph = ListPlot[TLLFit, Joined -> True, PlotStyle -> {Brown, Thick}];

Show[BCDgraph, BCDFitgraph, CADgraph, CADFitgraph]

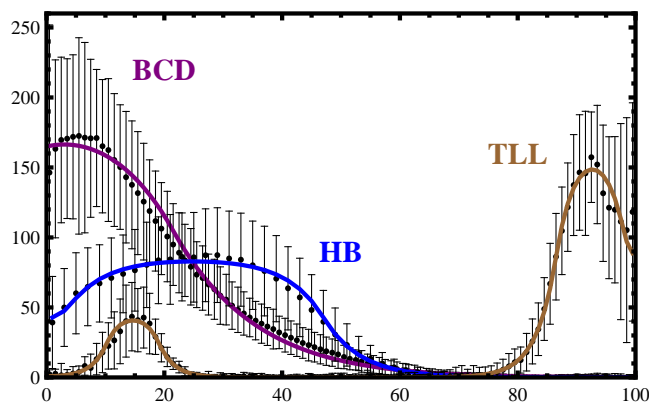
```



```

gInit = Show[BCDgraph, BCDfitgraph,
  HB13Cyclegraph, HB13CycleFitgraph, TLLgraph, TLLfitgraph,
  Graphics[Text["HB", {50, 90}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 14, FontWeight -> "Bold", Blue}]],
  Graphics[Text["TLL", {80, 160}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 14, FontWeight -> "Bold", Brown}]],
  Graphics[Text["BCD", {20, 220}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 14, FontWeight -> "Bold", Purple}]],
  PlotRange -> {{0, 100}, {0, 260}},
  Frame -> True, FrameStyle -> Thick]

```



```
(* Read Data *)
```

```

test = OpenRead[
  "C:\\Users\\Daniele Muraro\\Documents\\GeneticNetworksV280709\\input\\best_found.txt"]

InputStream[
  C:\\Users\\Daniele Muraro\\Documents\\GeneticNetworksV280709\\input\\best_found.txt, 51]

data = {};
text = {};
temp = 0;
While[! MatchQ[temp, EndOfFile], temp = Read[test, Word];
  temp2 = ToExpression[temp];
  If[MatchQ[temp2, _Integer] || MatchQ[temp2, _Real], AppendTo[data, temp2]];
  If[! MatchQ[temp2, _Integer] && ! MatchQ[temp2, _Real], AppendTo[text, temp2]];]

```

```

LengthVariables = data[[1]];
LengthParametersRNA = data[[2]];
LengthParametersEmptyGenes = data[[3]];
LengthParametersFw = data[[4]];
LengthParametersProduction = data[[5]];
LengthParametersDegradation = data[[6]];
LengthParametersBw = data[[7]]; LengthProportionalityConstants = data[[8]];

LengthL = 0;
a = Table[0, {i, LengthL}];
start = 8; (*Vector of Lengths*)
Do[a[[i]] = data[[i + start]], {i, LengthL}];
L = Table[0, {i, LengthL}];
start += LengthL;
Do[L[[i]] = data[[i + start]], {i, LengthL}];

emptyGenesValue = Table[0, {i, LengthParametersEmptyGenes}];
start += LengthL;
Do[emptyGenesValue[[i]] = data[[i + start]], {i, LengthParametersEmptyGenes}];

emptyGenesPositionData = Table[0, {i, LengthParametersEmptyGenes}];
start += LengthParametersEmptyGenes;
Do[emptyGenesPositionData[[i]] = data[[i + start]], {i, LengthParametersEmptyGenes}];

parametersFwValue = Table[0., {i, LengthParametersFw}];
start += LengthParametersEmptyGenes;
Do[parametersFwValue[[i]] = data[[i + start]], {i, LengthParametersFw}];

parametersProductionValue = Table[1, {i, LengthParametersProduction}];
start += LengthParametersFw;
Do[parametersProductionValue[[i]] = data[[i + start]],
  {i, LengthParametersProduction}];

parametersDegradationValue = Table[1., {i, LengthParametersDegradation}];
start += LengthParametersProduction;
Do[parametersDegradationValue[[i]] = data[[i + start]],
  {i, LengthParametersDegradation}];

parametersBwValue = Table[0., {i, LengthParametersBw}];
start += LengthParametersDegradation;
Do[parametersBwValue[[i]] = data[[i + start]], {i, LengthParametersBw}];

parametersPropConst = Table[0., {i, LengthProportionalityConstants}];
start += LengthParametersBw;
Do[parametersPropConst[[i]] = data[[i + start]], {i, LengthProportionalityConstants}];

start += LengthProportionalityConstants;
Time = data[[1 + start]];

```

```
a
L
emptyGenesValue
emptyGenesPositionData
parametersFwValue
parametersProductionValue
parametersDegradationValue
parametersBwValue
parametersPropConst
Time

{}

{}

{282.042, 299.528}

{3, 7}

{0.911508, 1.99037, 0.694221, 1.13994, 1.99477}

{15.1128, 0., 16.5423, 0.024902, 16.1644, 0.021352}

{0.900265, 0.955021}

{0.199906, 0.050465, 0.050144, 0.053843, 0.118673}

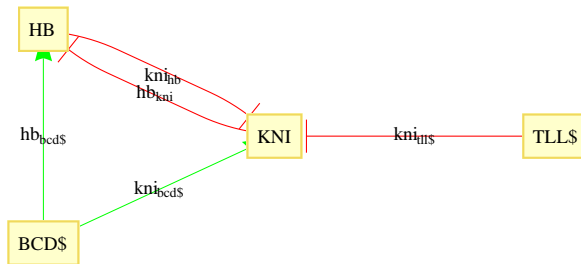
{0.117246, 1.91264, 0.135616, 1.2691}

7.7962
```



```
(* HB-KNI Network *)
```

```
activations = {BCD$ -> HB, BCD$ -> KNI};
repressions = {HB -> KNI, KNI -> HB, TLL$ -> KNI};
NetworkGraph[activations, repressions]
```



```
Reactions[activations, repressions];
```

```
Print["\n\nGraph of reactions"];
ReactionGraph[reactions, parameters];
```

```
Print["\n\nSubstances"];
names = SubstanceNames[reactions];
variables = SubstanceVariables[reactions];
initialConditions = SubstanceInitialConditions[reactions];
Print[names]; Print[variables]; Print[initialConditions];
```

```
Print["\n\nParameters"];
parametersInput = ParameterInput[parameters];
Print[parametersInput];
```

```
Print["\n\nTime evolution equations"];
eqs = Equations[reactions, parameters];
Print[eqs // Column]
```

```
Print["\n\nConservation laws"];
cons = ConservationLaws[reactions];
Print[cons // Column];
```

Proteins = {BCD\$, HB, KNI, TLL\$}

Activation Sites per Gene = {0, 1, 1, 0}

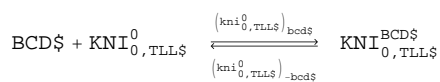
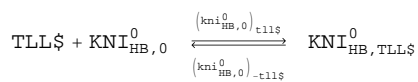
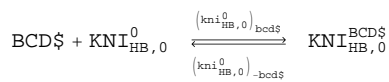
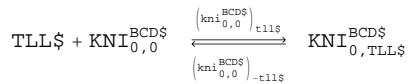
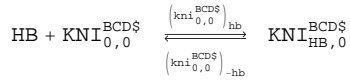
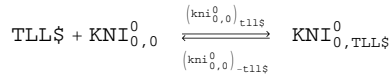
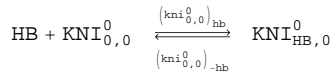
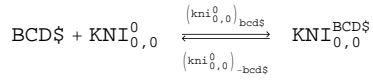
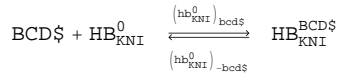
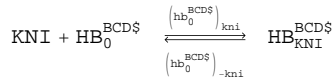
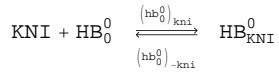
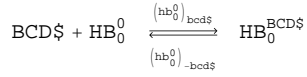
Repression Sites per Gene = {0, 1, 2, 0}

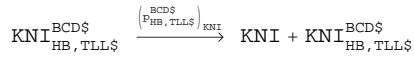
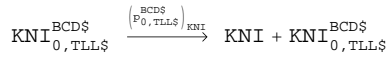
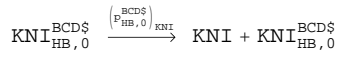
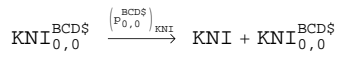
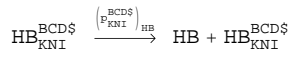
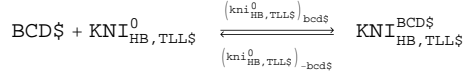
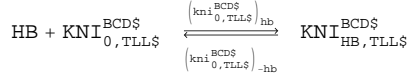
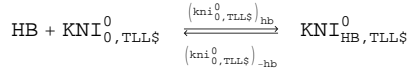
Total Sites per Gene = {0, 2, 3, 0}

Proteins Without Genes = {BCD\$, TLL\$}

Constant Proteins = {BCD\$, TLL\$}

Graph of reactions





Substances

$$\left\{ \begin{aligned} & \text{HB}, \text{KNI}, \text{HB}_0^0, \text{HB}_0^{\text{BCD\$}}, \text{HB}_{\text{KNI}}^0, \text{HB}_{\text{KNI}}^{\text{BCD\$}}, \text{KNI}_{0,0}^0, \text{KNI}_{0,0}^{\text{BCD\$}}, \\ & \text{KNI}_{0,\text{TLL\$}}^0, \text{KNI}_{0,\text{TLL\$}}^{\text{BCD\$}}, \text{KNI}_{\text{HB},0}^0, \text{KNI}_{\text{HB},0}^{\text{BCD\$}}, \text{KNI}_{\text{HB},\text{TLL\$}}^0, \text{KNI}_{\text{HB},\text{TLL\$}}^{\text{BCD\$}} \end{aligned} \right\}$$

$$\left\{ \begin{aligned} & \text{HB}[\text{t}], \text{KNI}[\text{t}], \text{HB}_0^0[\text{t}], \text{HB}_0^{\text{BCD\$}}[\text{t}], \text{HB}_{\text{KNI}}^0[\text{t}], \text{HB}_{\text{KNI}}^{\text{BCD\$}}[\text{t}], \text{KNI}_{0,0}^0[\text{t}], \text{KNI}_{0,0}^{\text{BCD\$}}[\text{t}], \\ & \text{KNI}_{0,\text{TLL\$}}^0[\text{t}], \text{KNI}_{0,\text{TLL\$}}^{\text{BCD\$}}[\text{t}], \text{KNI}_{\text{HB},0}^0[\text{t}], \text{KNI}_{\text{HB},0}^{\text{BCD\$}}[\text{t}], \text{KNI}_{\text{HB},\text{TLL\$}}^0[\text{t}], \text{KNI}_{\text{HB},\text{TLL\$}}^{\text{BCD\$}}[\text{t}] \end{aligned} \right\}$$

$$\left\{ \begin{aligned} & \text{HB}[0] = \text{InitCond}[1], \text{KNI}[0] = \text{InitCond}[2], \text{HB}_0^0[0] = \text{InitCond}[3], \text{HB}_0^{\text{BCD\$}}[0] = \text{InitCond}[4], \\ & \text{HB}_{\text{KNI}}^0[0] = \text{InitCond}[5], \text{HB}_{\text{KNI}}^{\text{BCD\$}}[0] = \text{InitCond}[6], \text{KNI}_{0,0}^0[0] = \text{InitCond}[7], \text{KNI}_{0,0}^{\text{BCD\$}}[0] = \text{InitCond}[8], \\ & \text{KNI}_{0,\text{TLL\$}}^0[0] = \text{InitCond}[9], \text{KNI}_{0,\text{TLL\$}}^{\text{BCD\$}}[0] = \text{InitCond}[10], \text{KNI}_{\text{HB},0}^0[0] = \text{InitCond}[11], \\ & \text{KNI}_{\text{HB},0}^{\text{BCD\$}}[0] = \text{InitCond}[12], \text{KNI}_{\text{HB},\text{TLL\$}}^0[0] = \text{InitCond}[13], \text{KNI}_{\text{HB},\text{TLL\$}}^{\text{BCD\$}}[0] = \text{InitCond}[14] \end{aligned} \right\}$$

Parameters

$$\left\{ \begin{aligned} & (\text{hb}_0^0)_{\text{bcd\$}} \rightarrow \text{ParInputVar}[1], (\text{hb}_0^0)_{\text{kni}} \rightarrow \text{ParInputVar}[2], (\text{hb}_0^{\text{BCD\$}})_{\text{kni}} \rightarrow \text{ParInputVar}[3], \\ & (\text{hb}_{\text{KNI}}^0)_{\text{bcd\$}} \rightarrow \text{ParInputVar}[4], (\text{kni}_{0,0}^0)_{\text{bcd\$}} \rightarrow \text{ParInputVar}[5], (\text{kni}_{0,0}^0)_{\text{hb}} \rightarrow \text{ParInputVar}[6], \\ & (\text{kni}_{0,0}^0)_{\text{tll\$}} \rightarrow \text{ParInputVar}[7], (\text{kni}_{0,0}^{\text{BCD\$}})_{\text{hb}} \rightarrow \text{ParInputVar}[8], (\text{kni}_{0,0}^{\text{BCD\$}})_{\text{tll\$}} \rightarrow \text{ParInputVar}[9], \\ & (\text{kni}_{\text{HB},0}^0)_{\text{bcd\$}} \rightarrow \text{ParInputVar}[10], (\text{kni}_{\text{HB},0}^0)_{\text{tll\$}} \rightarrow \text{ParInputVar}[11], (\text{kni}_{0,\text{TLL\$}}^0)_{\text{bcd\$}} \rightarrow \text{ParInputVar}[12], \\ & (\text{kni}_{0,\text{TLL\$}}^0)_{\text{hb}} \rightarrow \text{ParInputVar}[13], (\text{kni}_{\text{HB},0}^{\text{BCD\$}})_{\text{tll\$}} \rightarrow \text{ParInputVar}[14], (\text{kni}_{0,\text{TLL\$}}^{\text{BCD\$}})_{\text{hb}} \rightarrow \text{ParInputVar}[15], \\ & (\text{kni}_{\text{HB},\text{TLL\$}}^0)_{\text{bcd\$}} \rightarrow \text{ParInputVar}[16], (\text{p}_0^{\text{BCD\$}})_{\text{HB}} \rightarrow \text{ParInputVar}[17], (\text{p}_{\text{KNI}}^{\text{BCD\$}})_{\text{HB}} \rightarrow \text{ParInputVar}[18], \\ & (\text{p}_{0,0}^{\text{BCD\$}})_{\text{KNI}} \rightarrow \text{ParInputVar}[19], (\text{p}_{\text{HB},0}^{\text{BCD\$}})_{\text{KNI}} \rightarrow \text{ParInputVar}[20], (\text{p}_{0,\text{TLL\$}}^{\text{BCD\$}})_{\text{KNI}} \rightarrow \text{ParInputVar}[21], \\ & (\text{p}_{\text{HB},\text{TLL\$}}^{\text{BCD\$}})_{\text{KNI}} \rightarrow \text{ParInputVar}[22], \text{d}_{\text{HB}} \rightarrow \text{ParInputVar}[23], \text{d}_{\text{KNI}} \rightarrow \text{ParInputVar}[24], \\ & (\text{hb}_0^0)_{\text{-bcd\$}} \rightarrow \text{ParInputVar}[25], (\text{hb}_0^0)_{\text{-kni}} \rightarrow \text{ParInputVar}[26], (\text{hb}_0^{\text{BCD\$}})_{\text{-kni}} \rightarrow \text{ParInputVar}[27], \\ & (\text{hb}_{\text{KNI}}^0)_{\text{-bcd\$}} \rightarrow \text{ParInputVar}[28], (\text{kni}_{0,0}^0)_{\text{-bcd\$}} \rightarrow \text{ParInputVar}[29], (\text{kni}_{0,0}^0)_{\text{-hb}} \rightarrow \text{ParInputVar}[30], \\ & (\text{kni}_{0,0}^0)_{\text{-tll\$}} \rightarrow \text{ParInputVar}[31], (\text{kni}_{0,0}^{\text{BCD\$}})_{\text{-hb}} \rightarrow \text{ParInputVar}[32], (\text{kni}_{0,0}^{\text{BCD\$}})_{\text{-tll\$}} \rightarrow \text{ParInputVar}[33], \\ & (\text{kni}_{\text{HB},0}^0)_{\text{-bcd\$}} \rightarrow \text{ParInputVar}[34], (\text{kni}_{\text{HB},0}^0)_{\text{-tll\$}} \rightarrow \text{ParInputVar}[35], (\text{kni}_{0,\text{TLL\$}}^0)_{\text{-bcd\$}} \rightarrow \text{ParInputVar}[36], \\ & (\text{kni}_{0,\text{TLL\$}}^0)_{\text{-hb}} \rightarrow \text{ParInputVar}[37], (\text{kni}_{\text{HB},0}^{\text{BCD\$}})_{\text{-tll\$}} \rightarrow \text{ParInputVar}[38], (\text{kni}_{0,\text{TLL\$}}^{\text{BCD\$}})_{\text{-hb}} \rightarrow \text{ParInputVar}[39], \\ & (\text{kni}_{\text{HB},\text{TLL\$}}^0)_{\text{-bcd\$}} \rightarrow \text{ParInputVar}[40], \text{BCD\$} \rightarrow \text{ParInputVar}[41], \text{TLL\$} \rightarrow \text{ParInputVar}[42] \end{aligned} \right\}$$

Time evolution equations

$$\begin{aligned}
HB'[t] &= -HB[t] d_{HB} + \left(p_{0,0}^{BCD\$}\right)_{HB} HB_0^{BCD\$}[t] + \left(p_{KNI}^{BCD\$}\right)_{HB} HB_{KNI}^{BCD\$}[t] - HB[t] (kni_{0,0}^0)_{hb} KNI_{0,0}^0[t] - \\
&HB[t] \left(kni_{0,0}^{BCD\$}\right)_{hb} KNI_{0,0}^{BCD\$}[t] - HB[t] (kni_{0,TLL\$}^0)_{hb} KNI_{0,TLL\$}^0[t] - HB[t] \left(kni_{0,TLL\$}^{BCD\$}\right)_{hb} KNI_{0,TLL\$}^{BCD\$}[t] + \\
&(kni_{0,0}^0)_{-hb} KNI_{HB,0}^0[t] + (kni_{0,0}^{BCD\$})_{-hb} KNI_{HB,0}^{BCD\$}[t] + (kni_{0,TLL\$}^0)_{-hb} KNI_{HB,TLL\$}^0[t] + (kni_{0,TLL\$}^{BCD\$})_{-hb} KNI_{HB,TLL\$}^{BCD\$}[t] \\
KNI'[t] &= -KNI[t] d_{KNI} - KNI[t] (hb_0^0)_{kni} HB_0^0[t] - KNI[t] (hb_0^{BCD\$})_{kni} HB_0^{BCD\$}[t] + \\
&(hb_0^0)_{-kni} HB_{KNI}^0[t] + (hb_0^{BCD\$})_{-kni} HB_{KNI}^{BCD\$}[t] + \left(p_{0,0}^{BCD\$}\right)_{KNI} KNI_{0,0}^{BCD\$}[t] + \\
&\left(p_{0,TLL\$}^{BCD\$}\right)_{KNI} KNI_{0,TLL\$}^{BCD\$}[t] + \left(p_{HB,0}^{BCD\$}\right)_{KNI} KNI_{HB,0}^{BCD\$}[t] + \left(p_{HB,TLL\$}^{BCD\$}\right)_{KNI} KNI_{HB,TLL\$}^{BCD\$}[t] \\
(G_1)'[t] &= HB[t] d_{HB} \\
(G_2)'[t] &= KNI[t] d_{KNI} \\
(HB_0^0)'[t] &= -BCD\$ (hb_0^0)_{bcd\$} HB_0^0[t] - KNI[t] (hb_0^0)_{kni} HB_0^0[t] + (hb_0^0)_{-bcd\$} HB_0^{BCD\$}[t] + (hb_0^0)_{-kni} HB_{KNI}^0[t] \\
(HB_0^{BCD\$})'[t] &= \\
&BCD\$ (hb_0^0)_{bcd\$} HB_0^0[t] - (hb_0^0)_{-bcd\$} HB_0^{BCD\$}[t] - KNI[t] (hb_0^{BCD\$})_{kni} HB_0^{BCD\$}[t] + (hb_0^{BCD\$})_{-kni} HB_{KNI}^{BCD\$}[t] \\
(HB_{KNI}^0)'[t] &= KNI[t] (hb_0^0)_{kni} HB_0^0[t] - (hb_0^0)_{-kni} HB_{KNI}^0[t] - BCD\$ (hb_{KNI}^0)_{bcd\$} HB_{KNI}^0[t] + (hb_{KNI}^0)_{-bcd\$} HB_{KNI}^{BCD\$}[t] \\
(HB_{KNI}^{BCD\$})'[t] &= \\
&KNI[t] (hb_0^{BCD\$})_{kni} HB_0^{BCD\$}[t] + BCD\$ (hb_{KNI}^0)_{bcd\$} HB_{KNI}^0[t] - (hb_0^{BCD\$})_{-kni} HB_{KNI}^{BCD\$}[t] - (hb_{KNI}^0)_{-bcd\$} HB_{KNI}^{BCD\$}[t] \\
(KNI_{0,0}^0)'[t] &= -BCD\$ (kni_{0,0}^0)_{bcd\$} KNI_{0,0}^0[t] - HB[t] (kni_{0,0}^0)_{hb} KNI_{0,0}^0[t] - \\
&TLL\$ (kni_{0,0}^0)_{tll\$} KNI_{0,0}^0[t] + (kni_{0,0}^0)_{-bcd\$} KNI_{0,0}^{BCD\$}[t] + (kni_{0,0}^0)_{-tll\$} KNI_{0,TLL\$}^0[t] + (kni_{0,0}^0)_{-hb} KNI_{HB,0}^0[t] \\
(KNI_{0,0}^{BCD\$})'[t] &= BCD\$ (kni_{0,0}^0)_{bcd\$} KNI_{0,0}^0[t] - (kni_{0,0}^0)_{-bcd\$} KNI_{0,0}^{BCD\$}[t] - HB[t] (kni_{0,0}^{BCD\$})_{hb} KNI_{0,0}^{BCD\$}[t] - \\
&TLL\$ (kni_{0,0}^{BCD\$})_{tll\$} KNI_{0,0}^{BCD\$}[t] + (kni_{0,0}^{BCD\$})_{-tll\$} KNI_{0,TLL\$}^{BCD\$}[t] + (kni_{0,0}^{BCD\$})_{-hb} KNI_{HB,0}^{BCD\$}[t] \\
(KNI_{0,TLL\$}^0)'[t] &= TLL\$ (kni_{0,0}^0)_{tll\$} KNI_{0,0}^0[t] - (kni_{0,0}^0)_{-tll\$} KNI_{0,TLL\$}^0[t] - BCD\$ (kni_{0,TLL\$}^0)_{bcd\$} KNI_{0,TLL\$}^0[t] - \\
&HB[t] (kni_{0,TLL\$}^0)_{hb} KNI_{0,TLL\$}^0[t] + (kni_{0,TLL\$}^0)_{-bcd\$} KNI_{0,TLL\$}^{BCD\$}[t] + (kni_{0,TLL\$}^0)_{-hb} KNI_{HB,TLL\$}^0[t] \\
(KNI_{0,TLL\$}^{BCD\$})'[t] &= TLL\$ (kni_{0,0}^{BCD\$})_{tll\$} KNI_{0,0}^{BCD\$}[t] + BCD\$ (kni_{0,TLL\$}^0)_{bcd\$} KNI_{0,TLL\$}^0[t] - (kni_{0,0}^{BCD\$})_{-tll\$} KNI_{0,TLL\$}^{BCD\$}[t] - \\
&(kni_{0,TLL\$}^0)_{-bcd\$} KNI_{0,TLL\$}^{BCD\$}[t] - HB[t] (kni_{0,TLL\$}^{BCD\$})_{hb} KNI_{0,TLL\$}^{BCD\$}[t] + (kni_{0,TLL\$}^{BCD\$})_{-hb} KNI_{HB,TLL\$}^{BCD\$}[t] \\
(KNI_{HB,0}^0)'[t] &= HB[t] (kni_{0,0}^0)_{hb} KNI_{0,0}^0[t] - (kni_{0,0}^0)_{-hb} KNI_{HB,0}^0[t] - BCD\$ (kni_{HB,0}^0)_{bcd\$} KNI_{HB,0}^0[t] - \\
&TLL\$ (kni_{HB,0}^0)_{tll\$} KNI_{HB,0}^0[t] + (kni_{HB,0}^0)_{-bcd\$} KNI_{HB,0}^{BCD\$}[t] + (kni_{HB,0}^0)_{-tll\$} KNI_{HB,TLL\$}^0[t] \\
(KNI_{HB,0}^{BCD\$})'[t] &= HB[t] (kni_{0,0}^{BCD\$})_{hb} KNI_{0,0}^{BCD\$}[t] + BCD\$ (kni_{HB,0}^0)_{bcd\$} KNI_{HB,0}^0[t] - (kni_{0,0}^{BCD\$})_{-hb} KNI_{HB,0}^{BCD\$}[t] - \\
&(kni_{HB,0}^0)_{-bcd\$} KNI_{HB,0}^{BCD\$}[t] - TLL\$ (kni_{HB,0}^{BCD\$})_{tll\$} KNI_{HB,0}^{BCD\$}[t] + (kni_{HB,0}^{BCD\$})_{-tll\$} KNI_{HB,TLL\$}^{BCD\$}[t] \\
(KNI_{HB,TLL\$}^0)'[t] &= \\
&HB[t] (kni_{0,TLL\$}^0)_{hb} KNI_{0,TLL\$}^0[t] + TLL\$ (kni_{HB,0}^0)_{tll\$} KNI_{HB,0}^0[t] - (kni_{0,TLL\$}^0)_{-hb} KNI_{HB,TLL\$}^0[t] - \\
&(kni_{HB,0}^0)_{-tll\$} KNI_{HB,TLL\$}^0[t] - BCD\$ (kni_{HB,TLL\$}^0)_{bcd\$} KNI_{HB,TLL\$}^0[t] + (kni_{HB,TLL\$}^0)_{-bcd\$} KNI_{HB,TLL\$}^{BCD\$}[t] \\
(KNI_{HB,TLL\$}^{BCD\$})'[t] &= \\
&HB[t] (kni_{0,TLL\$}^{BCD\$})_{hb} KNI_{0,TLL\$}^{BCD\$}[t] + TLL\$ (kni_{HB,0}^{BCD\$})_{tll\$} KNI_{HB,0}^{BCD\$}[t] + BCD\$ (kni_{HB,TLL\$}^0)_{bcd\$} KNI_{HB,TLL\$}^0[t] - \\
&(kni_{0,TLL\$}^{BCD\$})_{-hb} KNI_{HB,TLL\$}^{BCD\$}[t] - (kni_{HB,0}^{BCD\$})_{-tll\$} KNI_{HB,TLL\$}^{BCD\$}[t] - (kni_{HB,TLL\$}^0)_{-bcd\$} KNI_{HB,TLL\$}^{BCD\$}[t]
\end{aligned}$$

Conservation laws

$$\text{KNI}_{0,0}^0[t] + \text{KNI}_{0,0}^{\text{BCD}\$}[t] + \text{KNI}_{0,\text{TLL}\$}^0[t] + \text{KNI}_{0,\text{TLL}\$}^{\text{BCD}\$}[t] + \text{KNI}_{\text{HB},0}^0[t] + \text{KNI}_{\text{HB},0}^{\text{BCD}\$}[t] + \text{KNI}_{\text{HB},\text{TLL}\$}^0[t] + \text{KNI}_{\text{HB},\text{TLL}\$}^{\text{BCD}\$}[t]$$

$$\text{HB}_0^0[t] + \text{HB}_0^{\text{BCD}\$}[t] + \text{HB}_{\text{KNI}}^0[t] + \text{HB}_{\text{KNI}}^{\text{BCD}\$}[t]$$

$$\text{TLL}\$$$

$$\text{BCD}\$$$

(** PARAMETER INPUT **)

(** Reactions Genes-Proteins **)

(* Forward *)

(*parametersFwValue=Table[1.,{i,Length@parametersFwWithoutRepetitions}];*)

(* $(\text{hb}_0^0)_{\text{bcd}\$}$ *) ParInputVar[1] = parametersFwValue[[1]];

(* $(\text{hb}_{\text{KNI}}^0)_{\text{bcd}\$}$ *) ParInputVar[4] = parametersFwValue[[1]];

(* $(\text{hb}_0^0)_{\text{kni}}$ *) ParInputVar[2] = parametersFwValue[[2]];

(* $(\text{hb}_0^{\text{BCD}\$})_{\text{kni}}$ *) ParInputVar[3] = parametersFwValue[[2]];

(* $(\text{kni}_{0,0}^0)_{\text{bcd}\$}$ *) ParInputVar[5] = parametersFwValue[[3]];

(* $(\text{kni}_{\text{HB},0}^0)_{\text{bcd}\$}$ *) ParInputVar[10] = parametersFwValue[[3]];

(* $(\text{kni}_{0,\text{TLL}\$}^0)_{\text{bcd}\$}$ *) ParInputVar[12] = parametersFwValue[[3]];

(* $(\text{kni}_{\text{HB},\text{TLL}\$}^0)_{\text{bcd}\$}$ *) ParInputVar[16] = parametersFwValue[[3]];

(* $(\text{kni}_{0,0}^0)_{\text{hb}}$ *) ParInputVar[6] = parametersFwValue[[4]];

(* $(\text{kni}_{0,0}^{\text{BCD}\$})_{\text{hb}}$ *) ParInputVar[8] = parametersFwValue[[4]];

(* $(\text{kni}_{0,\text{TLL}\$}^0)_{\text{hb}}$ *) ParInputVar[13] = parametersFwValue[[4]];

(* $(\text{kni}_{0,\text{TLL}\$}^{\text{BCD}\$})_{\text{hb}}$ *) ParInputVar[15] = parametersFwValue[[4]];

(* $(\text{kni}_{0,0}^0)_{\text{tll}\$}$ *) ParInputVar[7] = parametersFwValue[[5]];

(* $(\text{kni}_{0,0}^{\text{BCD}\$})_{\text{tll}\$}$ *) ParInputVar[9] = parametersFwValue[[5]];

(* $(\text{kni}_{\text{HB},0}^0)_{\text{tll}\$}$ *) ParInputVar[11] = parametersFwValue[[5]];

(* $(\text{kni}_{\text{HB},0}^{\text{BCD}\$})_{\text{tll}\$}$ *) ParInputVar[14] = parametersFwValue[[5]];

(* Backward *)

(*parametersBwValue=Table[0.1,{i,Length@parametersBwWithoutRepetitions}];*)

(* $(\text{hb}_0^0)_{-\text{bcd}\$}$ *) ParInputVar[25] = parametersBwValue[[1]];

(* $(\text{hb}_{\text{KNI}}^0)_{-\text{bcd}\$}$ *) ParInputVar[28] = parametersBwValue[[1]];

```

(*  $(hb_0^0)_{-kni}$  *) ParInputVar[26] = parametersBwValue[[2]];
(*  $(hb_0^{BCD\$})_{-kni}$  *) ParInputVar[27] = parametersBwValue[[2]];

(*  $(kni_{0,0}^0)_{-bcd\$}$  *) ParInputVar[29] = parametersBwValue[[3]];
(*  $(kni_{HB,0}^0)_{-bcd\$}$  *) ParInputVar[34] = parametersBwValue[[3]];
(*  $(kni_{0,TLL\$}^0)_{-bcd\$}$  *) ParInputVar[36] = parametersBwValue[[3]];
(*  $(kni_{HB,TLL\$}^0)_{-bcd\$}$  *) ParInputVar[40] = parametersBwValue[[3]];

(*  $(kni_{0,0}^0)_{-hb}$  *) ParInputVar[30] = parametersBwValue[[4]];
(*  $(kni_{0,0}^{BCD\$})_{-hb}$  *) ParInputVar[32] = parametersBwValue[[4]];
(*  $(kni_{0,TLL\$}^0)_{-hb}$  *) ParInputVar[37] = parametersBwValue[[4]];
(*  $(kni_{0,TLL\$}^{BCD\$})_{-hb}$  *) ParInputVar[39] = parametersBwValue[[4]];

(*  $(kni_{0,0}^0)_{-tll\$}$  *) ParInputVar[31] = parametersBwValue[[5]];
(*  $(kni_{0,0}^{BCD\$})_{-tll\$}$  *) ParInputVar[33] = parametersBwValue[[5]];
(*  $(kni_{HB,0}^0)_{-tll\$}$  *) ParInputVar[35] = parametersBwValue[[5]];
(*  $(kni_{HB,0}^{BCD\$})_{-tll\$}$  *) ParInputVar[38] = parametersBwValue[[5]];

(** Productions **)
(*parametersProductionValue=
  Table[1,{i,Length@parametersProductionWithoutRepetitions}];*)
(*  $(P_0^{BCD\$})_{HB}$  *) ParInputVar[17] = parametersProductionValue[[1]];
(*  $(P_{KNI}^{BCD\$})_{HB}$  *) ParInputVar[18] = parametersProductionValue[[2]];
(*  $(P_{0,0}^{BCD\$})_{KNI}$  *) ParInputVar[19] = parametersProductionValue[[3]];
(*  $(P_{HB,0}^{BCD\$})_{KNI}$  *) ParInputVar[20] = parametersProductionValue[[4]];
(*  $(P_{0,TLL\$}^{BCD\$})_{KNI}$  *) ParInputVar[21] = parametersProductionValue[[5]];
(*  $(P_{HB,TLL\$}^{BCD\$})_{KNI}$  *) ParInputVar[22] = parametersProductionValue[[6]];

(** Degradations **)
(*parametersDegradationValue=
  Table[1.,{i,Length@parametersDegradationWithoutRepetitions}];*)
(*  $d_{HB}$  *) ParInputVar[23] = parametersDegradationValue[[1]];
(*  $d_{KNI}$  *) ParInputVar[24] = parametersDegradationValue[[2]];

```

```

(** Interpolation data **)
BCDInitial = Table[0, {i, 100}];
BCDInitial[[1]] = BCDFit[[1, 2]];
For[i = 2, i <= 100, i++, m = BCDFit[[i, 2]] - BCDFit[[i - 1, 2]];
BCDInitial[[i]] = BCDFit[[i - 1, 2]] + m 0.5]

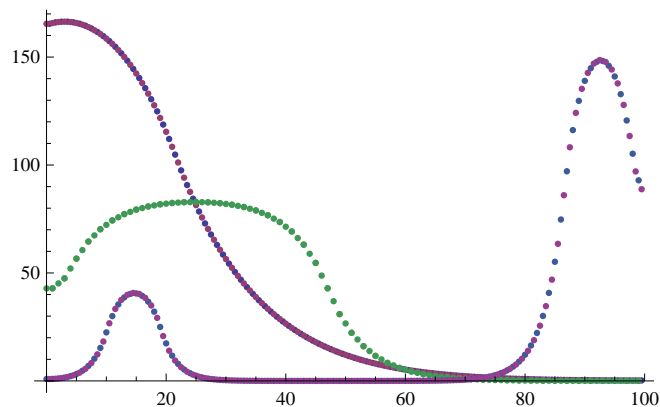
HB13CycleInitial = Table[0, {i, 100}];
HB13CycleInitial[[1]] = HB13CycleFit[[1, 2]];
HB13CycleInitial[[2]] = HB13CycleFit[[1, 2]];
i = 3;
For[j = 2, j <= 50, j++, m = (HB13CycleFit[[j, 2]] - HB13CycleFit[[j - 1, 2]]) / 2;
HB13CycleInitial[[i]] = HB13CycleFit[[j - 1, 2]] + m;
HB13CycleInitial[[i + 1]] = HB13CycleFit[[j - 1, 2]] + 2 m;
i += 2;]

TLLInitial = Table[0, {i, 100}];
TLLInitial[[1]] = TLLFit[[1, 2]];
For[i = 2, i <= 100, i++, m = TLLFit[[i, 2]] - TLLFit[[i - 1, 2]];
TLLInitial[[i]] = TLLFit[[i - 1, 2]] + m 0.5]

BCDInterpolationTest = Table[{i - 1, BCDInitial[[i]]}, {i, 1, 100}];
HBInterpolationTest = Table[{i - 1, HB13CycleInitial[[i]]}, {i, 1, 100}];
TLLInterpolationTest = Table[{i - 1, TLLInitial[[i]]}, {i, 1, 100}];

ListPlot[{BCDFit, BCDInterpolationTest,
HB13CycleFit, HBInterpolationTest, TLLInterpolationTest, TLLFit}]

```




```

(** Constant Proteins **)
i = 50;
(* BCD *) ParInputVar[41] = BCDInitial[[i]];
(* TLL *) ParInputVar[42] = TLLInitial[[i]];

(** INITIAL CONDITIONS INPUT **)
(** Not Constant Proteins **)
Do[InitCond[j] = 0., {j, 1, Length@substanceInitialConditions}]
(* HB *) InitCond[1] = HB13CycleInitial[[i]];

(** Empty Genes **)
(*emptyGenesValue=Table[100,{i,2}];*)
(* HB0[0] *) InitCond[3] = emptyGenesValue[[1]];
(* KNI00,0[0] *) InitCond[7] = emptyGenesValue[[2]];

Print["Initial conditions"];
Print[initialConditions];
Print["\n\nParameters"];
Print[parametersInput];

```

Initial conditions

$$\{ \text{HB}[0] = 30.9341, \text{KNI}[0] = 0., \text{HB}_0^0[0] = 282.042, \text{HB}_0^{\text{BCD}\$}[0] = 0., \\ \text{HB}_{\text{KNI}}^0[0] = 0., \text{HB}_{\text{KNI}}^{\text{BCD}\$}[0] = 0., \text{KNI}_{0,0}^0[0] = 299.528, \text{KNI}_{0,0}^{\text{BCD}\$}[0] = 0., \text{KNI}_{0,\text{TLL}\$}^0[0] = 0., \\ \text{KNI}_{0,\text{TLL}\$}^{\text{BCD}\$}[0] = 0., \text{KNI}_{\text{HB},0}^0[0] = 0., \text{KNI}_{\text{HB},0}^{\text{BCD}\$}[0] = 0., \text{KNI}_{\text{HB},\text{TLL}\$}^0[0] = 0., \text{KNI}_{\text{HB},\text{TLL}\$}^{\text{BCD}\$}[0] = 0. \}$$

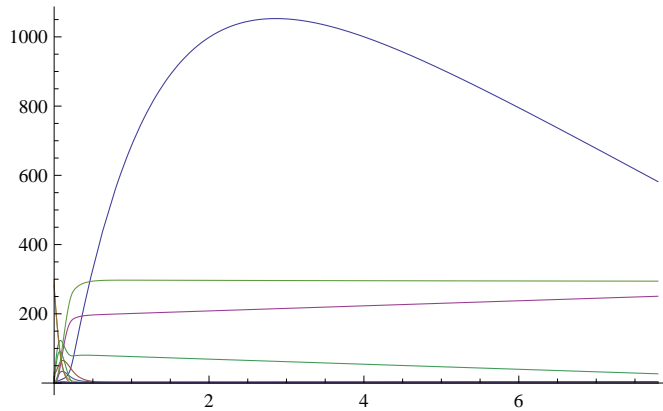
Parameters

$$\{ (\text{hb}_0^0)_{\text{bcd}\$} \rightarrow 0.911508, (\text{hb}_0^0)_{\text{kni}} \rightarrow 1.99037, (\text{hb}_0^{\text{BCD}\$})_{\text{kni}} \rightarrow 1.99037, (\text{hb}_{\text{KNI}}^0)_{\text{bcd}\$} \rightarrow 0.911508, \\ (\text{kni}_{0,0}^0)_{\text{bcd}\$} \rightarrow 0.694221, (\text{kni}_{0,0}^0)_{\text{hb}} \rightarrow 1.13994, (\text{kni}_{0,0}^0)_{\text{tll}\$} \rightarrow 1.99477, (\text{kni}_{0,0}^{\text{BCD}\$})_{\text{hb}} \rightarrow 1.13994, \\ (\text{kni}_{0,0}^{\text{BCD}\$})_{\text{tll}\$} \rightarrow 1.99477, (\text{kni}_{\text{HB},0}^0)_{\text{bcd}\$} \rightarrow 0.694221, (\text{kni}_{\text{HB},0}^0)_{\text{tll}\$} \rightarrow 1.99477, (\text{kni}_{0,\text{TLL}\$}^0)_{\text{bcd}\$} \rightarrow 0.694221, \\ (\text{kni}_{0,\text{TLL}\$}^0)_{\text{hb}} \rightarrow 1.13994, (\text{kni}_{\text{HB},0}^{\text{BCD}\$})_{\text{tll}\$} \rightarrow 1.99477, (\text{kni}_{0,\text{TLL}\$}^{\text{BCD}\$})_{\text{hb}} \rightarrow 1.13994, (\text{kni}_{\text{HB},\text{TLL}\$}^0)_{\text{bcd}\$} \rightarrow 0.694221, \\ (\text{p}_0^{\text{BCD}\$})_{\text{HB}} \rightarrow 15.1128, (\text{p}_{\text{KNI}}^{\text{BCD}\$})_{\text{HB}} \rightarrow 0., (\text{p}_{0,0}^{\text{BCD}\$})_{\text{KNI}} \rightarrow 16.5423, (\text{p}_{\text{HB},0}^{\text{BCD}\$})_{\text{KNI}} \rightarrow 0.024902, \\ (\text{p}_{0,\text{TLL}\$}^{\text{BCD}\$})_{\text{KNI}} \rightarrow 16.1644, (\text{p}_{\text{HB},\text{TLL}\$}^{\text{BCD}\$})_{\text{KNI}} \rightarrow 0.021352, d_{\text{HB}} \rightarrow 0.900265, d_{\text{KNI}} \rightarrow 0.955021, \\ (\text{hb}_0^0)_{-\text{bcd}\$} \rightarrow 0.199906, (\text{hb}_0^0)_{-\text{kni}} \rightarrow 0.050465, (\text{hb}_0^{\text{BCD}\$})_{-\text{kni}} \rightarrow 0.050465, (\text{hb}_{\text{KNI}}^0)_{-\text{bcd}\$} \rightarrow 0.199906, \\ (\text{kni}_{0,0}^0)_{-\text{bcd}\$} \rightarrow 0.050144, (\text{kni}_{0,0}^0)_{-\text{hb}} \rightarrow 0.053843, (\text{kni}_{0,0}^0)_{-\text{tll}\$} \rightarrow 0.118673, (\text{kni}_{0,0}^{\text{BCD}\$})_{-\text{hb}} \rightarrow 0.053843, \\ (\text{kni}_{0,0}^{\text{BCD}\$})_{-\text{tll}\$} \rightarrow 0.118673, (\text{kni}_{\text{HB},0}^0)_{-\text{bcd}\$} \rightarrow 0.050144, (\text{kni}_{\text{HB},0}^0)_{-\text{tll}\$} \rightarrow 0.118673, \\ (\text{kni}_{0,\text{TLL}\$}^0)_{-\text{bcd}\$} \rightarrow 0.050144, (\text{kni}_{0,\text{TLL}\$}^0)_{-\text{hb}} \rightarrow 0.053843, (\text{kni}_{\text{HB},0}^{\text{BCD}\$})_{-\text{tll}\$} \rightarrow 0.118673, \\ (\text{kni}_{0,\text{TLL}\$}^{\text{BCD}\$})_{-\text{hb}} \rightarrow 0.053843, (\text{kni}_{\text{HB},\text{TLL}\$}^0)_{-\text{bcd}\$} \rightarrow 0.050144, \text{BCD}\$ \rightarrow 13.0709, \text{TLL}\$ \rightarrow 0.00118 \}$$

```

equations = Drop[equations, {3, 4}];
sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],
  substanceNames, {t, 0, Time}];
g1 = Plot[Evaluate[substanceVariables /. sol], {t, 0, Time}, PlotRange -> All]

```



```

table1 = {}; table2 = {}; table3 = {}; table4 = {}; table5 = {};
For[i = 1, i <= 100, i++,

  (* BCD *) ParInputVar[41] = BCDInitial[[i]];
  (* TLL *) ParInputVar[42] = TLLInitial[[i]];

  (* HB *) InitCond[1] = HB13CycleInitial[[i]];

  sol = Flatten@NDSolve[Join[equations /. parametersInput, substanceInitialConditions],
    substanceNames, {t, 0, Time}];

  table1 = Append[table1, {i - 1, parametersPropConst[[1]] InitCond[1]}];
  table2 = Append[table2,
    {i - 1, parametersPropConst[[1]] Evaluate[HB[t] /. sol] /. t -> Time}];
  table3 = Append[table3, {i - 1, parametersPropConst[[2]]
    Evaluate[KNI[t] /. sol] /. t -> Time}];
  table4 = Append[table4, {i - 1, parametersPropConst[[3]] ParInputVar[41]}];
  table5 = Append[table5, {i - 1, parametersPropConst[[4]] ParInputVar[42]}];

];
(*gNDSolve = ListPlot[{table1, table2, table3, table4, table5}, PlotRange -> All];*)

HBFitgraph = ListPlot[table2, Joined -> True, PlotStyle -> {Blue, Thick}];
KNIFitgraph = ListPlot[table3, Joined -> True, PlotStyle -> {Red, Thick}];

HBCompleteData = ReadList["C:\\Users\\Daniele
  Muraro\\Documents\\GeneticNetworksV280709\\input\\gap_data\\hb14A-4-FRDWT.txt",

```

```

{Number, Number, Real, Real, Real}];

HBx = HBCompleteData[All, 3];
HBmean = HBCompleteData[All, 4];
HBstdev = HBCompleteData[All, 5];

HBgraph = ErrorListPlot[
  Table[{HBx[[i]], HBmean[[i]], ErrorBar[HBstdev[[i]]}], {i, Length@HBx}],
  PlotStyle -> {PointSize[0.01], Black}, PlotRange -> All, Joined -> False];

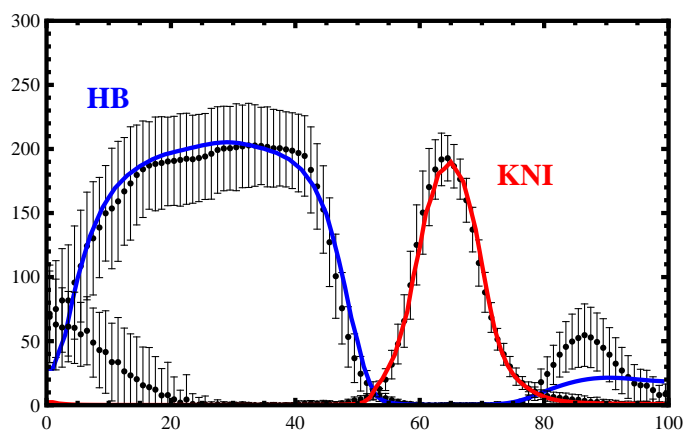
KNICompleteData = ReadList["C:\\Users\\Daniele
  Muraro\\Documents\\GeneticNetworksV280709\\input\\gap_data\\kni14A-4-FRDWT.txt",
  {Number, Number, Real, Real, Real}];

KNIx = KNICompleteData[All, 3];
KNImean = KNICompleteData[All, 4];
KNIstdev = KNICompleteData[All, 5];

KNIgraph = ErrorListPlot[
  Table[{KNIx[[i]], KNImean[[i]], ErrorBar[KNIstdev[[i]]}], {i, Length@KNIx}],
  PlotStyle -> {PointSize[0.01], Black}, PlotRange -> All, Joined -> False];

g1 = Show[HBgraph, HBFitgraph, KNIgraph, KNIFitgraph,
  Graphics[Text["HB", {10, 240},
    TextStyle -> {FontFamily -> "Times", FontSize -> 14, FontWeight -> "Bold", Blue}]],
  Graphics[Text["KNI", {77, 180}, TextStyle ->
    {FontFamily -> "Times", FontSize -> 14, FontWeight -> "Bold", Red}]],
  PlotRange -> {{0, 100}, {0, 300}},
  Frame -> True, FrameStyle -> Thick]

```



Example 6 : Running Kinetics inside GeneticNetworks

```
reactions = {Z1 → B, Z2 → H, B + G → GB, GB → B + G, GB → GB + KN, KN → G1,
  H + G → GH, GH → H + G, GB + H → GBH, GBH → GB + H, GH + B → GBH, GBH → GH + B};
```

```
ReactionGraphKinetics[reactions]
```

$$Z1 \xrightarrow{k_1} B$$

$$Z2 \xrightarrow{k_2} H$$

$$B + G \xrightarrow{k_3} GB$$

$$GB \xrightarrow{k_4} B + G$$

$$GB \xrightarrow{k_5} GB + KN$$

$$KN \xrightarrow{k_6} G1$$

$$G + H \xrightarrow{k_7} GH$$

$$GH \xrightarrow{k_8} G + H$$

$$GB + H \xrightarrow{k_9} GBH$$

$$GBH \xrightarrow{k_{10}} GB + H$$

$$B + GH \xrightarrow{k_{11}} GBH$$

$$GBH \xrightarrow{k_{12}} B + GH$$

```
Print["Substances with 'Kinetics':"]
```

```
substancenames1 = SubstanceNamesKinetics[reactions]
```

```
substancevariables1 = SubstanceVariablesKinetics[reactions]
```

```
substanceinitcond1 = SubstanceInitialConditionsKinetics[reactions]
```

```
Print["Substances with 'GeneticNetworks':"]
```

```
substancenames1 = SubstanceNames[reactions]
```

```
substancevariables1 = SubstanceVariables[reactions]
```

```
substanceinitcond1 = SubstanceInitialConditions[reactions]
```

```
Print["Parameters:"]
```

```
par1 = ParameterNamesKinetics[reactions]
```

```
parInput1 = ParameterInputKinetics[reactions]
```

```
Print["Time evolution Equations:"]
```

```
equations1 = EquationsKinetics[reactions]; equations1 // Column
```

```
Print["Conservation Laws:"];
```

```
cons1 = ConservationLawsKinetics[reactions]; cons1 // Column
```

```
Substances with 'Kinetics':
```

```
{B, G, G1, GB, GBH, GH, H, KN, Z1, Z2}
```

```
{B[t], G[t], G1[t], GB[t], GBH[t], GH[t], H[t], KN[t], Z1[t], Z2[t]}
```

```
{B[0] == InitCond[1], G[0] == InitCond[2], G1[0] == InitCond[3],
  GB[0] == InitCond[4], GBH[0] == InitCond[5], GH[0] == InitCond[6],
  H[0] == InitCond[7], KN[0] == InitCond[8], Z1[0] == InitCond[9], Z2[0] == InitCond[10]}
```

Substances with 'GeneticNetworks':

```
{B, G, G1, GB, GBH, GH, H, KN, Z1, Z2}

{B[t], G[t], G1[t], GB[t], GBH[t], GH[t], H[t], KN[t], Z1[t], Z2[t]}

{B[0] == InitCond[1], G[0] == InitCond[2], G1[0] == InitCond[3],
 GB[0] == InitCond[4], GBH[0] == InitCond[5], GH[0] == InitCond[6],
 H[0] == InitCond[7], KN[0] == InitCond[8], Z1[0] == InitCond[9], Z2[0] == InitCond[10]}
```

Parameters:

```
{k1, k2, k3, k4, k5, k6, k7, k8, k9, k10, k11, k12}

{k1 → ParInputVar[1], k2 → ParInputVar[2], k3 → ParInputVar[3], k4 → ParInputVar[4],
 k5 → ParInputVar[5], k6 → ParInputVar[6], k7 → ParInputVar[7], k8 → ParInputVar[8],
 k9 → ParInputVar[9], k10 → ParInputVar[10], k11 → ParInputVar[11], k12 → ParInputVar[12]}
```

Time evolution Equations:

```
B'[t] == -B[t] G[t] k3 + GB[t] k4 - B[t] GH[t] k11 + GBH[t] k12 + k1 Z1[t]
G'[t] == -B[t] G[t] k3 + GB[t] k4 - G[t] H[t] k7 + GH[t] k8
G1'[t] == KN[t] k6
GB'[t] == B[t] G[t] k3 - GB[t] k4 - GB[t] H[t] k9 + GBH[t] k10
GBH'[t] == GB[t] H[t] k9 - GBH[t] k10 + B[t] GH[t] k11 - GBH[t] k12
GH'[t] == G[t] H[t] k7 - GH[t] k8 - B[t] GH[t] k11 + GBH[t] k12
H'[t] == -G[t] H[t] k7 + GH[t] k8 - GB[t] H[t] k9 + GBH[t] k10 + k2 Z2[t]
KN'[t] == GB[t] k5 - KN[t] k6
Z1'[t] == -k1 Z1[t]
Z2'[t] == -k2 Z2[t]
```

Conservation Laws:

```
-G[t] - GB[t] + H[t] + Z2[t]
B[t] + GB[t] + GBH[t] + Z1[t]
G[t] + GB[t] + GBH[t] + GH[t]
```

equations1

```
{B'[t] == -B[t] G[t] k3 + GB[t] k4 - B[t] GH[t] k11 + GBH[t] k12 + k1 Z1[t],
 G'[t] == -B[t] G[t] k3 + GB[t] k4 - G[t] H[t] k7 + GH[t] k8, G1'[t] == KN[t] k6,
 GB'[t] == B[t] G[t] k3 - GB[t] k4 - GB[t] H[t] k9 + GBH[t] k10,
 GBH'[t] == GB[t] H[t] k9 - GBH[t] k10 + B[t] GH[t] k11 - GBH[t] k12,
 GH'[t] == G[t] H[t] k7 - GH[t] k8 - B[t] GH[t] k11 + GBH[t] k12,
 H'[t] == -G[t] H[t] k7 + GH[t] k8 - GB[t] H[t] k9 + GBH[t] k10 + k2 Z2[t],
 KN'[t] == GB[t] k5 - KN[t] k6, Z1'[t] == -k1 Z1[t], Z2'[t] == -k2 Z2[t]}
```

equations2 = equations1 /. k1 → 0 /. k2 → 0

```
{B'[t] == -B[t] G[t] k3 + GB[t] k4 - B[t] GH[t] k11 + GBH[t] k12,
 G'[t] == -B[t] G[t] k3 + GB[t] k4 - G[t] H[t] k7 + GH[t] k8, G1'[t] == KN[t] k6,
 GB'[t] == B[t] G[t] k3 - GB[t] k4 - GB[t] H[t] k9 + GBH[t] k10,
 GBH'[t] == GB[t] H[t] k9 - GBH[t] k10 + B[t] GH[t] k11 - GBH[t] k12,
 GH'[t] == G[t] H[t] k7 - GH[t] k8 - B[t] GH[t] k11 + GBH[t] k12,
 H'[t] == -G[t] H[t] k7 + GH[t] k8 - GB[t] H[t] k9 + GBH[t] k10,
 KN'[t] == GB[t] k5 - KN[t] k6, Z1'[t] == 0, Z2'[t] == 0}
```

```

equations2 = Drop[equations2, {9, 10}];
equations2 = Drop[equations2, {3}]; equations2 // Column

B'[t] == -B[t] G[t] k3 + GB[t] k4 - B[t] GH[t] k11 + GBH[t] k12
G'[t] == -B[t] G[t] k3 + GB[t] k4 - G[t] H[t] k7 + GH[t] k8
GB'[t] == B[t] G[t] k3 - GB[t] k4 - GB[t] H[t] k9 + GBH[t] k10
GBH'[t] == GB[t] H[t] k9 - GBH[t] k10 + B[t] GH[t] k11 - GBH[t] k12
GH'[t] == G[t] H[t] k7 - GH[t] k8 - B[t] GH[t] k11 + GBH[t] k12
H'[t] == -G[t] H[t] k7 + GH[t] k8 - GB[t] H[t] k9 + GBH[t] k10
KN'[t] == GB[t] k5 - KN[t] k6

parInput1

{k1 → ParInputVar[1], k2 → ParInputVar[2], k3 → ParInputVar[3], k4 → ParInputVar[4],
 k5 → ParInputVar[5], k6 → ParInputVar[6], k7 → ParInputVar[7], k8 → ParInputVar[8],
 k9 → ParInputVar[9], k10 → ParInputVar[10], k11 → ParInputVar[11], k12 → ParInputVar[12]}

parInput2 = parInput1; parInput2 = Drop[parInput2, 2];

parInput2

{k3 → ParInputVar[3], k4 → ParInputVar[4], k5 → ParInputVar[5],
 k6 → ParInputVar[6], k7 → ParInputVar[7], k8 → ParInputVar[8], k9 → ParInputVar[9],
 k10 → ParInputVar[10], k11 → ParInputVar[11], k12 → ParInputVar[12]}

substanceinitcond2 = Drop[Drop[substanceinitcond1, {9, 10}], {3}]

{B[0] == InitCond[1], G[0] == InitCond[2], GB[0] == InitCond[4],
 GBH[0] == InitCond[5], GH[0] == InitCond[6], H[0] == InitCond[7], KN[0] == InitCond[8]}

substancenames2 = Drop[Drop[substancenames1, {9, 10}], {3}]

{B, G, GB, GBH, GH, H, KN}

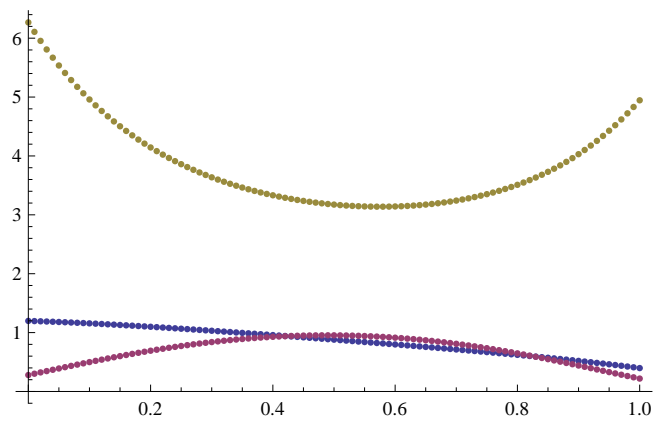
parInputVar[1] = 0; ParInputVar[2] = 0;
(* degradation rate of KNI *) ParInputVar[6] = 0.1;
ParInputVar[3] = 1.0; ParInputVar[4] = 0.1;
(* production of KNI *) ParInputVar[5] = 1.0;
(*repression*)
ParInputVar[7] = 1.0; ParInputVar[8] = 0.1;
ParInputVar[9] = 0.1; ParInputVar[10] = 0.1;
ParInputVar[11] = 0.1; ParInputVar[12] = 0.1;

InitCond[2] = 1.0; InitCond[4] = 0.0; InitCond[5] = 0; InitCond[6] = 0; InitCond[8] = 0;
(*BCD*) InitCond[1] = 2.0; (*HB*) InitCond[7] = 0.5;

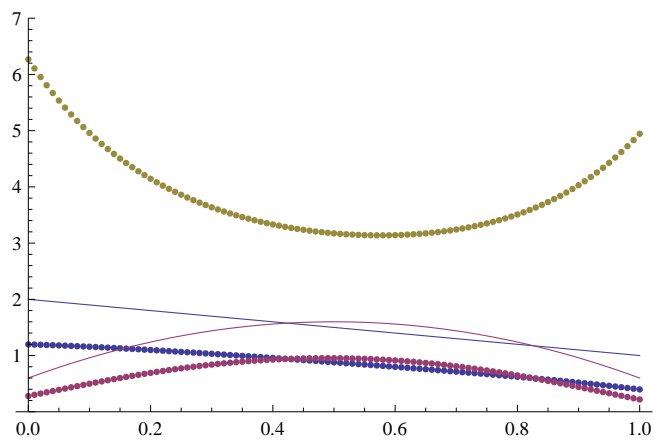
fb[x_] := 2 - x; fh[x_] := 1.6 - 4 (x - 0.5)2;
B1 = {}; H1 = {}; KN1 = {};
For[x = 0, x ≤ 1.0, x += 0.01,
  InitCond[2] = 1.0; InitCond[4] = 0.0; InitCond[5] = 0; InitCond[6] = 0; InitCond[8] = 0;
  (*BCD*) InitCond[1] = fb[x]; (*HB*) InitCond[7] = fh[x];
  sol = NDSolve[
    Join[equations2 /. parInput2, substanceinitcond2, substancenames2, {t, 0, 100}];
    B1 = Append[B1, {x, B[100.0] /. sol[[1]]}];
    H1 = Append[H1, {x, H[100.0] /. sol[[1]]}];
    KN1 = Append[KN1, {x, KN[100.0] /. sol[[1]]}];
  ];

```

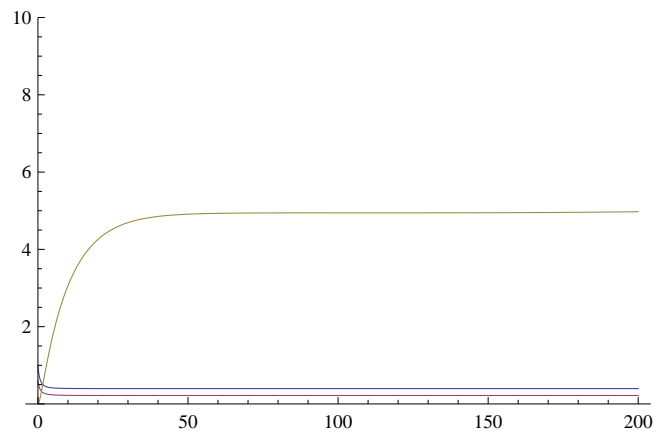
```
g1 = ListPlot[{B1, H1, KN1}]
```



```
Show[{g1, g3}, PlotRange -> {0, 7}]
```



```
Plot[Evaluate[{B[t], H[t], KN[t]} /. sol], {t, 0, 200}, PlotRange -> {0, 10}]
```



```
g3 = Plot[{fb[x], fh[x]}, {x, 0, 1}, PlotRange -> {0, 2}]
```

