BioSwitch: a tool for the detection of multi-steady state behaviour and bistability in signaling and gene regulatory networks version 1.0.0

User's guide

Pencho Yordanov [*a], Jörg Stelling [†a], and Irene Otero-Muras [‡b]

[a]Department of Biosystems Science and Engineering,
ETH-Zürich and SIB Swiss Institute of Bioinformatics, Basel,
Switzerland.
[b]Process Engineering Group, IIM-CSIC, Spanish National
Research Council, Eduardo Cabello 6, 36208 Vigo, Spain.

CSIC
Consejo Superior de Investigaciones Científicas

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

[*] Current Affiliation: Department of Systems Biology, Harvard Medical School, 200 Longwood Avenue, Boston Ma 02115, USA, pencho_yordanov@hms.harvard.edu

[†] joerg.stelling@bsse.ethz.ch

[‡] ireneotero@iim.csic.es

Contents

1 BioSwitch 4

1.1 Software requirements and compatibility 4

1.2 Overview 5

1.2.1 Limit point, limit point bifurcation and bistability . . . 5

1.2.2 Detection of fold bifurcations 6

1.2.3 Inputs and outputs of BioSwitch 6

1.2.4 Applicability of BioSwitch 6

1.2.5 Scheme of the toolbox 8

1.3 Installation 10

1.4 Questions and troubleshooting 10

1.5 About this manual 10

2 Quick start 10

3 Definition of a reaction network to be analyzed with BioSwitch 11

4 Evaluation of a reaction network 12

5 Searching for limit points in reaction networks with mass conservation 14

5.1 Generating the files necessary for limit point detection in reaction networks with mass conservation 14

5.2 Detection of limit points for networks with mass conservation 15

6 Searching for limit points in networks without mass conservation laws 18

6.1 Obtaining the semi-diffusive counterpart of a reaction network 18

6.1.1 If we start from a network with mass conservation laws 18

6.1.2 If the evaluation of the network by `BioSWITCH_Evaluate` detects that there are no mass conservation laws 18

6.2 Generating the files necessary for limit point detection in semi-diffusive reaction networks 19

6.3 Detection of limit points in semi-diffusive reaction networks . . 21

7 Starting a numerical continuation from a detected limit point for reaction networks with mass conservation 24

7.1 Generating the necessary files for the numerical continuation for networks with mass conservation 24

7.2 Numerical continuation from a detected limit point in networks with mass conservation 25

8 Starting a numerical continuation from a detected limit point for semi-diffusive reaction networks 26

8.1 Generating the necessary files for the numerical continuation for semi-diffusive networks 26

8.2 Numerical continuation from a detected limit point in networks with mass conservation 27

9 Examples 28

9.1 Analysis of the G1/S transition in the cell cycle of *Saccharomyces cerevisiae* 30

1 BioSwitch

- Webpage: `https://sites.google.com/view/bioswitch`
- Authors: Irene Otero-Muras, Pencho Yordanov, Jörg Stelling.
- e-mail: ireneotero@iim.csic.es
- Version: 1.0.0
- License: GPLv3
- Copyright: CSIC, Spanish National Research Council

1.1 Software requirements and compatibility

✓ BioSwitch is compatible with Matlab under Windows and Linux.

✓ BioSwitch has been tested with Matlab versions 2015b and 2018b (Matlab Symbolic Toolbox is required).

✓ BioSwitch allows the search of limit points in reaction networks using optimization. This feature requires the MEIGO optimization toolbox, freely available at:

`http://gingproc.iim.csic.es/meigom.html`

Download the MEIGO-M package (version MEIGO_M-v03-07-2014) and copy folder MEIGO_M-v03-07-2014 to BioSWITCH_files/Opt_solvers.

✓ BioSwitch automatically generates the files and default data to start a continuation with Cl_Matcont. Numerical continuation requires Cl_Matcont, freely available at:

`https://sourceforge.net/projects/matcont/files/matcont/matcont5p4/`

Download the Cl_Matcont package (version cl_matcont5p4) and copy folder cl_matcont5p4 to BioSWITCH_files/Cont_solvers.

1.2 Overview

1.2.1 Limit point, limit point bifurcation and bistability

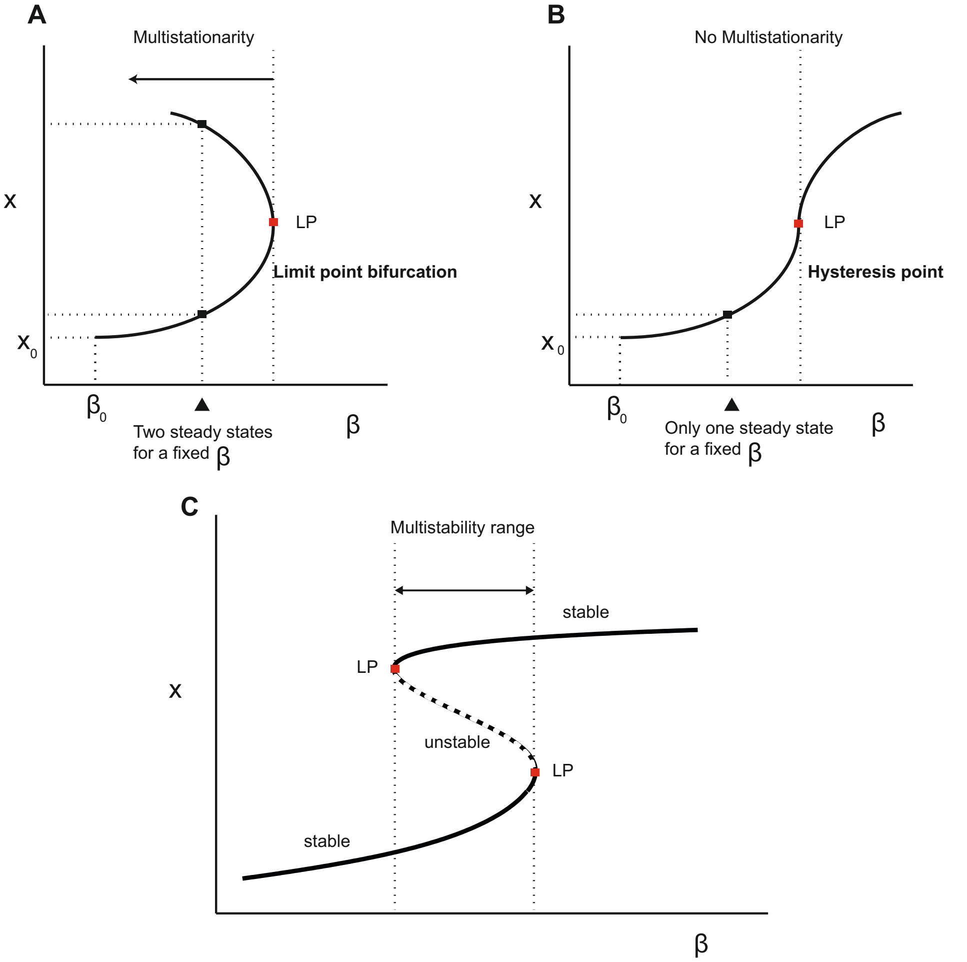

Figure 1: Bifurcation diagrams representative of A) Limit point bifurcation B) Hysteresis point C) Bistability) The steady state concentration x of a species of interest is plotted against the bifurcation parameter β (usually a kinetic parameter or a mass conservation constant). A) Limit point that is also a limit point bifurcation (also known as saddle-node or fold bifurcation), in such case the system is multistationary. B) Limit point that is not a fold bifurcation, in such case the system is not multistationary. C) Typical bifurcation diagram for a bistable system, the region of bistability is enclosed by two fold bifurcations.

The fold bifurcation is a precursor of bistability in biochemical reaction networks. We illustrate this in Fig. 1 by means of representative bifurcation diagrams. Detailed methods and definitions can be found in [Otero-Muras *et al* (2017)] (Supporting information).

1.2.2 Detection of fold bifurcations

BioSwitch implements two different algorithms for the detection of fold bifurcations in biochemical reaction networks with mass action kinetics:

1. An algorithm searching for fold bifurcations in networks with mass conservation, based on [Otero-Muras (2012)] and extended in [Otero-Muras *et al* (2017)]. This method relies on the *deficiency* branch of the Chemical Reaction Network Theory (CRNT).
2. An algorithm searching for fold bifurcations in networks without mass conservation, based on [Otero-Muras *et al* (2017)]. This method relies on the *injectivity* branch of CRNT.

1.2.3 Inputs and outputs of BioSwitch

INPUT: Only the reaction network graph (C-graph) of the network under study is required. Parameter values are not needed. If some parameters are available, their values can be incorporated (fixed during the search).

OUTPUT (1): The toolbox searches for limit points and provides their coordinates if a limit point is found.

OUTPUT (2): If Cl_Matcont is installed, a bifurcation diagram (starting from the limit point found) is automatically generated.

1.2.4 Applicability of BioSwitch

The broad majority of mass action reaction networks with relevance in biology directly fulfill or can be easily accommodated to fulfill the assumptions required by BioSwitch.

BioSwitch analyzes two types of networks: i) networks with mass conservation laws and ii) semi-diffusive networks (no conservation laws are active). In both types the following assumptions need to be satisfied:

- A.1 All reactions are endowed with mass action kinetics.
- A.2 The network admits a strictly positive steady state (where the concentrations of all species are positive). This condition can be checked by simulation.

Note that both assumptions are mild in the context of signaling and gene regulation. In particular, assumption A.2 is mild in the context of signaling, due to the inherent reversibility of most protein-protein interactions. In practice, a signaling model can always be modified without loss of generality to avoid zero steady states (assuming very low concentrations instead) such that the assumption is fulfilled.

Next we describe the additional assumptions made for each network type.

i) Networks with mass conservation.

The number of mass conservation relations of a reaction network is given by $\lambda = Ns$ where N is the number of species and s is the rank of the stoichiometric matrix. If $\lambda > 0$ we say that the network has mass conservation laws, see [Otero-Muras *et al* (2017)] for more details. The method implemented in BioSwitch to detect limit points for networks with mass conservation requires an additional assumption:

- A.3 The graph of the network is uniterminal [Otero-Muras *et al* (2017)].

Assumption A.3 is mild, since networks which are non-uniterminal are considered, in general, to be unsuited for the description of real chemical systems.

ii) Semi-diffusive networks.

In fully diffusive networks all species are present in the feed and outflow streams. Fully diffusive networks do not have mass conservation laws but they do not comply in general with natural assumptions in signaling pathways.

Semi-diffusive networks do not have mass conservation laws, and they comply with the natural assumptions for signaling pathways and gene regulatory networks.

In semi-diffusive networks, there is a degradation reaction for each species, as well as an inflow (or basal formation) of one species per conservation law. This species is chosen to be the free form of the corresponding protein participating in the conservation law. If the network is not weakly reversible we need to ensure that our choice is compatible with the existence of a strictly positive steady state (A.2).

Important: Note that i) the semi-diffusive conditions are mild in the context of signaling and gene regulation and ii) every reaction network can be transformed to obtain its semi-diffusive counterpart.[1]

1.2.5 Scheme of the toolbox

The BioSWITCH directory contains the following folders:

- BioSWITCH_files contains internal toolbox code (not to be modified by the user) and external software (see Section 1.1).
- BioSWITCH_documentation contains the user's manual.
- BioSWITCH_examples contains several examples (ready to run from the Command Window).
- USR_inputfiles contains the functions to generate the files needed for Lpsearch (searching for a limit point) and Ncontin (starting an equilibrium continuation).

[1] For more information on the implications of semi-diffusive networks we recommend reading [Otero-Muras *et al* (2017)].

- USR_networks contains the file defining the reaction network.
- USR_results is the folder where the results are stored.

The main tasks available in BioSWITCH (functions in the BioSWITCH folder) are:

- `BioSWITCH_Evaluate` evaluates a reaction network and provides useful information, including whether there are conservation laws or not.
- `BioSWITCH_Lpsearch` performs a parameter space exploration to search for a limit point.
- `BioSWITCH_Ncontin` calls the continuation solver to perform numerical continuation starting from a detected limit point.

The USR_inputfiles directory contains two folders:

- lpsearch_files contains functions for the following tasks:
  - `BioSWITCH_mkfiles_lpsearch_MassCon` generates files for limit point exploration in networks with mass conservation.
  - `BioSWITCH_mkfiles_lpsearch_SemiDiff` generates files for limit point exploration in semi-diffusive networks (no mass conservation).
- continua_files contains functions for the following tasks:
  - `BioSWITCH_mkfiles_ncontin_MassCon` generates files for numerical continuation in networks with mass conservation.
  - `BioSWITCH_mkfiles_ncontin_SemiDiff` generates files for numerical continuation in semi-diffusive networks (no mass conservation).

1.3 Installation

1. Copy the BioSWITCH folder in a directory of your choice.
2. Copy the MEIGO_M-v03-07-2014 folder[2] to BioSWITCH/BioSWITCH_files/Opt_solvers.
3. Copy the cl_matcont5p4 folder[3] to BioSWITCH/BioSWITCH_files/Cont_solvers.
4. Every time before its first use in a Matlab session, run BioSWITCH_Startup and all relevant files will be added to the Matlab path.

1.4 Questions and troubleshooting

For questions, feedback and troubleshooting, please contact `ireneotero@iim.csic.es`.

1.5 About this manual

Basic knowledge on reaction networks, dynamical systems and Matlab usage is assumed.

2 Quick start

- Start Matlab.
- Go to the BioSWITCH directory.
- Type: BioSWITCH_Startup (BioSWITCH folders and required software are added to the Matlab path).
- Solve the first example by running:

```
    Run_Example_0
```

[2] See Section 1.1

[3] See Section 1.1

You can run any of the examples provided in the USR_networks folder by typing its name in the command window.

3 Definition of a reaction network to be analyzed with BioSwitch

Define the network to be analyzed in a network function file of the form:

```
function [ESP, COM, KNS] = EX0
% user defined reaction network

ESP ={'A','B','AB','ABp'}; % species names
COM = {'A+B','AB','ABp'}; % complexes
KNS = {'k_1_2','k_2_1','k_2_3'}; % kinetic constants

end
```

Figure 2: Network function file

and store it in the USR_networks folder. The network function file contains three cell arrays with the names of the species, the complexes, and the kinetic constants, respectively. The graph of complexes (C-graph) corresponding to the network function file in Fig. 2 is shown in Fig. 3.

The complexes of a reaction network are the objects at the head or tail of reaction arrows. The C-graph is a directed graph associated to the reaction network in which nodes are the complexes and edges are associated to the reactions.

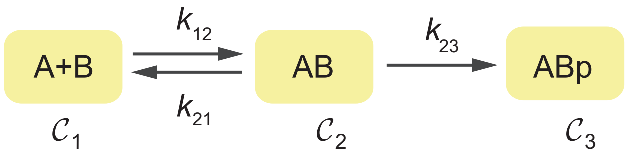

Figure 3: Graph of complexes (C-graph) of the EX0 network.

Important: Please, use the following naming conventions:

- Species: 'A', 'E1', 'ACRp', do not use 'E' or 'I'.
- Reactions: 'k_1_10', source complex 1, product complex 10.
- Complexes: 'A+B', '2*A', use A+A or 2*A but not 2A. Use '0' for the zero complex (environment).

4 Evaluation of a reaction network

Within the BioSWITCH directory, run the BioSWITCH_evaluate function with the name of the network file as an argument, for example:

```
BioSWITCH_Evaluate('EX0')
```

The report with the evaluation results is displayed in the commmand window and saved in a txt file (in this case `Evaluation_EX0.txt`). The following information is provided:

```
The number of species (n) is: 4

species 1: A
species 2: B
species 3: AB
species 4: ABp

The number of complexes (m) is: 3

complex 1: A+B
complex 2: AB
complex 3: ABp

The number of linkage classes (l) is: 1

n > m-l (underdimensioned network)
```

```
The rank of the stoichiometrix matrix (rho) is: 2

The number of mass conservation laws (n-rho) is: 2

Conservation law 1: A + AB + ABp
Conservation law 2: AB + ABp + B

Reactions are classified in:
k_1_2 : true reaction
k_2_1 : true reaction
k_2_3 : true reaction
```

Variables internally used by BioSwitch that can be of interest for the user are also shown:

```
outd =
        def: 0
     n_link: 1
      n_com: 3
      n_esp: 4
      n_rec: 3
      Psi_c: [3x1 sym]
        Psi: [3x1 sym]
    modeleq: [4x1 sym]
          N: [4x3 double]
        esp: {'A'  'B'  'AB'  'ABp'}
        com: {'A+B'  'AB'  'ABp'}
        kns: {'k_1_2'  'k_2_1'  'k_2_3'}
          Y: [4x3 double]
      basis: [3x0 double]
        A_k: [3x3 sym]
     A_kPsi: [3x1 sym]
        rhs: [3x1 double]
  alpha_vec: []
```

Finally, the deficiency of the network is provided:

```
The deficiency of the network is: 0
If the network is weakly reversible: there is a unique
   steady state per stoichiometric compatibility class (for
    all parameter sets)
If the network is not weakly reversible, no positive
   equilibrium is admitted
```

In case the network is of deficiency zero, no multistationarity is possible and the two possible outcomes are indicated.

5 Searching for limit points in reaction networks with mass conservation

5.1 Generating the files necessary for limit point detection in reaction networks with mass conservation

Run BioSWITCH_mkfiles_lpsearch_MassCon, located in folder USR_inputfiles/lpsearch_files, with the name of the network file as an argument, for example:

```
BioSWITCH_mkfiles_lpsearch_MassCon('EX1')
```

where 'EX1' is the name of the reaction network function of interest. The command produces a txt file containing the manifold equations. Next, the following message is displayed:

```
 Selection of independent variables:
 Choose 0 for automatic selection,
 Choose 1 for manual selection

Type 0 (automatic), 1 (manual):0
```

If we type 0, the algorithm automatically selects the independent variables to solve the manifold equations.

If we type 1, we can choose the independent variables to solve the manifold equations. The options are displayed, for example, for EX1:

```
There are 7 possible vectors of independent variables
(check EX1_manifold.txt), choose a vector from 1 to 7
```

We then type the number corresponding to the selected vector. The following files are generated:

- `EX1_ObjF_DEF.m` (contains the objective function),
- `EX1_param_MassCon.m` (contains the expressions of the steady state concentrations in terms of the kinetic parameters),
- `EX1_input_lpsearch_MassCont_default` (contains the file with the default information for the optimization problem).

5.2 Detection of limit points for networks with mass conservation

Search Options: Before starting the search, the user can modify the default options for the search by editing `EX1_input_lpsearch_DEF_default.m`. We recommend to keep the default options for the first exploration.

```
% Default Input File for Optimization (networks with mass conservation)

% Default Decision Variables
% k_1_5
% k_5_1
% k_1_6
% k_2_8
% k_8_2
% k_2_7
% k_3_9
% k_9_3
% k_3_4
% alpha_1
% alpha_2
% x_4

% Objective Function File
```

```
problem.f='EX1_ObjF_DEF';

% lower and upper bounds for the decision variables
problem.x_L=[0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   0.01   -1e+03   -1e+03   0.01];
problem.x_U=[1e+02   1e+02   1e+02   1e+02   1e+02   1e+02   1e+02   1e+02   1e+02   1e+03   1e+03   1e+02];
problem.x_0=[1   1   1   1   1   1   1   1   1   1   1   1];
problem.c_L=[0.01   0.01   0.01   0.01   0.01   0.01];
problem.c_U=[1e+02   1e+02 1e+02 1e+02 1e+02 1e+02];
opts.maxtime=150;
```

The optimization problem contains the following options:

- problem.X_L contains the lower bounds of the decision variables,
- problem.X_U contains the upper bounds for the decision variables,
- problem.c_L contains the lower bounds for the steady state concentrations,
- problem.c_L contains the upper bounds for the steady state concentrations,
- opts.maxtime is the maximum time allowed for the optimization solver.

All options for the optimization can be found in the MEIGO_Matlab_userguide [Egea *et al* (2014)].

Running an exploration: Go to the main folder BioSWITCH and type in the command window:

```
BioSWITCH_Lpsearch('input_lpsearch_MassCont_default')
```

where 'input_lpsearch_MassCont_default' is the name of the default options file generated with `BioSWITCH_mkfiles_lpsearch_MassCon`. For example, for the network 'EX1':

```
BioSWITCH_Lpsearch('EX1_input_lpsearch_MassCont_default')
```

The intermediate and final results of the search are displayed in the command window. Additionally, the results are stored in the file:

`'EX1_input_lpsearch_MassCon_default_Results.mat'`.

Within the file, the structure Results contains:

- Results.f: a vector of values of the objective function at different times (in Results.time),
- Results.x: a matrix with rows being the values of the decision variables at different times (in Results.time),
- Results.time: a vector of times (in seconds),
- Results.neval: contains the number of objective function evaluations,
- Results.fbest: contains the optimal value of the objective function found,
- Results.xbest: contains the optimal decision vector,
- Results.cputime: contains the CPU time consumed by the algorithm.

More details about the Results structure can be found in the MEIGO user-manual.

If Results.fbest is zero, the vector Results.xbest contains the decision variables leading to a limit point.

Please note that, due to numerical issues, we do not expect to find $f = 0.0$ for a limit point, but a real number which is small enough.

If a zero is not found we recommend to call BioSWITCH_Lpsearch again. Note that the optimization algorithm is stochastic and this means that every time we run BioSWITCH_Lpsearch it may find a different solution. Note also that, if the system has one limit point, then it has infinite limit points (i.e. infinetely many parameter vectors leading to a limit point) and BioSWITCH_Lpsearch can be called several times to find different valid solutions.

Please note that any prior information available with respect to the parameters and/or steady state concentrations that we want to impose, can be introduced by modifying accordingly the `input_lpsearch_MassCont_default` file (in this case we recommend to save the file with a different name such as `input_lpsearch_MassCont_v1`, or similar).

We recommend to store all files generated for a particular network in a folder with the name of the network in BioSWITCH/USR_results. For example, for EX1 the user should create the folder BioSWITCH/USR_results/EX1.

6 Searching for limit points in networks without mass conservation laws

BioSwitch allows to explore the existence of limit points in semi-diffusive networks. Note that we can always build the semi-diffusive counterpart of any reaction network and that this semi-diffusive counterpart is by construction compatible with standard assumptions in signaling.

6.1 Obtaining the semi-diffusive counterpart of a reaction network

6.1.1 If we start from a network with mass conservation laws

In this case the semi-diffusive counterpart is generated automatically by BioSwitch, starting from the original reaction network file.

6.1.2 If the evaluation of the network by `BioSWITCH_Evaluate` detects that there are no mass conservation laws

In this case we have to:

- Build a network with the true reactions only and save it in USR_networks.

(We recommend to use ”tr” at the end of the file name, for example EX2tr). Note that the true reactions are indicated in the evaluation with `BioSWITCH_Evaluate`.

- Use this network with true reactions to construct the files and perform the search (BioSwitch automatically generates the semi-diffusive counterpart of a network with mass conservation).

Important: When building the reaction network file with only true reactions, do not forget to renumber the kinetic constants accordingly.

6.2 Generating the files necessary for limit point detection in semi-diffusive reaction networks

Starting from a network with mass conservation and true reactions only, the function `BioSWITCH_mkfiles_lpsearch_SemiDiff` generates the files for limit point detection in the semi-diffusive network counterpart.

If we have a network with inflow and/or outflow reactions (with or without mass conservation) we need to generate first a network file with true reactions only (see 6.1), and `BioSWITCH_mkfiles_lpsearch_SemiDiff` will generate the files for limit point detection in the semi-diffusive counterpart.

Go to USR_inputfiles/lpsearch_files and execute in the command window:

```
BioSWITCH_mkfiles_lpsearch_SemiDiff('network_file',
    input_species_vector)
```

where ’network_file’ is the name of the file containing the network with true reactions and input_species_vector is a row vector containing indices of input species (one per conservation law) in the semi-diffusive network. Any choice of indices is valid as soon as we pick exactly one species from each of the conservation laws.

Important: The indices must be in the right order according to the labeling of the species (from small to large indices). For example, if the species are ordered as 'A', 'B', 'C', 'D' and the conservation laws are $C1 = A + B + C$, $C2 = 2A + D$, if we choose C and A to be in the inflow, the species vector is [1,3].

To illustrate the procedure, let us take EX2 from USR_networks. The evaluation reveals that EX2 has one conservation law, 5 true reactions, 2 inflow and 2 outflow reactions.

```
Conservation law 1: X + XS + XSS

Reactions are classified in:
k_1_2 : true reaction
k_2_1 : true reaction
k_2_3 : true reaction
k_4_5 : true reaction
k_5_4 : true reaction
k_6_8 : outflow reaction
k_8_6 : inflow reaction
k_7_8 : outflow reaction
k_8_7 : inflow reaction
```

We need to build a reaction network file containing only the true reactions as follows (`EX2tr.m`):

```
function [ESP, COM, KNS] = EX2tr
% network with true REACTIONS only

 ESP={'X','S','XS','XSS','P'};
 COM={'X+S','XS','X+P','XS+S','XSS'};
 KNS={'k_1_2' 'k_2_1' 'k_2_3' 'k_4_5' 'k_5_4'};
```

If we evaluate `EX2tr` we get:

```


Conservation law 1: X + XS + XSS
Conservation law 2: P + S + XS + 2*XSS

Reactions are classified in:
k_1_2 : true reaction
k_2_1 : true reaction
k_2_3 : true reaction
```

```
k_4_5 : true reaction
k_5_4 : true reaction
```

There are two conservation laws and, therefore, we need to choose one species from each conservation law to be part of the inflow. We choose X and S to be in the inflow (i.e. species 1 and 2 of the network).

In order to build the files for limit point detection in the corresponding semi-diffusive network, we go to USR_inputfiles/lpsearch_files and execute in the command window:

```
BioSWITCH_mkfiles_lpsearch_SemiDiff('EX2tr',[1,2])
```

The following two files are generated:

- `EX2tr_ObjF_INJ.m` (contains the objective function),
- `EX1tr_input_lpsearch_SemiDiff_default` (contains the file with the default information for the optimization problem).

6.3 Detection of limit points in semi-diffusive reaction networks

Search Options: Before starting the search, the user can modify the default search options by editing `EX2tr_input_lpsearch_DEF_default`. We recommend to keep the default options for the first exploration.

```
% Default Input File for Optimization (semidiffusive case)

% input_species 1 = X (index 1)
% input_species 2 = S (index 2)

% Default Decision Variables
% mu_1
% mu_2
% mu_3
% mu_4
% mu_5
% mu_6
% mu_7
% mu_8
% mu_9
% mu_10

% Objective Function File
problem.f='EX2tr_ObjF_INJ';

% lower and upper bounds for the decission variables
problem.x_L=[    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.01];
problem.x_U=[    1e+02      1e+02      1e+02      1e+02      1e+02      1e+02      1e+02      1e+02      1e+02      1e+02];
problem.x_0=[    1    1    1    1    1    1    1    1    1    1];

% lower and upper bounds for inequality constraints
problem.c_L=[    0    0];
problem.c_U=[    Inf    Inf];

problem.neq=3;

opts.maxtime=150;
```

In this case the decision variables are the fluxes (μ_i) [Otero-Muras *et al* (2017)].

Running an exploration: Go to the main folder BioSWITCH and type in the command window:

```
BioSWITCH_Lpsearch('input_lpsearch_SemiDiff_default')
```

where 'input_lpsearch_MassCont_default' is the name of the default options file generated with `BioSWITCH_mkfiles_lpsearch_MassCon`. For example, for the network 'EX2tr':

```
BioSWITCH_Lpsearch('EX2tr_input_lpsearch_SemiDiff_default')
```

The intermediate and final results of the search are displayed in the command window. Additionally, the results are stored in the file:

`'EX2tr_input_lpsearch_SemiDiff_default_Results.mat'`.

Within the file, the structure Results contains:

- Results.f: a vector of values of the objective function at different times (in Results.time),
- Results.x: a matrix with rows being the values of the decision variables at different times (in Results.time),
- Results.time: a vector of times (in seconds),
- Results.neval: contains the number of objective function evaluations,
- Results.fbest: contains the optimal value of the objective function found,
- Results.xbest: contains the optimal decision vector,
- Results.cputime: contains the CPU time consumed by the algorithm.

More details about the Results structure can be found in the MEIGO user-manual [Egea *et al* (2014)].

If Results.fbest is zero, the vector Results.xbest contains the decision variables leading to a limit point.

Please note that due to numerical issues we do not expect to find $f = 0.0$ for a limit point but a real number which is small enough.

If a zero is not found we recommend to call BioSWITCH_Lpsearch again. Note that the optimization algorithm is stochastic, and this means that every time that we run BioSWITCH_Lpsearch it may find a different solution. Note also that, if the system has a limit point, then it has infinite limit points, and BioSWITCH_Lpsearch can be called several times to find different valid solutions.

Please note that any prior information available with respect to the parameters and/or steady state concentrations that we want to impose, can be introduced by modifying accordingly the `input_lpsearch_SemiDiff_default` file (in this case we recommend to save the file with a different name such as `input_lpsearch_SemiDiff_v1`, or similar).

We recommend to store all generated files for a particular network in a folder with the name of the network in BioSWITCH/USR_results. For example, for EX2tr the user should create the folder BioSWITCH/USR_results/EX2tr.

7 Starting a numerical continuation from a detected limit point for reaction networks with mass conservation

7.1 Generating the necessary files for the numerical continuation for networks with mass conservation

Starting from a network with mass conservation laws and true reactions only, the function `BioSWITCH_mkfiles_ncontin_MassCon` generates the files needed for a numerical continuation.

Go to USR_inputfiles/continua_files and execute in the command window:

```
BioSWITCH_mkfiles_ncontin_MassCon('network_file',
    species_vector)
```

where `'network_file'` is the name of the network file, and `species_vector` is a vector containing the species that we need to remove to get an equivalent DAE system with full rank.

How to build the `species_vector`: it is a row vector containing the indices of species (one per conservation law) we remove from the ODEs to obtain a complete rank system.

Important: The indices must be in the right order according to the labeling of the conservation laws. For example: if the species are ordered as 'A', 'B', 'C', 'D' and the conservation laws are C1 = A + B + C, C2 = 2A + D, if we choose 'A' and 'C' the species vector is [3,1].

To illustrate the procedure, let us take EX1 from USR_networks. The evaluation reveals that EX1 has three conservation laws.

```
The number of species (n) is: 7

species 1: A
species 2: E1
species 3: E2
species 4: Ap
species 5: AE1
species 6: ApE2
species 7: AAp

The number of mass conservation laws (n-rho) is: 3

Conservation law 1: ApE2 + E2
Conservation law 2: AE1 + E1
Conservation law 3: A + 2*AAp + AE1 + Ap + ApE23
```

We choose A, $E1$ and $E2$ to be removed from the ODE system (to build the equivalent DAE system). Then, the `species_vector` is [3,2,1].

We go to USR_inputfiles/continua_files and execute in the command window:

```
BioSWITCH_mkfiles_ncontin_SemiDiff('EX1',[3,2,1])
```

The following two files are generated:

- `EX1_ode_MassCon.m` (contains the ODE file for numerical continuation),
- `EX1_input_cont_MassCon_default` (contains the file with the default information to be used by the continuation algorithm; for more details regarding the options, see the Cl-Matcont manual [Govaerts *et al* (2012)]).

7.2 Numerical continuation from a detected limit point in networks with mass conservation

Changing default options. The default options for the continuation (including tolerances and maximum number of points) can be modified in the default generated file. For EX1, that file is `EX1_input_file_cont_MassCon_default`:

```
opt=contset(opt,'VarTolerance',1e-8);
opt=contset(opt,'FunTolerance',1e-8);
opt=contset(opt,'MaxNumPoints',300);
opt=contset(opt,'Singularities',1);
```

For more details on the available options go to the Cl-Matcont Manual [Govaerts *et al* (2012)]. If changes are made, we recommend to store the altered file with a different name, for example `EX1_input_file_cont_MassCon_v1`.

Calling the continuation algorithm. We go to the main directory and execute in the command window:

```
BioSWITCH_Ncontin('input_file')
```

where `'input_file'` is the name of the file with the previously generated options for the continuer Cl_Matcont [Govaerts *et al* (2012)], for example:

```
BioSWITCH_Ncontin('EX1_input_file_cont_MassCon_default')
```

The `BioSWITCH_Ncontin` function calls Cl_Matcont to perform a continuation in both forward and backward directions starting from the limit point. The expected outcomes are i) a time course plot verifying that the starting limit point is a steady state and ii) a bifurcation diagram which allows the user to check whether the system is multistationary and bistable.

8 Starting a numerical continuation from a detected limit point for semi-diffusive reaction networks

8.1 Generating the necessary files for the numerical continuation for semi-diffusive networks

Starting from a network with true reactions only (for example `EX2tr`, see section 6.1), the function `BioSWITCH_mkfiles_ncontin_SemiDiff` generates the files needed for a numerical continuation of the semi-diffusive counterpart.

Go to USR_inputfiles/continua_files and execute in the command window:

```
BioSWITCH_mkfiles_ncontin_SemiDiff('network_file',
    input_species_vector)
```

where 'network_file' is the name of the network file, and input_species_vector is the vector containing the species selected to be in the input.

Important: Use the same input species vector that has been used to search for the limit point.

If we continue with the EX2tr example, we execute:

```
BioSWITCH_mkfiles_ncontin_SemiDiff('EX2tr',[1  2])
```

We generate the following files:

- EX2tr_ode_SemiDiff.m (contains the ODE file for numerical continuation),
- EX2tr_param_SemiDiff.m (contains the parameters),
- EX2tr_input_cont_SemiDiff_default.m (contains default options for the numerical continuation).

8.2 Numerical continuation from a detected limit point in networks with mass conservation

Changing default options. The default options for the continuation (including tolerances and maximum number of points) can be modified in the default generated file. For EX1, that file is EX1_input_file_cont_MassCon_default:

```
opt=contset(opt,'VarTolerance',1e-8);
opt=contset(opt,'FunTolerance',1e-8);
opt=contset(opt,'MaxNumPoints',300);
opt=contset(opt,'Singularities',1);
```

For more details on the available options go to the Cl-Matcont Manual [Govaerts *et al* (2012)]. If changes are made, we recommend to store the altered file with a different name, for example `EX2tr_input_cont_SemiDiff_v1`.

Calling the continuation algorithm. We go to the main directory and execute in the command window:

```
BioSWITCH_Ncontin('input_file')
```

where `'input_file'` is the name of the previously generated file with the options for the continuer, for example:

```
BioSWITCH_Ncontin('EX2tr_input_cont_SemiDiff_default')
```

The `BioSWITCH_Ncontin` function calls Cl_Matcont to perform a continuation in both forward and backward directions starting from the limit point. The expected outcomes are i) a time course plot verifying that the starting limit point is a steady state and ii) a bifurcation diagram which allows the user to check whether the system is multistationary and bistable.

9 Examples

BioSwitch incorporates a number of built-in examples, ready to be run by the user. For example, to solve example 0, go to the main directory and type:

```
Run_Example_0
```

The user is requested to input an option for the selection of independent variables (0 for automatic, 1 for manual selection).

For the general case, choose 0 (automatic) and press the Return key. If, with the automatic option, the algorithm is not able to solve the symbolic expressions, we recommend to chose 1 (manual), and select (looking at the manifold equations generated in the corresponding .txt file) that vector of variables whose elements lead to an easier solution of the system of equations when taken as independent variables (this is usually very intuitive).

Note that, since default options are used for the continuation, the number of points, or the scale figure axes might not be appropriate to see the bifurcation. If this is the case, we recommend to zoom in/out the portion of the figure around the limit point.

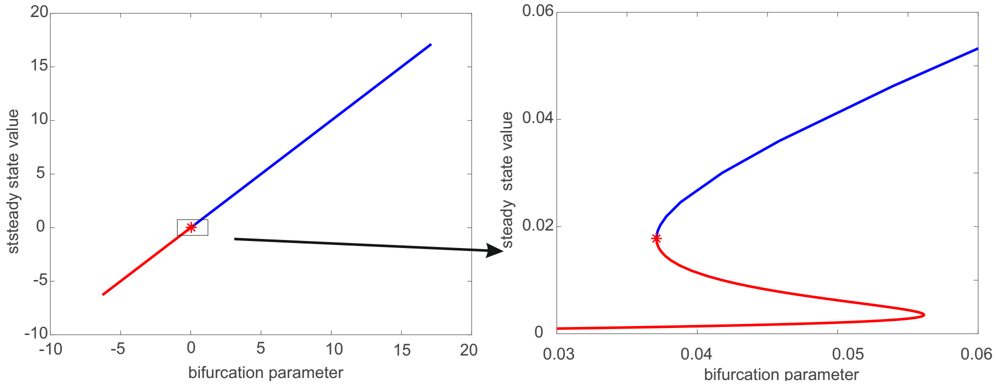

Figure 4: Example in which, although the bifurcation cannot be visually detected in the default diagram, it appears when zoom in around the limit point or modify the axes scale appropriately.

Important: Note that, since global optimization is used to detect limit points, the result will vary for different runs: i.e. a different limit point might be found in each run (different values of the parameters and steady state concentrations). Although unusual, it might also happen that, although the system has a limit point, it is not found in a first run (we always recommend to do a multi-run). If this happens during the execution of a built-in example, it will lead to an error. In case of failure when running the built-in examples, just run the example again.

BioSwitch incorporates a number of reaction networks in `USR_networks`. Next we provide a brief description, as well as the corresponding references and C-graphs of those networks.

- EX0 is a protein complex activation network.
- EX1 is a gene regulatory network with two mutually repressing genes G_1 and G_2 (C-Graph in Fig. 5) introduced in [Otero-Muras *et al* (2014)].

- EX2 is an enzymatic reaction with inhibition by substrate; enzyme X mediates the transformation of substrate S into product P, through the formation of a complex XS, which is in turn inhibited by S; see C-Graph in Fig. 5; introduced in [Otero-Muras *et al* (2009)].
- EX3 is an enzymatic reaction with simple substrate cycle involving two antagonist enzymes U and X (see C-Graph in Fig. 6); originally described by [Hervagault & Canu (2009)].
- EX4 is a signal transduction motif (C-Graph in Fig. 6) introduced in [Conradi *et al* (2005)].
- G1S is the network corresponding to the G1/S transition in the cell cycle of *Saccharomyces cerevisiae* (C-Graph in Fig. 7) described in [Conradi *et al* (2007)].

In the Tables 1 and 2 we include a summary of the analysis and results for networks EX0 to EX4. In the case of EX2 and EX4, we analyze both the original network with mass conservation (Run_Example_2 and Run_Example_4 respectively) and the corresponding semidiffussive versions (Run_Example_2B and Run_Example_4B).

Table 1: Scripts for the analysis of example networks in BioSwitch_examples.

Script	Problem
Run_Example_0	Search for LP in EX0 (Semidiffusive version)
Run_Example_1	Search for LP in EX1 (Mass Conservation)
Run_Example_2	Search for LP in EX2 (Mass Conservation)
Run_Example_2B	Search for LP in EX2 (Semidiffusive version)
Run_Example_3	Search for LP in EX3 (Mass Conservation)
Run_Example_4	Search for LP in EX4 (Mass Conservation)
Run_Example_4B	Search for LP in EX4 (Semidiffusive version)

9.1 Analysis of the G1/S transition in the cell cycle of *Saccharomyces cerevisiae*

We take as a case study the G1/S transition in the cell cycle of *Saccharomyces cerevisiae* originally described in [Conradi *et al* (2007)]. The network graph

Table 2: Results of the analysis with BioSwitch.

Network	Mass Conservation	Semidiffusive
Ex0	Def 0 (no multistationarity)	No LP found
Ex1	LP Found & Bistable	x
Ex2	LP Found & Bistable	No LP found
Ex3	LP Found & Bistable	x
Ex4	LP Found & Bistable	LP Found & Bistable

is depicted in Fig. 7 (G1S).

In `Run_Example_G1S` we have included all function calls needed to solve the problem. To run the example, we just type:

```
Run_Example_G1S
```

After the network evaluation, the algorithm proceeds to generate the files for limit point search, and the following information is displayed on the screen:

```
>>>>>>>>>>>>>>>>>>>>>>>>
>>>> Generate files for limit point search (system with
   mass conservation) ...


    'Selection of independent variables: Choose 0 for
       automatic selection, Choose 1 for manual selection'

Type 0 (automatic), 1 (manual):
```

We type 1 to select the option manual, and press the return button. The algorithm generates the txt file `G1S_manifold.txt`, and the following information is displayed:

```
>>>>>>>>>>>>>>>>>>>>>>>>
>>>>  'There are 91 possible vectors of independent
   variables (check G1S_manifold.txt), choose a vector from
    1 to 91'

chosen vector:
```

Now, we access `G1S_manifold.txt` to select the more convenient set of independent variables. Here we depict the content of `G1S_manifold.txt`:

```
Manifold equations

x_1 - (alpha_1 + alpha_2 - alpha_4)/k_1_2 - k_2_1/k_1_2=0
x_2 + (alpha_2 - alpha_4)/k_3_2=0
x_1*x_3 + (alpha_1*k_5_4 + alpha_2*k_5_4 + alpha_1*k_5_6 -
    alpha_3*k_5_4 + alpha_2*k_5_6 + alpha_5*k_5_4)/(k_4_5*
    k_5_6)=0
x_4 + (alpha_1 + alpha_2 - alpha_3 + alpha_5)/k_5_6=0
x_2*x_3 - (alpha_2*k_8_6 + alpha_2*k_8_7 - alpha_3*k_8_7 +
    alpha_5*k_8_7)/(k_7_8*k_8_6)=0
x_5 - (alpha_2 - alpha_3 + alpha_5)/k_8_6=0
x_3*x_4 + (alpha_3*(k_10_9 + k_10_11))/(k_9_10*k_10_11)=0
x_6 + alpha_3/k_10_11=0
x_2*x_7 + (alpha_4*(k_13_12 + k_13_14))/(k_12_13*k_13_14)=0
x_8 + alpha_4/k_13_14=0
x_5*x_7 + (alpha_5*(k_16_15 + k_16_17))/(k_15_16*k_16_17)=0
x_9 + alpha_5/k_16_17=0

Var Indep

var_indep_1=matrix([[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8
    , x_9, alpha_1, alpha_2, alpha_3]])
var_indep_2=matrix([[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8
    , x_9, alpha_1, alpha_2, alpha_4]])

                              [...]

var_indep_90=matrix([[x_2, x_4, x_5, x_6, x_7, x_8, x_9,
    alpha_1, alpha_2, alpha_3, alpha_4, alpha_5]])
var_indep_91=matrix([[x_3,x_4,x_5,x_6,x_7,x_8,x_9,alpha_1,
    alpha_2,alpha_3,alpha_4,alpha_5]])
```

We need to choose the 9 vector of variables (state vector x) for which the set of equations is going to be solved. If we check the equations of the manifold, we can see that, if we leave x_3 and x_7 out of the set of variables, the equations should be easily solved for x (since the system becomes linear on x). We choose then the following vector:

```
var_indep_63=matrix([[x_1, x_2, x_4, x_5, x_6, x_8, x_9,
    alpha_1, alpha_2, alpha_3, alpha_4, alpha_5]])
```

and consequently type 63 and press the Return button.

Important: For the general case, due to the nature of the manifold equations, it is always very intuitive to select the right set(s) of variables that lead to an easy solution of the equations (making the system "as linear as possible" by avoiding as many second and third order monomials as possible).

Next, the files for limit point search are generated and the search is performed. After finding a limit point, the algorithm calls the continuer Cl_Matcont [Govaerts *et al* (2012)]. The bifurcation diagram obtained is depicted in Fig. 8.

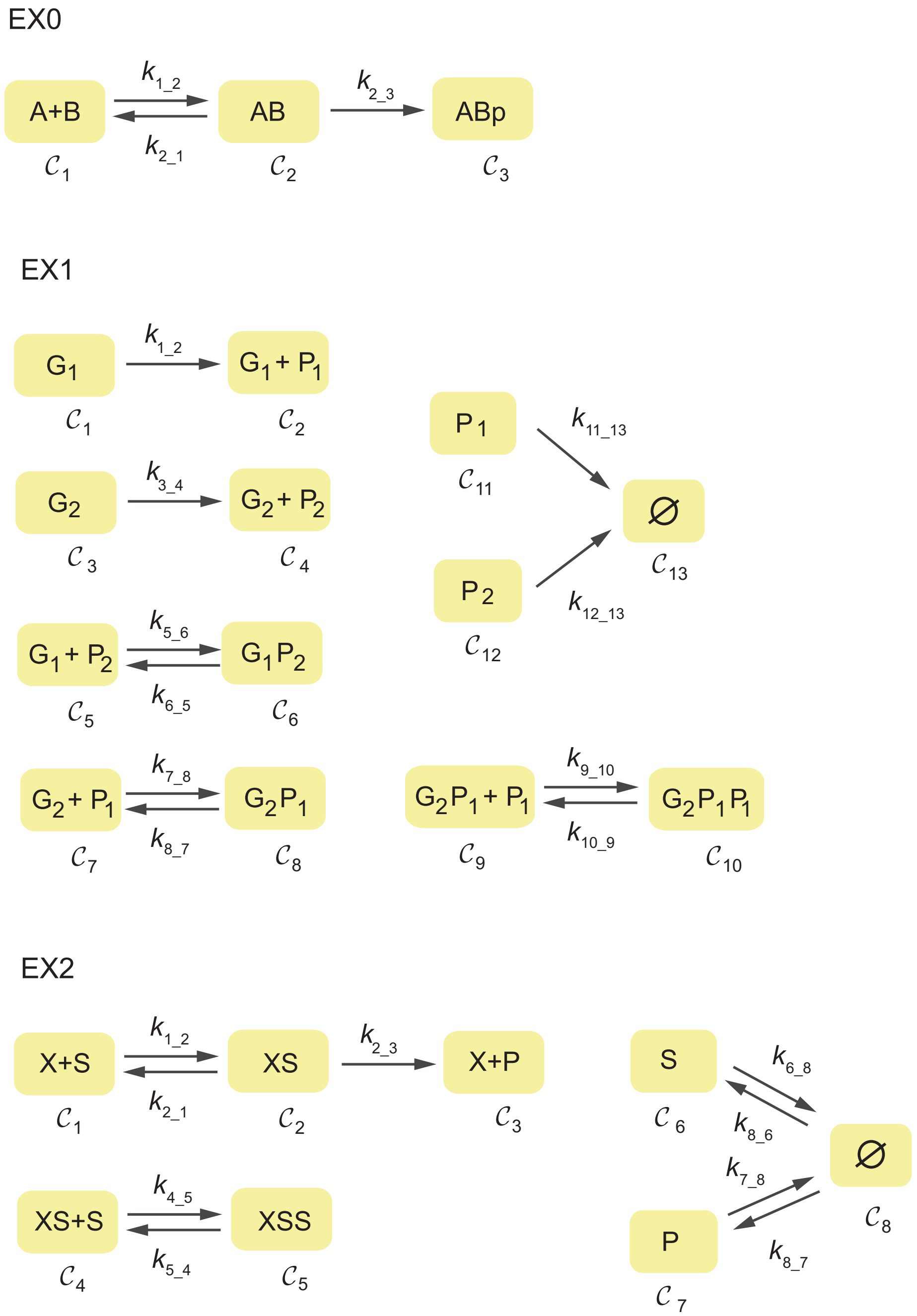

Figure 5: Graph of complexes (C-graph) of example networks EX0, EX1, EX2.

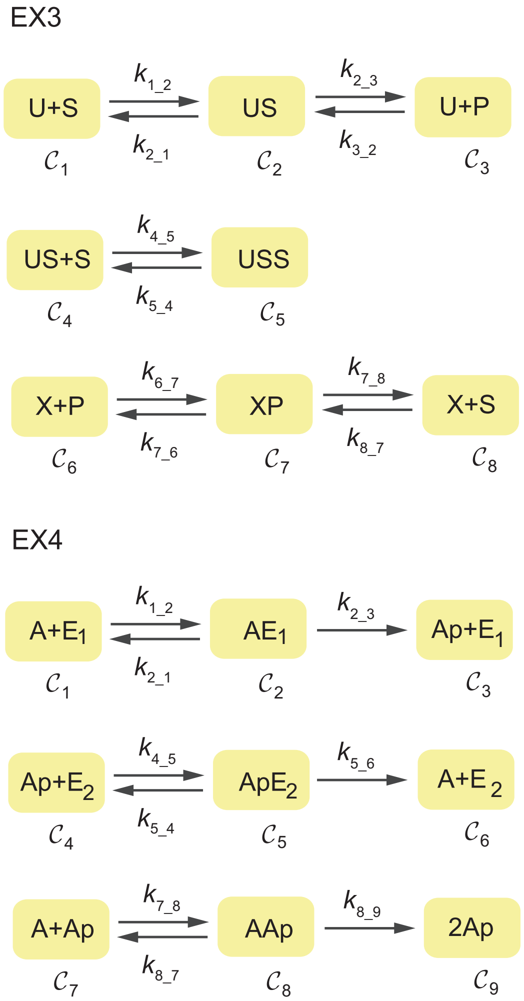

Figure 6: Graph of complexes (C-graph) of example networks EX3, EX4.

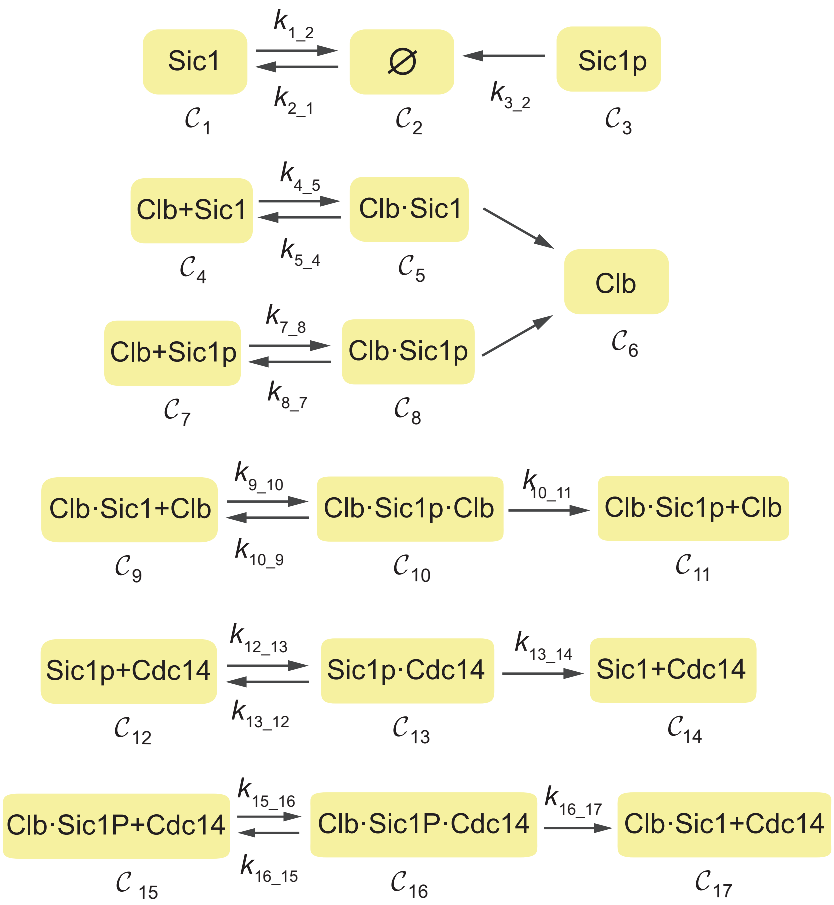

Figure 7: Graph of complexes (C-graph) of example network G1S.

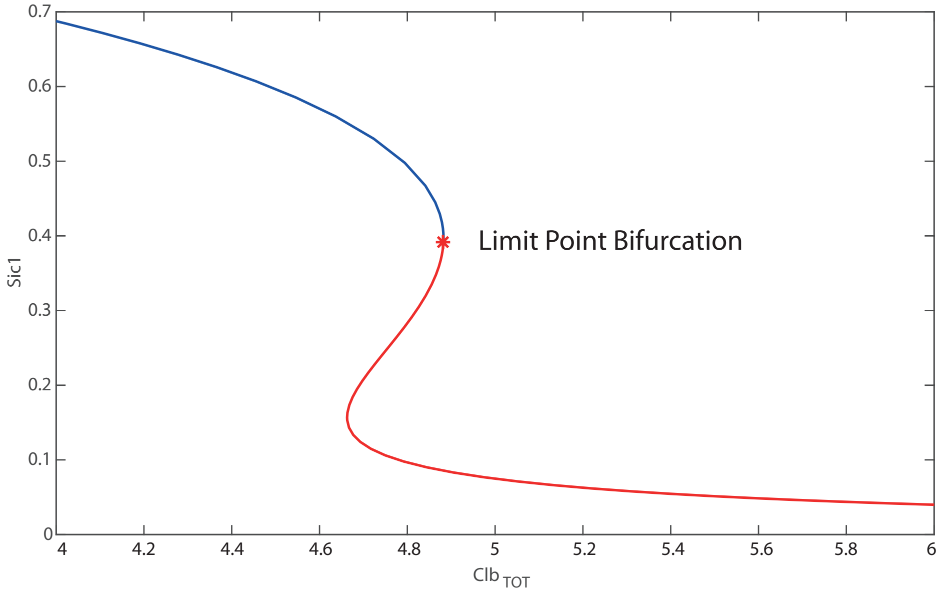

Figure 8: Bifurcation diagram for the G1S network starting from the limit point found.

References

[Otero-Muras (2012)] Otero-Muras I., Banga J. R. and Alonso, A. A. (2012). Characterizing multistationarity regimes in biochemical reaction networks. *PloS ONE* 7(7): e39194.

[Otero-Muras *et al* (2014)] Otero-Muras I., Yordanov P. and Stelling J. (2014). A method for inverse bifurcation of biochemical switches: inferring parameters from dose response curves. *BMC Syst. Biol.* 8: 114.

[Otero-Muras *et al* (2017)] Otero-Muras I., Yordanov P. and Stelling J. (2017). Chemical Reaction Network Theory elucidates sources of multistability in interferon signaling. *PloS Comp. Biol.* 13(4): e1005454.

[Egea *et al* (2014)] Egea J.A., Henriques D., Cokelaer T., Villaverde A.F., MacNamara A., Danciu D. P., Banga J.R. and Saez-Rodrguez J. (2014). MEIGO: an open-source sofware suite based on metaheuristic for global optimization in systems biology and bioinformatics. *BMC Bioinformatics* 15: 136.

[Govaerts *et al* (2012)] Govaerts W., Kuznetsov Y., De Witte A., Meijer H.G.E., Mestrom, W., Riet A. M., Sautois, B. (2012) MATCONT and CL MATCONT: Continuation toolboxes in Matlab.

`https://sourceforge.net/projects/matcont/files/Documentation/ManualSep2012.pdf`

[Otero-Muras *et al* (2009)] Otero-Muras I., Banga J.R. and Alonso A. A. Exploring multiplicity conditions in enzymatic reaction networks. *Biotechnology Progress* 25(3): 619–631.

[Hervagault & Canu (2009)] Hervagault J.F. and Canu S. Bistability and irreversible transitions in a simple substrate cycle. *J. Theor. Biol.* 127(4): 439–49.

[Conradi *et al* (2007)] Conradi C., Flockerzi D., Raisch J. and Stelling J. Subnetwork analysis reveals dynamic features of complex (bio)chemical networks. *Proc. Natl. Acad. Sci. USA* 104(49): 19175–19180.

[Conradi *et al* (2005)] Conradi C., Saez-Rodriguez J., Gilles E. D. and Raisch J. Using chemical reaction network theory to discard a kinetic mechanism hypothesis. *IEE Proceedings - Systems Biology* 152(4): 243–248.

[Saez-Rodriguez *et al* (2008)] Saez-Rodriguez J., Hammerle-Fickinger A., Dalal O. , Klamt S., Gilles E.D., Conradi C. Multistability of signal transduction motifs. *IET Systems Biology* 2(2): 80–93.