Instruction for CADLIVE toolbox

Table of contents

E	Environments	1
	* Install of CADLIVE GUI Network Constructor and CADLIVE Text Editor	1
	* Install of CADLIVE toolbox	1
	* Path for Java	1
	* Path on MATLAB	2
	* Make a folder for a model	2
0	Overview	3
1	Conversion into dynamic model and simulation	6
	1.1 Start	6
	1.2 Load of a regulator-reaction equation data file	7
	1.3 Selection of conversion methods	8
	1.4 Edition of mathematical model data	9
	1.5 Selection of analysis types and input control data	10
	1.6 Input of parameters	12
	1.6.1 CADLIVE_initial.m	19
	1.6.2 CADLIVE_param.m	21
	1.7 Results	24
	1.7.1 Dynamic analysis	24
	1.7.2 Steady-state analysis	30
	1.7.3 S-system	32
2	GA	33
	2.1 Start	34
	2.2 Search parameter setting	36
	2.3 Fitness function setting	37
	2.4 Execution	40
3	TPS	42
	3.1 Start	43
	3.2 QMPS on TPS	44

3.3 Execution	45
4 Execution on command line	
4.1 CADLIVE_simulator	49
4.2 CADLIVE_myGAcommand	52
4.3 CADLIVE_myTPScommand	53
4.4 CADLIVE_DispFigure	55
4.5CADLIVE_DispCumulFreqQMPS, CADLIVE_DispHistFreqQMPS,	
CADLIVE_DispHistCumulFreqQMPS	56
4.6 CADLIVE_optimtool	58
References	59

Environments

Developmental environment

Software	Version	
OS	Windows 7 (64 bit)	
MATLAB	Ver.7.8.0 (R2009a)	
JAVA	Java SE 6 (32bit)	
	Xerces Java Parser-2.7.0	

Path

Application	Path
Java	C:¥Program Files (x86)¥Java¥jre6
CADLIVE Text Editor	C:¥CADLIVE¥
CADLIVE MATLAB (this application)	C:¥CADLIVE_v1

* Install of CADLIVE GUI Network Constructor and CADLIVE Text Editor

These are described at http://www.cadlive.jp/cadlive/editor/download.html

* Install of CADLIVE toolbox

CADLIVE_v1.zip (<u>http://www.cadlive.jp</u>) is unzipped, the folder CADLIVE_v1 is copied to "C:¥".

* Path for Java

Open the file C:\[CADLIVE_v1\[CADLIVE_setenv.m, and change the path as follows if users use Windows 32 bit.

%setenv('JAVA_HOME32','C:¥Program Files (x86)¥Java¥jre6'); setenv('JAVA_HOME32','C:¥Program Files¥Java¥jre6');

* Path on MATLAB

Run MATLAB, and the folder, C:\CADLIVE_v1, is added to the path of MATLAB.



* Make a folder for a model

The files for a model are made in the current folder. Users should make a folder for the model and use this application in the folder.

0 Overview

A goal of systems biology is to construct biological systems at the molecular interaction levels and to understand some design principles underlying the molecular processes. Biochemical networks are the sound bases for pathway analysis and dynamic modeling. The CADLIVE system implements a variety of application modules to perform the network analysis and the dynamic simulations based on biochemical network maps (http://www.cadlive.jp) [1-5]. As an extension of CADLIVE, this standalone application is developed for constructing mathematical models which work on MATLAB.

This application has functions for conversion into dynamic model and simulation, parameter optimization, and system analysis (**Fig.1**). First, the conversion and simulation module automatically converts a biochemical map into a mathematical model and subsequently simulates the dynamic behaviors. Here, the mathematical model is made in MATLAB. Second, the parameter optimization module employs a genetic algorithm (GA) and two-phase search (TPS) method [6] to seek out a global minimum and to estimate many plausible values of the kinetic parameters that determine the dynamic behavior of systems, respectively. The employed GA is derived from the CADLIVE Optimizer [5]. On the other hand, the TPS smoothly combines a random search with an evolutionary algorithm to achieve both nonbiased and high-speed searches for a large parameter space. Finally, system analysis module includes the sensitivity analysis with respect to a single parameter and qausi-multiparameter sensitivity (QMPS) [7]. QMPS measures a robust property of the model to the uncertainty of all kinetic parameters and provides a theoretical or quantitative insight to an understanding of how specific network structures are related to robustness.

These algorithms in CADLIVE greatly facilitate simulating and analyzing a biological system, enhancing the efficiency for the research in systems biology.

This application has mainly three parts; conversion into dynamic model and simulation, and simulations for local parameter optimization by GA and global parameter optimization by TPS.



Fig.1 Overview of functions for CADLIVE toolbox

Here, we illustrate the simulation of a straight reaction chain model. The map of the model is made by the CADLIVE GUI Network Constructor (**Fig.2**). X0 is the constant. X1, X2, X3 and X4 are the time-dependent variables. X1, X2 and X3 don't decompose and X4 decompose. All the reactions occur in the metabolic layer. StraightChainModel.xml is written in the CADLIVE format (**Fig.3**).

Scadlive									
File Edit Window Vi	ew Config Help								
Straight	ChainModel								
ListOfSpecies									
SpeciesName	SpeciesClass	BindingSite	Compartment	MassBalance	TotalAmount	Decomposition	InitialAmount	IsExternal	
X0	metabolite		cytoplasm	off	constant	off	1	false	
X1	metabolite		cytoplasm	off	variable	off	0	false	
X2	metabolite		cytoplasm	off	variable	OTT	0	false	
×4	metabolite		otoplasm	off	variable	on	0	false	
E1	protein		cytoplasm	on	constant	off	1	true	
E2	protein		cytoplasm	on	constant	off	1	true	
E3	protein		cytoplasm	on	constant	off	1	true	
E4	protein		cytoplasm	on	constant	off	1	true	
X0:E1	modifier_complex		cytoplasm	on	variable	on		false	
X1:E2	modifier_complex		cytoplasm	on	variable	on		false	
X2:E3	modifier_complex		cytoplasm	on	variable	on		false	
X3:E4	modifier_complex		cytopiasm	on	variable	on	I	Taise	
ListOfReactions		Setwork Const	ructor - StraightChainM	odel					
ReactionName	ModifierReacti	Modifier						^	2.cytoplasm
#5 F1-0	X0 > X1								Species
#6 E2-0	X1 > X2								
#7 E3-0	X2 > X3								
#8 E4 -0	X3 > X4	Reaction							
								_	
		\leftrightarrow							
		\rightarrow							
			(E1)	E2	E3	E4		
<u> </u>				Ŧ `		÷			
				۰ <u> </u>	o	, d	_	_	
		TransitionState	XO	- X1	• X2	→ X:	3 🛃 🛃 🗡 🗡	(4	
		=							
		VirtualReaction							
		<>							WhiteBox
		(40)							61
									BlackBox
		#>O							5
								•	
				4					
			Synchronize Edit	Compartment	Expand	Zoom Out (-)	100%	Zoom In (+)	

Fig.2 A straight reaction chain model in the CADLIVE GUI Network Constructor

				٢.
😋 🗢 🗸 « sample	🕨 StraightChain 👻 😽	StraightChainの検索		م
整理 ▼ ライブラリに〕	追加 ▼ 共有 ▼ 書き込む	新しいフォルダー	··· ·)
🔶 お気に入り	名前	更新日時	種類	サイ
📔 Dropbox 🗉	🗎 MathParam.txt	10/4/2012 3:49	TXT ファイル	
📤 Google Drive	SSE.txt	10/8/2012 11:1	TXT ファイル	
🚺 ダウンロード	StraightChainModel.xml	10/2/2012 2:24	XML ファイル	
📃 デスクトップ				
📃 最近表示した場所				
-	•			F.
3 個の項目				

Fig.3 Folder prepared for the dynamic model for this application

1 Conversion into dynamic model and simulation

This part is an application program for automatically converting biochemical networks into mathematical models, simulating the model and performing sensitivity analysis.

1.1 Start

Execute "CADLIVE_start" on the MATLAB command window to start the CADLIVE toolbox (**Fig.4**).



Fig.4 Start of the CADLIVE toolbox

1.2 Load of a regulator-reaction equation data file

By clicking the "Launch CADLIVE TEXTEDITOR" button, the CADLIVE Text Editor is opened. The CADLIVE Text Editor helps users describing a network model. The manual of the CADLIVE Text Editor is downloaded at <u>http://www.cadlive.jp/cadlive/editor/download.html</u>.

By clicking the "Select regulator-reaction equation data file" button, users select the data file (CADLIVE format file) for the regulator-reaction equation model from their PC (**Fig.5**). The data file written in the XML format is built by the CADLIVE Text Editor [2], GUI Network Constructor [2,3] or Converter [4].



Fig.5 Selection of the regulator-reaction equation data file

*Once users convert an xml file, the DAE file with a mathematical model can only be loaded if users use the same model in the next.

1.3 Selection of conversion methods

The selected regulator-reaction equation data file is displayed in the window (**Fig.6**), where users choose the conversion methods with respect to gene-protein layer and metabolic layer, respectively. The conversion method can be selected out of the following methods:

Gene-Protein layer:

- \cdot CMA
- TPP_STEADYSTATE_1
- TPP_STEADYSTATE_2
- TPP_RAPID

Metabolic layer:

- GMA
- MM
- SAME_AS_GENE-PROTEIN

The details of description for them are in

http://www.cadlive.jp/cadlive/simulator/Suppl_method_1.pdf.

Select a Regulator-Reaction Equation Data Launch CADLIVE TEXTEDITOR Launch CADLIVE TEXTEDITOR Select regulator-reaction equation data file	File & Regulato	r-Reaction Equations	or
	aunch CADLIVE N	ETWORK CONSTRUCT	R
	C¥Users¥F	OO¥Desktop¥StraightC	hain¥StraightChainModelxml
Gene-Protein Layer: OMA		Metabolic Layer: X0 > X1 X1 > X2 X2 > X3 X3 > X4	GMA GMA MM SAME_AS_GENE-PROTEIN

Fig.6 Selection of conversion methods for the Gene-Protein Layer and Metabolic Layer

1.4 Edition of mathematical model data

By clicking the "Next" button on the "Select a Regulator-Reaction Equation Data File & Regulator-Reaction Equations" window, the "Edit Math Model Data" window is displayed. By clicking the "Edit Parameter" button on the window, the file for a math model (DAE file) is opened on the MATLAB editor so that users can change the model and its parameters (**Fig.7**). This file is saved as MathDAE.txt in the current folder.



Fig.7 Edition for a mathematical model

* The DAE file must be saved before clicking the "Next" button, "Back" button or "Quit" button if users change its content.

* If users manually make the mathematical model from scratch, users can load a dummy xml file and edit the DAE file in this control or the edited DAE file can be loaded in Section 1.2.

1.5 Selection of analysis types and input control data

By clicking the "Next" button on the "Edit Math Model Data" window, the "Select Analysis Type & Set Control Data for Simulation" window is displayed, where users select the analytical type for a mathematical model and input conditions of the type (**Fig.8**).

Analysis type

Users can choose either "Dynamic Analysis" or "Steady-state Analysis". "Dynamic Analysis" simulates the time evolution of the values by calculating differential and algebraic equations, and "Steady-state Analysis" calculates the values at steady state by solving algebraic equations. The checkbox of "Parameter survey" determines if the simulation surveys the parameter space. The checkbox of "Use S-system", which employs S-system differential equations, appears only under the condition that the simulation has been solved before. It never appears when the TPP is selected as the conversion method.

Control data for simulation

In "Dynamic Analysis", users set "Solver Type (tolerance)" and "Set time span and time step-size". "relative" is a relative tolerance and "absolute" is an absolute tolerance.

In "Steady-state Analysis", users set "Set values for Newton-Raphson Method". Sensitivities with respect to a change in each parameter are simultaneously calculated when "Steady-state Analysis".

ect Analysis Type	& Set Control Data for Sin	nulation		
F	haha			
Equation type				
Analysis type	Dynamic Analysis	•		
🗌 Use S-system				
🔲 Parameter surv	еу			
- Solver Type (tole	rance)			
💿 Runge-Kutta	a (Adaptive step-size)[ode45]	(relative:	1e-012)	
NDF[ode15s]	(relative:	1e-012 , absolute:	1e-012)
Set time span and	l time step-size.	ī		
Start time	0			
End time	1			
(Initial)time step	-size 0.01			
Monitoring inter	val 0.01			
— Set values for Ne Maximum trial ti	wton-Raphson Method	20		
Tolerance for co	onvergence of functions	1e-012		
Tolerance for co	onvergence of variables	1e-012		
Ratio of changir	ng parameters	1.1		
Change width fo	r calc.sensitivity(STD)	0.001		
- Other				
G-value	1			
Y default value	0.01			
Quit			Back	K Next

Fig.8 Selection of Analysis Type and Set of Control Data for Simulation

1.6 Input of parameters

By clicking the "Next" button on the "Select Analysis Type & Set Control Data for Simulation" window, the "Set Parameters and Initial Values" window is displayed (**Fig.9**), where users input the (initial) values of kinetic parameters and the variables for the mathematical model.

User Functions

Users select "usr_fvec" or "usr_fjac" button. "usr_fvec" (CADLIVE_usr_fvec.m) is the function for the differential equations (**Fig.10**). "usr_fjac" (CADLIVE_usr_fjac.m) is for the Jacobian function (**Fig.11**). By clicking the "Edit User Function" button, the file selected is opened and can be edited on the MATLAB editor.

Initial Values

Users select the "Initial Value" or "Parameters" button. "Initial Value" (CADLIVE_initial.m) has information of the control data and dependent variables (**Fig.12**). "Parameters" (CADLIVE_param.m) has information of the (kinetic) constant parameters (**Fig.13**). By clicking the "Edit Initial Values" button, the file selected is opened on the MATLAB editor.

Merge File

The Merge File helps users setting the parameters by copying or merging the existing data, which greatly reduces laborious parameter setting. By clicking the "Execute Merge" button, the existing data of a parameter file is copied to CADLIVE_initial.m and CADLIVE_param.m (**Fig.14**). The parameter file can be downloaded as "MathParam.txt" (**Fig.15**). By checking the "Update Blank Only" button, the only blank data of initial values and parameters are added to CADLIVE_initial.m and CADLIVE_param.m as uploaded data. By checking the "Update All" button, all the data are input.

Here, users mainly edit CADLIVE_initial.m and CADLIVE_param.m. The meanings for each variable are described in section 1.6.1 and 1.6.2. If users edit the differential equations in "usr_fvec" and/or add new variables, the variables need to be added to CADLIVE_initial.m and CADLIVE_param.m. The entire edition needs to be saved before next operation.

User Functions	- Initail Values	Merge File
Edit User Function	Edit Initial Values	
ie usr_tvec © usr_fjac	 Initial Value Parameters 	 Update Blank Unly Update All
	<u></u>	

Fig.9 Control of setting parameters

ファイル(E) 編集(E) デキスト(I) 移動(G) セル(G) ツール(Q) デパッグ(B) デスクトップ(Q) ウィンドウ(W) ヘルプ(H) * * * * * ご ご ご (I)
Image: Second secon
Image: Second
1 k TITLE:StraightChainModel 2 % INFO:StraightChainModel 3 % CONVERSION TYPE 4 % GENE-PROTEIN:NONE 5 METABOLIC:NM 8 □function [fvec] = CADLIVE_usr_fvec(y, Gene, p) 7 constantPlayer_len = length(p.constantPlayer); 8 □ 9 constantPlayer(i) = p.constantPlayer(i).value;
2 % INFO:StraightChainModel 3 % CONVERSION TYPE 4 % GERE-PROTEIN:NONE 5 % METABOLIC:NM 8 □function [fvec] = CADLIVE_usr_fvec(y, Gene, p) 7 - constantPlayer_len = length(p.constantPlayer); 8 - □ for i=1:1:constantPlayer_len 9 - constantPlayer(i) = p.constantPlayer(i).value;
<pre>3 * COMPENSION TYPE 4 % GENE-PROTEIN:NONE 5 % METABOLIC:NM 8 □ function [fvec] = CADLIVE_usr_fvec(y, Gene, p) 7 - constantPlayer_len = length(p.constantPlayer); 8 - □ for i=1:1:constantPlayer_len 9 - constantPlayer(i) = p.constantPlayer(i).value;</pre>
<pre>4</pre>
6 = function [fvec] = CADLIVE_usr_fvec(y, Gene, p) 7 - constantPlayer_len = length(p.constantPlayer); 8 - for i=1:1:constantPlayer_len 9 - constantPlayer(i) = p.constantPlayer(i).value;
7 - constantPlayer_len = length(p.constantPlayer); 8 - for i=1:1:constantPlayer_len 9 - constantPlayer(i) = p.constantPlayer(i).value;
8 - constantPlayer_len 9 - constantPlayer(i) = p.constantPlayer(i).value;
9 - constantPlayer(i) = p.constantPlayer(i).value;
$11 - \bigcup_{i=1}^{n} e_{i} = 1 e_{i} g_{i} (n_{i}) e_{i} (n_{i}) = 0$
13 - 0(1) = 0.0(1).value:
14 – end
15 - Kmich_len = length(p.Kmich);
18 - c for i=1:1:Kmich_len
17 - Kmich(i) = p.Kmich(i).value;
18 – - end
$\int_{0}^{13} \int_{0}^{13} fvac(-1) = 0(1) + constant Blavar(0) + constant Blavar(1) / (Vmich(1) + constant Blavar(1)) = 0(2) + constant Blavar(2) + v(1) / (Vmich(2) + v(1)) + (1) + ($
$\frac{1}{21 - \frac{1}{2}} = \frac{1}{2} \left(\frac{1}{2} \right) \left(\frac{1}{2} \right)$
22 - fvec(3) = Q(3)*constantPlayer(4)*y(2)/(Kmich(3) + y(2)) - Q(4)*constantPlayer(5)*y(3)/(Kmich(4) + y(3));
23 - fvec(4) = Q(4)*constantPlayer(5)*y(3)/(Kmich(4) + y(3)) - Q(5)*y(4)/(Kmich(5) + y(4));
24 - Lend
25
28
CADLIVE_usr_fvec

Fig.10 CADLIVE_usr_fvec.m

The "fvec" indicates the differential equation for a dependent variable y.

アディター - C:¥Users¥F00¥Desktop¥StraightChain¥CADLIVE_usr_fjac.m	
ファイル(E) 編集(E) テキスト(I) 移動(G) セル(C) ツール(Q) デバッグ(B) デスクトップ(D) ウィンドウ(W) ヘルプ(H)	X 5 K
: 🖺 😂 🖩 👗 🤊 🤭 😓 🖅 🔸 🆛 🔶 🈥 🖌 🖬 🖛 🗰 🏚 📾 🖬 🕼 🖓 スタック(Ϗ): ベース 🖃 🥠 🖽 🖽 🖽	880
+= μ= − 1.0 + ÷ 1.1 × % % % 0.	
1 % TITLE:StraightChainModel	
2 % INFO:StraightChainModel	
3 % CONVERSION TYPE	
4 % GENE-PROTEIN:NONE	
5 % MelABULUSAM	
Figure 1 - Start -	
8 - constantPlayer len = length(p.constantPlayer):	
9 - for i=1:1:constantPlayer len	
10 - constantPlayer(i) = p.constantPlayer(i).value;	—
11 - end	
12 - Q_len = length(p.Q);	
13 - 🗇 for i=1:1:0_len	
14 - Q(i) = p.Q(i).value;	-
15 end	
16 - Kmich_len = length(p-Kmich);	
17 - Tor I-I.I.Kmich_len	_
19 - ond	
20	
21 - fiac(1, 1) = -(Q(2)*constantPlayer(3)*(Kmich(2) + y(1)) - Q(2)*constantPlayer(3)*y(1))/((Kmich(2) + y(1))^2	2);
22 - fjac(2, 1) = (Q(2)*constantPlayer(3)*(Kmich(2) + y(1)) - Q(2)*constantPlayer(3)*y(1))/((Kmich(2) + y(1))^2));
23 - fjac(2, 2) = -(Q(3)*constantPlayer(4)*(Kmich(3) + y(2)) - Q(3)*constantPlayer(4)*y(2))/((Kmich(3) + y(2))^2	?);
24 - fjac(3, 2) = (Q(3)*constantPlayer(4)*(Kmich(3) + y(2)) - Q(3)*constantPlayer(4)*y(2))/((Kmich(3) + y(2))^2)	/; []
25 - fjac(3, 3) = -(Q(4)*constantPlayer(5)*(Kmich(4) + y(3)) - Q(4)*constantPlayer(5)*y(3))/((Kmich(4) + y(3))^2	!);
28 - fjac(4, 3) = (Q(4)*constantPlayer(5)*(Kmich(4) + y(3)) - Q(4)*constantPlayer(5)*y(3))/((Kmich(4) + y(3))^2)	6
$\frac{2}{2} - \frac{1}{2} - \frac{1}$	
28	
CADLIVE_usr_fjac	上書き .::

Fig.11 CADLIVE_usr_fjac.m

The "fjac" indicates the partial derivative for "fvec".

🧭 エディ	ター - C:¥Users¥F00¥Desktop¥QMPStest¥StraightChain_dynamic¥CADLIVE_initial.m 🛛 💼 📧
ファイル	(E) 編集(E) テキスト(I) 移動(G) セル(C) ツール(Q) デバッグ(B) デスクトップ(D) » 🛚 🗗 🗙
: 🛍 🖨	■ & ■ ■ ウ ♥ 🌭 🖅 - 🗛 🗣 ➡ 🍂 💽 - 🗟 🛠 🖷 ■ 💷 मी 細 fx 🔲 📼
*=	$ - 1.0 + \dot{+} 1.1 \times \%^{\%}_{+} \%^{\psi}_{-} $
1	function [CADLIVE_CTL Y_START]=CADLIVE_initial()
2	
3	%=control data
4 -	CADLIVE_CTL.N_VAR=4;
5 -	CADLIVE_CTL.N_ALGEBR=0;
6	
7	%==solver
8 -	CADLIVE_CTL.SOLVER=3;
9 -	CADLIVE_CTL.P_SURVEY=0;
10 -	CADLIVE_CTL.RK_EPS=1.000000e-012;
11 -	CADLIVE_CTL.NDF_RTOL=1.000000e-012;
12 -	CADLIVE_CTL.NDF_ATOL=1.000000e-006;
13	
14	%==time span and time step-size
15 -	CADLIVE_CTL.T_START=0.000000;
16 -	CADLIVE_CTL.T_END=1.000000;
17 -	CADLIVE_CTL.DELTA_T=0.010000;
18 -	CADLIVE_CTL.DELTA_M=0.020000;
19	
20	%==Newton-Raphson Method
21 -	CADLIVE_CTL.NR_TRIAL=20;
22 -	CADLIVE_CTL.NR_TOL_F=1.000000e-012;
23 -	CADLIVE_CTL.NR_TOL_X=1.000000e-012;
24 -	CADLIVE_CTL.NR_RATIO=1.100000;
25 —	CADLIVE_CTL.NR_SENS_CW=0.001000;
26	
27	%==Other
28 -	CADLIVE_CTL.G_VALUE=1.000000;
29 -	CADLIVE_CTL.Y_DEFAULT=1.000000e-002;
30	
31	%=Initial parameters
32 -	Y_START(1).value = 0.0000e+000; % X1.cyt
33 -	Y_START(1).tag = 'X1.cyt';
34 -	Y_START(2).value = 0.0000e+000; % X2.cyt
35 -	Y_START(2).tag = 'X2.cyt';
36 -	Y_START(3).value = 0.0000e+000; % X3.cyt
37 -	Y_START(3).tag = 'X3.cyt';
38 -	Y_START(4).value = 0.0000e+000; % X4.cyt
39 -	Y_START(4).tag = 'X4.cyt';
40	
41	%=solver parameter
42 -	CADLIVE_CIL.MASS=eye(CADLIVE_CTL.N_VAR);
43 -	tor i=1:CADLIVE_CTL.N_ALGEBR
44 -	CADLIVE_CIL.MASS(i,i)=0;
45 -	- end
46	
4/	»=odelos wEvents parameter

Fig.12 CADLIVE_initial.m

The control data for simulation (section 1.5) and initial values of dependent variables are set.

🦻 エディ	(ター - C:¥Users¥F00¥Desktop¥StraightChain¥CADLIVE_param.m			• ×
ファイル	<i>ν</i> (E) 編集(E) テキスト(T) 移動(G) セル(C) ツール(Q) デバッグ(B) デスクトッフ	プ(<u>D</u>)	»	X 5 K
1 🗂 🗃		1 ベース	- <i>fx</i>	
: += _ =	$ -1.0$ + $ \div 1.1$ × $ \%, \%$ 0			
1	Function [param, event]=CADLIVE param()			
2				Â
3	% T EVENT			
4 -				
5 -	event(1).index = 0;			
6 -	event(1).time = 0.000000;			
7 -	event(1).value = 0.000000;			
8 -	event(2).name = 'time_end';			
9 -	event(2).index = 0;			
10 -	event(2).time = 1.000000;			
11 -	event(2).value = 0.000000;			
12				
13	% PARAMETER			
14 -	param.constantPlayer(1).value = 1.0000e+000;			
15 -	param.constantPlayer(1).num_survey = U;			=
16 -	param.constantPlayer(1).d_r_s = 101;			
17 -	param.constantPlayer(I).tag = XU.cyt ;			
18 -	param.constantPlayer(2).value = 1.0000e+000;			
18 -	param.constantPlayer(2).num_survey = 0,			
20 -	param constantFlaver(2) tag = 'F1 cvt':			
22 -	param.constantPlayer(2).value = 1.0000e+000:			
22 -	param.constantPlayer(3).num survey = 0:			
24 -	param.constantPlayer(3).d r s = 'D':			
25 -	param.constantPlayer(3).tag = 'E2.cvt':			
26 -	param.constantPlayer(4).value = 1.0000e+000;			
27 -	param.constantPlayer(4).num_survey = 0;			
28 -	param.constantPlayer(4).d_r_s = 'D';			
29 -	param.constantPlayer(4).tag = 'E3.cyt';			
30 -	param.constantPlayer(5).value = 1.0000e+000;			
31 -	param.constantPlayer(5).num_survey = 0;			
32 -	param.constantPlayer(5).d_r_s = 'D';			
33 -	param.constantPlayer(5).tag = 'E4.cyt';			
34	% param.Q(1).value = is not value;			
35 —	param.Q(1).num_survey = 0;			
36 -	param.Q(1).d_r_s = 'D';			
37 -	<pre>param.Q(1).tag = 'reaction_rate_constant_E1.cyt_XU.cyt_MM';</pre>			
38	% param.Q(2).value = is not value;			
39 -	param.w(2).num_survey = 0;			
40 -	param.u(2).d_r_s = 'D';			
41 -	<pre>param.w(z).tag = reaction_rate_constant_cz.cyt_At.Cyt_MM ;</pre>			
42	$p_{aram}(q_{3})$, $v_{arue} = 1$ s not varue,			
43 -	param.0(3).drs = 'D':			
45 -	param.Q(3).tag = 'reaction rate constant E8.cvt X9.cvt WM':			
46	% param.Q(4).value = is not value:			
	A/() 0.			T
	CADLIVE_param	行 1	列 1	上書き …

Fig.13 CADLIVE_param.m

The kinetic parameters and the events that change the parameter values in a given time are set.

🛃 Please select th	e parameter file to be merged.		×
ファイルの場所(1):	DirectedModel	← 🗈 💣 📰▼	
œ.	名前	更新日時	種類
最近表示した場所	📄 dbg_log.txt	2012/08/28 13:09	TXT ファイル
	MathDAE.txt	2012/08/28 13:09	TXT ファイル
-112 bit	MathParam.txt	2012/08/28 13:14	TXT ファイル
テスクトッノ	📄 MathUsage.txt	2012/08/28 13:09	TXT ファイル
ライブラリ	📄 sys_log.txt	2012/08/28 13:09	TXT ファイル
ネットワーク	•		Þ
	ファイル名(N): MathParam.txt	•	開((<u>0</u>)
	ファイルの種類(I): (*.txt)	_	キャンセル

Fig.14 Select "MathParam.txt"

Http://kurata2	3.bia. kyutech.ac.jp /Life/index.html	CADLIVE Simulator ×
	Simulator Start from database Start fro Math model data Dire	om scratch Start from local PC file Regulator-reaction equations
CADLIVE Simulator user : cadive Simulator Database	Results Succeeded !! Back Print Select ana. Resistration Down Save for input Graph Quit Straight Chain Noocer	nload Regulator-reaction eqs. Math model User funcs
maintenance	======================================	Download
File interface	###### Thu Oct 4 15:48:33 2012 ######	Select a file which you want to download to your PC.
	Solver No. : 3 Param Survey : 0 Start Time : 0.000E+00 End Time : 1.000E+00 (initial)Time Step : 1.000E+02 Monitoring interval : 2.000E+02 NDF_ATOL : 1.000E+02 NDF_ATOL : 1.000E+08 G=Value : 1.000E+00	Close StraightChainModel StraightChainModel
	######## Starting Y Values ###### y[1]= 0.0000e+00 #X1.cvt y[2]= 0.00000e+00 #X2.cvt y[3]= 0.0000e+00 #X3.cvt y[4]= 0.0000e+00 #X4.cvt	Model data file(DAE File) Download Control data file Download Parameter file Download
Exit	######### Parameters ####### constantPlayer[1] = 1.00000+00 #X0.cyt 0[1] = 5.0000e-01 #reaction_rate_cons	Parameter file (S-system) Download User func file Download User func file (S-system) Download Flux file
		User func file (Sensitivity) Download Results File (Dynamic) Download Results file (Steady state) Download All above files (all tgp)

Fig.15 "MathParam.txt" in the CADLIVE Dynamic Simulator (Web application)

1.6.1 CADLIVE_initial.m

This file is written as control data ("CADLIVE_CTL") for simulation, and information ("Y_START") for the dependent variables. "CADLIVE_CTL" is automatically input by editing parameters in section 1.5. Users usually edit the "value" of "Y_START" in this file.

Field	Meanings					
N_VAR	Number of variables (all) ; Integer value > 0					
N_ALGEBR	Number of variables (algebraic equations) ; Integer value ≥ 0					
SOLVER	Solver number; 2: ode45, 3: ode15, 11: steady-state, 12: S-system					
P_SURVEY	Parameter survey; 1: yes, 0: no					
RK_EPS	Relative error tolerance for Runge-Kutta; Real value > 0					
NDF_RTOL	Relative error tolerance for NDF; Real value > 0					
NDF_ATOL	Absolute error tolerance for NDF; Real value > 0					
T_START	Start time; Real value ≥ 0					
T_END	End time; Real value > 0					
DELTA_T	Initial time step size; Real value > 0					
DELTA_M	Monitoring interval; Real value > 0					
NR_TRIAL	Maximum trial times for "Steady-state analysis"					
	(Newton-Raphson) ; Integer value > 0					
NR_TOL_F	Tolerance for convergence of functions for "Steady-state analysis"					
	(Newton-Raphson); Real value > 0					
NR_TOL_X	Tolerance for convergence of variables for "Steady-state analysis"					
	(Newton-Raphson); Real value > 0					
NR_RATIO	Ratio of changing parameters for "Steady-state analysis"					
	(Newton-Raphson); Real value > 0					
NR_SENS_CW	Change width calculation sensitivity for "Steady-state analysis";					
	Real value > 0					
G_VALUE	Value for one molecule concentration in a cell;					
	Real value > 0					
Y_DEFAULT	Default values for y (molecular concentrations) ;					
	Real value > 0					
MASS	Mass matrix for "Dynamic analysis";					
	0: differential equation, 1: algebraic equation					

[Structure variables "CADLIVE_CTL"]

[Structure variables "Y_START"]

Field	Meanings
value	Initial value; Real value ≥ 0
tag	Name of variable distinguished location;
	(name).(location)

variable	Meanings
MAXSIMTIME	Limit actual time (second); default: 5 minutes
Т0	Start time for ode15s options (Events)

If the simulation is stiff, the calculation takes a long time or doesn't finish. When the actual time of the calculation is MAXSIMTIME, the calculation forcibly finishes even if stiff simulation.

1.6.2 CADLIVE_param.m

This file is written as information for events ("event") and (kinetic) constant parameters ("param"). The events indicate changes in parameter values in a given time. That can indicate, for example, environmental changes to stimuli.

Field	Meanings				
name	Field name of "param", i.e. constantPlayer, Kb, Q				
	etc., for events.				
	time_start and time_end have to be fixed at first and				
	last index, respectively.				
index	Index for "param.(name)"				
time	Time occurred an event; Real value ≥ 0				
value	Value for "param.(name)" on the time;				
	Real value ≥ 0				

[Structure variables "event"]

*Events need to be registered in the ascending order of time between time_start and time_end.

Example for events:

% T_EVENT

```
event(1).name = 'time_start';
event(1).index = 0;
event(1).time = 0.000000;
event(1).value = 0.000000;
event(2).name = 'kx';
event(2).index = 2;
event(2).time = 60.000000;
event(2).value = 150.000000;
event(3).name = 'kp';
event(3).index = 1;
event(3).time = 60.000000;
event(3).value = 80.000000;
event(4).name = 'time_end';
event(4).index = 0;
event(4).time = 100.000000;
event(4).value = 0.000000;
```

[Structure variable "param"]

Field	Meanings
constantPlayer	Independent variable
Ka	Binding association rate constant
Kx	Reaction rate constant
Kb	Binding constant
Кр	Translation rate constant
Kpd	Decomposition rate constant
Km	Transcription rate constant
Kmd	Decomposition rate constant
Ktr	Transport rate constant
Kxg	Reaction rate constant
F Power coefficient (Target variable)	
Q	Reaction rate constant
Kmich	Michaelis constant

*These fields are defined by structure variables. For CMA and TPP, ka, kd, kx, Kb, kp, kpd, km, kmd and ktr are utilized. For GMA, kxg and f are utilized. For MM, Q and Kmich are utilized.

Field	Meanings					
value	Value for the constant; Real value ≥ 0					
num_survey	Number of parameter survey; Integer ≥ 0					
d_r_s	D: parameter survey by arithmetic series					
	R: parameter survey by geometric series					
	S: parameter search for GA or TPS					
tag	Name of the constant					
change_val	Change for parameter survey; Real value ≥ 0					
upperBound	Upper bound of parameter search for GA in section					
	2 or TPS in section 3; Real value ≥ 0					
lowerBound	Lower bound of parameter search for GA in section					
	2 or TPS in section 3; Real value ≥ 0					

[Structure variable "param.(name)"]

*The Fields, change_val, upperBound and lowerBound, are written by users.

¥ Parameter survey ¥

When checking the checkbox of parameter survey on the "Select Analysis Type & Set Control Data for Simulation" window, "num_survey" and "change_val" need to be more than zero at least one parameter. For example, parameter survey is executed as follows.

1) Parameter survey by arithmetic series

When the fields of "param.(name)" are $r_d = D$; num_survey=3; change_val = 1-e3; value=1e-3; the three cases for {1e-3, 2e-3, 3e-3} are calculated.

2) Parameter survey by geometric series

When the fields of "param.(name)" are $r_d_s=R'$; num_survey=3; change_val = 1-e3; value=1e-3; , the three cases for {1e-3, 1e-6, 1e-9} are calculated.

1.7 Results

By clicking the "Execute" button on the "Set Parameters and Initial Values" window, the simulation starts and the "Results" window is displayed if the simulation is successful. If the simulation fails, the error message is displayed on the MATLAB command window.

The window for the result depends on analysis types; dynamic analysis, steady-state analysis and S-system.

1.7.1 Dynamic analysis

The window for "Dynamic analysis" is displayed as shown in **Fig.16**. The table indicates the temporal data of the simulation. By clicking the "Show graph" button, the graphs of the simulation are displayed (**Fig.17**). The variables to be displayed at the graph can be set at CADLIVE_DispFigure.m, where users input the indices of variables and graph type. By clicking the "setting" button, CADLIVE_DispFigure.m is opened (**Fig.18**). The "Reset" button sets the setting for the graph to default. CADLIVE_DispFigure.m is copied in the current folder. By clicking the "Export CSV" button, the result of the simulation is saved as a CSV file (**Fig.19**). By clicking the "Export MATLAB" button, the result of the simulation is saved as a mat file. By clicking the "Save for input" button, the concentrations at the final time of the simulation are saved as "CADLIVE_saveInitial.m", which can be used as the initial values for the next simulation and is written the same format as "CADLIVE_initial.m". By clicking the "QMPS" button, "SetQMPS" is opened (**Fig.20**), where users can calculate QMPS using the simulation results.

sults Show graph Settir	ng Reset]	Show Make_S_	Param Result	Show Eigen Values Export CSV Export MATL
PARAMETER SURVE	EY No.	-	Save for in	put	QMPS
Results	y[1].X1.cyt	y[2].X2.cyt	y[3].X3.cyt	y[4].X4.cyt	
0.000E+000	0	0	0	0	
2.000E-002	0.0046	0.0024	0.0014	8.1908e-04	
4.000E-002	0.0067	0.0040	0.0029	0.0021	
6.000E-002	0.0081	0.0051	0.0039	0.0031	
8.000E-002	0.0092	0.0059	0.0046	0.0038	
1.000E-001	0.0100	0.0065	0.0052	0.0044	
1.200E-001	0.0107	0.0070	0.0057	0.0049	
1.400E-001	0.0113	0.0074	0.0060	0.0053	
1.600E-001	0.0117	0.0078	0.0064	0.0056	
1.800E-001	0.0121	0.0081	0.0066	0.0058	
	0.0405	0.0002	0.0069	0.0061	

Fig.16 Results window for dynamic analysis





Fig.17 Graphs for simulations Top: graph for type 1, Bottom: graph for type 2



Fig.18 CADLIVE_DispFigure.m

Users edit the "select" and "type" below the "User edit". The "select" indicates the indices of y to be displayed. The "type" indicates the number of graph type. 1 and 2 of "type" indicate the top and bottom figures in **Fig.17**, respectively.

	- 10 - (21 -	-	re	sult.csv - M	icrosoft Exc	el			• **
רידר	ルーホーム	挿入	ページ レイアウ	ト数式	データ	校閲	表示	۵ 🤇) — @ X
	🛛 🐰 М	S Pゴシック	- 11 -	= = =	計 標準	ŧ ,	A	₽━挿入・	$\Sigma = \frac{A}{Z} \gamma =$
Btb/c	+(+ ₽ - B	<i>I</i> <u>U</u> -	A A		-a 🕎	- % ,		🚰 削除 🝷	💽 • 🖓 •
第日ワイン	10 🝼 📋] - 💩 - 🛛	<u>-</u> <u>∓</u> -	🛊 🛊 🚿	··· *.0	.00 ->.0	~	🏥 書式 🗸	2-
クリップ	新一ド 回	フォント	G.	配置	En de	数値 「	ā.	セル	編集
	A1	•	🥚 f _x	time					~
	A	В	С	D	E	F		G	H
1	time	X1.cyt	X2.cyt	X3.cyt	X4.cyt				
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	I			
3	2.00E-02	4.60E-03	2.39E-03	1.42E-03	8.19E-04				
4	4.00E-02	6.72E-03	4.00E-03	2.85E-03	2.11 E-03				
5	6.00E-02	8.14E-03	5.09E-03	3.87E-03	3.09E-03				
6	8.00E-02	9.20E-03	5.90E-03	4.62E-03	3.84E-03				
7	1.00E-01	1.00E-02	6.53E-03	5.21 E-03	4.43E-03				
8	1.20E-01	1.07E-02	7.03E-03	5.67E-03	4.89E-03				
9	1.40E-01	1.13E-02	7.44E-03	6.05E-03	5.27E-03				
10	1.60E-01	1.17E-02	7.78E-03	6.36E-03	5.58E-03				
11	1.80E-01	1.21 E-02	8.06E-03	6.62E-03	5.84E-03				
12	2.00E-01	1.25E-02	8.30E-03	6.84E-03	6.06E-03				
13	2.20E-01	1.27E-02	8.51 E-03	7.03E-03	6.24E-03				
14	2.40E-01	1.30E-02	8.68E-03	7.19E-03	6.40E-03				
15	2.60E-01	1.32E-02	8.84E-03	7.33E-03	6.54E-03				
16	2.80E-01	1.34E-02	8.97E-03	7.44E-03	6.65E-03				
17	3.00E-01	1.36E-02	9.08E-03	7.55E-03	6.75E-03				
18	3.20E-01	1.37E-02	9.18E-03	7.64E-03	6.84E-03				
19	3.40E-01	1.38E-02	9.27E-03	7.71 E-03	6.91 E-03				
H A	▶ ▶ result		0.055.00	7 705 00	n na fili	•			
コマン	r						100% 🤅	∋—_⊽	

Fig.19 Excel data exported

Variables exported by "Export MAILAB"	[V	/ariables	exported	by	"Export	MATL	AB"
---------------------------------------	----	-----------	----------	----	---------	------	-----

Variable	Meanings
Eigen	Eigen values
TimeStep	Time
tag	Name of the variables
у	Dynamics of each dependent variable (each column
	indicates each dependent variable)

QMPS

The "Delta" indicates the change in parameters when calculating the QMPS. By clicking the "Target function" button, the file CADLIVE_setTarget.m is opened (**Fig.21**) if the file exists in the current folder. If the file does not exist, the file is newly made and opened. The target function is defined as a particular function such as oscillation, for sensitivity in the biological system. Users write a target function below "%input a target function for QMPS below" according to the MATLAB language. By clicking the "Execute" button, the QMPS is calculated. The result is displayed in "QMPS" of the "SetQMPS" window (**Fig.22**). If some target functions are set, users can select the target number of "Target No." and display its QMPS.

sults Show graph Setti	ng Reset)	Show Make_S_Para	m Result Sho	w Eigen Values	Export CSV	Export MATLAB
PARAMETER SURV	EY No		📣 SetQMPS				OMPS
Results	v[1] Y1 ort	v[2] ¥2 o/t	Edit QMPS		\neg		
0.000E±000	y[1].A1.Gyc	y[z].^z.cyc	Delta:	0.001			
2.000F-002	0.0046	0.0024	Targ	et function			
4.000E-002	0.0067	0.002					=
6.000E-002	0.0081	0.0051	Result				
8.000E-002	0.0092	0.0059	Target	No.: 1 ▼			
1.000E-001	0.0100	0.0065	QMPS =	0			
1.200E-001	0.0107	0.0070					
1.400E-001	0.0113	0.0074	Quit	E	recute		
1.600E-001	0.0117	0.0078	0.0064	0.0056			
1.800E-001	0.0121	0.0081	0.0066	0.0058			
2 0005 001	0.0125	0.0093	8300.0	0.0061			-

Fig.20 SetQMPS window

ゴディター - C:¥Users¥F00¥Desktop¥sample¥StraightChain¥CADLIVE_setTarget.m		x
ファイル(E) 編集(E) テキスト(I) 移動(G) セル(C) ツール(O) デバッグ(B) デスクトップ(D)	ж ч	× ×
: 🔁 🚔 📓 & 🐂 🖷 🍠 🕫 🌺 🆅 - 🏘 🗢 🔶 🈥 💽 - 🗟 🗶 🖷 🖷 💷 🕮	- fx 🔲	•
+≣ ⊑ - 1.0 + ÷ 1.1 × ‰ ‰ ↓ 0.		
1function target = CADLIVE_setTarget(y,t)		
2 - % target is a vector and values of target functions.		
4 % y(row,col) is dynamics of each dependent variable.		
5 -% row is index of value according to t and col is index of dependent variable	э.	
7 Winput a target function for QMPS below.		
8 - target(1)=y(end,1);		
9 - target(2)=y(end,2);		
11		
12		
CADLIVE setTarget 行 12 多		<u></u>
	1-80	

Fig.21 CADLIVE_setTarget.m

Show graph Setting	Reset		Show Make_S_Pa	ram Result	Show Eigen \	/alues	Export CS	V Export MATLAE
PARAMETER SURVEY	/ No.		etQMPS — Edit QMPS———		• •			QMPS
Results	y[1].X1.cyt	/[2].>	Delta:	0.001				
0.000E+000	0							*
2.000E-002	0.0046		l arget f	unction				=
4.000E-002	0.0067							-
6.000E-002	0.0081	Γ	– Result – Target No					
8.000E-002	0.0092							
1.000E-001	0.0100		QMPS =	1.4735e+UU1				
1.200E-001	0.0107		Quit	E.	ecute			
1.400E-001	0.0113		suit	EX	coule			
1.600E-001	0.0117	0.0078	0.0064	0.0056				
1.800E-001	0.0121	0.0081	0.0066	0.0058				
2.000E-001	0.0125	0.0083	0.0068	0.0061				+

Fig.22 Result of QMPS in dynamic analysis

1.7.2 Steady-state analysis

The window for "Steady-state analysis" is displayed as shown in **Fig.23**. The "Results" on the window show the steady-state values. The "Sensitivity" on the window indicates the sensitivities with respect to each parameter and QMPSs. In this case, the target functions of QMPS are the steady-state values of each dependent variable y. By clicking the "Show Make_S_Param Result" button, the parameters for S-system are displayed (**Fig.24**). By clicking the "Show Eigen Values" button, the eigen values are displayed (**Fig.25**).

Sults Show g	raph Set	ing Reset) s	how Make_S_Paran	n Result S	how Eigen Valu	ues Export C	SV Export MATLA
PAR	AMETER SUR	VEY No.	-	Save for input				
Result	ts	S	ensitivity					
	y value			y[1].X1.cyt	y[2].X2.cyt	y[3].X3.cyt	y[4].X4.cyt	
1	0.0148		3 Q(2)	-5.9261	0	0	-2.2216e-13	
2	0.0099		4 Q(3)	-1.9994e-12	-3.4783	0	-2.2216e-13	
3	0.0083		5 Q(4)	-1.9994e-12	0	-2.6556	-2.2216e-13	Г
4	0.0075		6 Q(5)	0	0	0	-2.2430	
			7 Kmich(1)	-0.0119	-0.0070	-0.0053	-0.0045	
			8 Kmich(2)	1.0000	0	0	0	=
			9 Kmich(3)	0	1.0000	0	0	
			10 Kmich(4)	0	0	1.0000	0	
			11 Kmich(5)	0	0	0	1.0000	
			12 QMPS(1)	71.5859	25.2575	15.1281	11.0747	•

Fig.23 Results window for steady-state analysis

02					
	Variable	Meanings			
	Eigen	Eigen values			
	Sensitivity	Sensitivities with respect to each parameter and			
		QMPSs			
	tag	Name of the variables			
	у	Steady state values			

[Variables exported by "Export MATLAB"]

БТΑ	ART		Para	meters					
	Value	Tag			Value	num_survey	D/R/S	tag	
Γ	0.0148	X1.cyt	1	constantPlay	1	0 D		X0.cyt	
	0.0099	X2.cyt	1	kplus(1)	0.4990	0 D		alpha_1	
	0.0083	X3.cyt	2	kplus(2)	1.0139	0 D		alpha_2	
	0.0075	X4.cyt	3	kplus(3)	1.8761	0 D		alpha_3	
			4	kplus(4)	3.0287	0 D		alpha_4	-
			1	kminus(1)	1.0139	0 D		beta_1	
			2	kminus(2)	1.8761	0 D		beta_2	
			3	kminus(3)	3.0287	0 D		beta_3	
			4	kminus(4)	4.4233	0 D		beta_4	
			1	h(1)	0.1683	0 D		hd_1_1	
			2	h(2)	0.2871	0 D		hd_2_2	
				w/200	0.0700				-

Fig.24 Paramters of the s-system converted

1	CAE)LIVE_DispEig	jenValues		×
	Value:	s			
		Real	Imaginary		
	1	-29.7777	0		
	2	-22.6499	0		
	3	-14.4286	0		
	4	-5.6670	0		
				Close	

Fig.25 Eigen values

1.7.3 S-system

For the sensitivity analysis in S-system, dynamic analysis is first performed as described above (1.7.1). By clicking the "Show Make_S_Param Result" button in the results window for dynamic analysis (Fig.16), the parameters for S-system are displayed (Fig.24). By clicking 'Back' on the "Set Parameters and Initial Values" window (Fig.9), "Select Analysis Type & Set Control Data for Simulation" window is re-displayed (Fig.8). Users check the 'Use S-system' in the check box. By clicking the "Next" button, users can execute the simulation. The window for "S-system" is displayed as shown in Fig.26, where the sensitivities or logarithmic gains with respect to each parameter are displayed.

Results							
			(Show Eigen Values	Export	CSV Exp	ort MATLAB
PARAMETER :	SURVEY No.	T					
STD Values and	Fluxes						
STD valu	Je Flux	Tag					
1 0.0	148 0.4990 X	(1.cyt					
2 0.0	0.4990 X	2.cyt	E				
3 0.0	0.4990 X	(3.cyt					
			Ŧ				
Logarythmic Gair	is of Metabolites			Logarythmic Gain:	s of Fluxes		
		y[1]	y[2]			v[1]	v[2]
consta	ntPlayer[1]	0.0119	≜ .0 €	consta	ntPlayer[1]	0.0020	0.1
consta	ntPlayer[2]	5.940	7 3.	consta	ntPlayer[2]	1	
ronsta	ntPlaver[3]	5 040°	7 *	consta	ntPlaver[3]	∩	
Sepaitivities of M	latabalitaa with roor		tanto	Sopoitiuition of El	luves with respect to		
	v[1]		นโวโ				31
alnha[1]	5 9407	2 / 926	26579	alnha[1]	V[+]	*[2] *[4	1
alpha[2]	0.3407	3.4826	2.6578	alpha[2]	0	1	1
alnha[3]	0	0.4020	2.6578 -	alnha[3]	0	'n	4 -
	•		•		•		F
Sensitivities of M	letabolites with resp	pect to kinetic on	ders	Sensitivities of F	luxes with respect to	kinetic orders	
, i i i i i i i i i i i i i i i i i i i	y[1] y[2]	y[3]	y[4	- N	/[1] v[2]	v[3]	v[4
g[1]	0	0	0	g[1]	0	0 0	*
g[2]	0	0	0	g[2]	0	0 0	
n[3]	0 2/	1800 1.88	12 1	0[3]	0 0.700	0 0 7000	- T
							Close
						l	

Fig.26 Results window for S-system

2 GA

Genetic algorithms (GAs) are known as one of the algorithms that can seek out the global minimum, based on the heuristic assumptions that the best solutions will be found in the regions of the parameter space containing a relatively high proportion of good solutions and that these regions can be explored by the genetic operators of selection, crossover, and mutation. For GA of this application (**Fig.27**), the unimodal normal distribution crossover (UNDX) and minimal generation gap (MGG) are employed as crossover and selection, respectively. The mutation is not employed.



Fig.27 Flowchart for GA

2.1 Start

Execute "CADLIVE_SetGA" on the MATLAB command window to start the GA, the "CADLIVE_SetGA" window is displayed (**Fig.28**).

Users edit number of maximum generation, termination condition, number of population, number of children generated, search type, and alpha and beta in UNDX in "Edit GA condition". Analysis type indicates the analysis type for model of the current folder, "Steady-state analysis", "Dynamic Analysis" or "S-system". If users change the analysis type, they need to change the analysis type in section 1.5.

3 c	ADLIVE_SetGA		
	Edit GA Condition		Analysis type
	Number of maximum generation :	200	Dynamic Analysis
	Termination condition :	1e-20	Edit Search Parameters
	Number of population :	10	Edit parameters
	Number of children generated :	10	Edit Fitness Function
	Search type :	log 💌	Manual
	Alpha in UNDX :	0.5	
	Beta in UNDX :	0.35	Upload
	Quit		Execute

Fig.28 CADLIVE_SetGA window

*To execute the search by GAs, users need to input the initial values and parameters in section 1.5. The parameters can be edited by clicking the "Edit parameters" button.

Parameter	Values
Number of maximum generation	Integer ≥ 1
Termination condition	Real value ≥ 0
Number of population	Integer ≥ 1
Number of children generated	Integer ≥ 1
Search type	log, real
Alpha in UNDX	Real value > 0
Beta in UNDX	Real value > 0

Table.1 Values that users are allowed to set with respect to each parameter.

2.2 Search parameter setting

By clicking the "Edit parameters" button, CADLIVE_param.m is opened in the MATLAB editor (**Fig.29**). Users select the search parameters, which are optimized by GA. The search parameters are written as follows;

param.(name)(index).d_r_s = 'S';

'S' is a search parameter. Then, users set the search ranges (the lower and upper bounds) of the search parameters.

param.(name)(index).lowerBound = 1e-6;

param.(name)(index).upperBound = 1e+4;

When the search ranges are not set, they are automatically set param.(name)(index).vaule×0.1 and param.(name)(index).vaule×10 as the lower and upper bounds, respectively. When param.(name)(index).value = 0, the search range is 10^{-5} - 10^{5} .

📝 エディター	C:¥Users¥F00¥Desktop¥StraightChain¥CADLIVE_param.m
ファイル(<u>E</u>)	編集(E) テキスト(I) 移動(G) セル(C) ツール(Q) デバッグ(B) » > ママ×
: 🎦 🖨 🔳	│ ǎ 🐂 🛱 🤊 (♥ 🌭 🌝 - 🚧 🗢 🔶 🈥 ▶ - 🗟 🔏 🖷 🖬 🗊 🗐 - 🏾 🗨
*=	$-1.0 + \div 1.1 \times \% \% \% 0$
25 -	param.constantPlayer(3).tag = 'E2.cyt';
26 -	param.constantPlayer(4).value = 1.0000e+000;
27 -	param.constantPlayer(4).num_survey = 0;
28 -	param.constantPlayer(4).d_r_s = 'D';
29 -	param.constantPlayer(4).tag = 'E3.cyt';
30 -	param.constantPlayer(5).value = 1.0000e+000;
31 —	param.constantPlayer(5).num_survey = 0;
32 —	param.constantPlayer(5).d_r_s = 'D';
33 -	param.constantPlayer(5).tag = 'E4.cyt';
34 -	param.Q(1).value = 5.0000e-001;
35 —	param.Q(1).lowerBound = 1.0000e-006;
36 -	param.Q(1).upperBound = 1.0000e+004;
37 -	param.Q(1).num_survey = 0;
38 -	param.Q(1).d_r_s = 'S';
39 -	<pre>param.Q(1).tag = 'reaction_rate_constant_E1.cyt_XU.cyt_MM';</pre>
40 -	param.Q(2).value = 6.0000e-001;
41 -	param.Q(1).lowerBound = 1.0000e-005;
42 -	param.Q(1).upperBound = 1.0000e+002;
43 -	param.U(2).num_survey = U;
44 -	param.W(2).d_r_s = 181;
45 -	<pre>param.u(z).tag = 'reaction_rate_constant_t2.cyt_X1.cyt_MM'; newer_0(2).usius0_0000.0000;</pre>
46 -	param.Q(5).value = 0.00000+000;
47 -	param.Q(3).num_survey = 0;
48 -	param. $Q(2)$ to $r_1 = -3$;
40 -	param.Q(4) value = 8 0000e-001:
50 -	param $Q(A)$ pum curvey = 0.
	varam·xx(4/)·inum_out/09/ = 0;
	CADLIVE_param 行 38 列 26 上書き

Fig.29 CADLIVE_param.m for set of search conditions

2.3 Fitness function setting

By clicking the "Manual" button, the file CADLIVE_getFitness.m is opened if the file exists in the current folder. If the file does not exist, the file is newly made and opened (**Fig.30**). CADLIVE_getFitness.m is a file for setting a fitness function. Users write a fitness function below "%input a fitness function below" according to the MATLAB language. In "Steady-state analysis", "y(col)" is steady-state values. In "Dynamic Analysis", "y(row,col)" indicates the time-dependent variable, where "row" is the monitoring index and "col" is index of the dependent variable. An index value in y corresponds to the index of "Y_START" in CADLIVE_initial.m. "t" is time. The fitness function is defined as the minimization problem of *fitness* ≥ 0 . *fitness* = 0 indicates that a set of the parameters is completely optimized with respect to the fitness function.

デエディター - C:¥Users¥F00¥Desktop¥sample¥DirectedModel¥CADLIVE_getFitness.m	
ファイル(E) 編集(E) テキスト(I) 移動(G) セル(C) ツール(Q) デバッグ(B) デスクトップ(D) ウィンドウ(W) »	XSK
: 🎦 😂 📓 👗 🍋 🛱 🤨 🤭 🥸 📨 - 🏘 🆛 🔶 🈥 - 🖥 📽 🖷 🏙 🕼 📾 🕼 🖉 - スーマー 🍂	
$\stackrel{*}{=} \ \ \ \ \ \ \ \ \ \ \ \ \ $	
1 ⊡ <mark>function fitness = CADLIVE_getFitness(y,t</mark>)	
2 P%t is time.	
3 % y(col) is steady-state values and t is not used, if analysis type is steady-state analysis.	
4 % y(row,col) is dynamics of each dependent variable, if analysis type is dynamic analysis.	
5 -% row is index of value according to t and col is index of dependent variable.	
6	
7 - Lfitness = 0;	
8	
9 %input a fitness function below.	
10	
CADLIVE_getFitness 行 1 列 1 _	上書き :

Fig.30 CADLIVE_getFitness.m

For "Dynamic Analysis", users can set the fitness function as the sum of squared errors (SSE). Users make a file with time-course data (**Fig.31**) such as experimental data. The fitness function is set as the SSE (**Fig.32**). When clicking the "upload" button the file is uploaded. The SSE should be minimized for a parameter set P.

$$SSE(P) = \sum_{i=1}^{N} \sum_{j=1}^{k} \left(\frac{x_{ij}(P) - y_{ij}}{y_{ij}} \right)^{2},$$

where $x_{ij}(P)$ are the simulated data corresponding to the experimental or reference data y_{ij} . *N* is the number of molecules for optimization. *k* is the number of the experimental data.

Here, the fitness function is defined as the SSE using the data (Fig.32).

🖺 デスクトップ¥sample¥StraightChain¥SSE.txt - sakura 1.6.5.0 🛛 💼 💷	×
ファイル(E) 編集(E) 変換(C) 検索(S) ツール(T) 設定(Q) ウィンドウ(W	0
ヘルプ(日)	
📑 😂 🔻 🗔 🔚 🗠 ભ 🖛 😁 😓 🏶 💩 💏 🖓 🆓 🖓	4 🗄
[] 1 time X1.cyt X2.cyt X3.cyt X4.cyt↔	<u>و</u>
2 0.00E+00 [^] 0.00E+00 [^] 0.00E+00 [^] 0.00E+00 [^] 0.00E+00 [←]	
3 8.00E-02 9.20E-03 5.90E-03 4.62E-03 3.84E-03 4 1 80E-01^ 1 17E-02^ 7 78E-03^ 8 98E-03^ 5 58E-03	
5 2.40E-01^ 1.30E-02^ 8.68E-03^ 7.19E-03^ 6.40E-03	
6 3.20E-01 1.37E-02 9.18E-03 7.64E-03 6.84E-03↔	
7 4.00E-01 1.41E-02 9.47E-03 7.89E-03 7.09E-03	
8 4.80E-01 1.44E-02 9.65E-03 8.04E-03 7.23E-03↔	
10 6.40E-01^ 1.47E-02^ 9.82E-03^ 8.19E-03^ 7.38E-03	
11 7.20E-01↑ 1.47E-02↑ 9.86E-03↑ 8.23E-03↑ 7.41E-03↔	
12 8.00E-01 1.48E-02 9.89E-03 8.25E-03 7.43E-03	
13 8.80E-01 1.48E-02 9.90E-03 8.27E-03 7.44E-03	
15 1.00E+00^ 1.48E-02^ 9.92E-03^ 8.28E-03^ 7.46E-03 -	
EOF)	
	*
	•
16行 1桁 CRLF SJIS REC 指	际 。

Fig.31 File for SSE

 エディター - C:¥Users¥F00¥Desktop¥sample¥StraightChain¥CADLIV	E_getFitness.m		- • ×	
ファイル(E) 編集(E) テキスト(I) 移動(G) セル(C) ツール(Q) :	デバッグ(<u>B)</u> デスクトップ(<u>D</u>)	ウィンドウ(<u>W</u>) ヘルプ(<u>н)</u> тх	
: 🞦 😂 📓 👗 🛍 🤊 🕫 🍓 🖅 - 🚧 🗭 🔶 🗩 -	8 * • • * * * * *	スタック (<u>K</u>): ベース 🖃	fx. 🗆 🗸	
# [# - 1.0 + ÷ 1.1 × % % 0				
1 KSquared error				
2 == function fitness = CADLIVE_getFitness(y,t)				
4 – fitness = O;				
5				
6 - Texp(1,1)=0.000000e+000;Yexp(1,1)=0.000000e+000;Yex	>(1,2)=0.000000e+000;Yexp	(1,3)=0.000000e+000;Ye	$\exp(1,4)=0.0$	
7 - Texp(2,1)=8.000000e-002; texp(2,1)=9.200000e-003; tex 8 - Texp(3,1)=1.600000e-001; Yexp(3,1)=1.170000e-002; Yex	o(2,2)=5.900000e-003;Yexp o(3.2)=7.78NNNNe-NN3:Yexp	(2,3)=4.620000e-003;Ye (3.3)=6.360000e-003;Ye	xp(2,4)=3.8 xp(3.4)=5.5	
9 - Texp(4,1)=2.400000e-001;Yexp(4,1)=1.300000e-002;Yex	(4,2)=8.680000e-003;Yexp	(4,3)=7.190000e-003;Ye	exp(4,4)=6.4	
10 - Texp(5,1)=3.200000e-001;Yexp(5,1)=1.370000e-002;Yex	o(5,2)=9.180000e-003;Yexp	(5,3)=7.640000e-003;Ye	xp(5,4)=6.8	
11 - Texp(6,1)=4.000000e-001;Yexp(6,1)=1.410000e-002;Yex	>(6,2)=9.470000e−003;Yexp (7,0)=0.050000e−003;Yexp	(6,3)=7.890000e-003;Ye	xp(6,4)=7.0	
12 - Texp(7,1)=4.800000e-001;Yexp(7,1)=1.440000e-002;Yex 13 - Texp(8,1)=5.600000e-001;Yexp(8,1)=1.460000e-002;Yex	0(/,2)=9.6500000e-003;Yexp (8.2)=9.750000e-003;Yexp	((/,3)=8.040000e-003;Ye ((8.3)=8.140000e-003;Ye	xp(7,4)=7.2	
14 - Texp(9,1)=6.400000e-001;Yexp(9,1)=1.400000e-002;Yex	o(9,2)=9.820000e-003;Yexp	(9,3)=8.190000e-003;Ye	exp(0,4)=7.3	
15 - Texp(10,1)=7.200000e-001;Yexp(10,1)=1.470000e-002;Y	exp(10,2)=9.860000e-003;	exp(10,3)=8.230000e-00	J3;Yexp(10,₄	
16 - Texp(11,1)=8.000000e-001;Yexp(11,1)=1.480000e-002;Y	exp(11,2)=9.890000e-003;Y	exp(11,3)=8.250000e-00	J3;Yexp(11,4	
17 - Texp(12,1)=8.800000e-001;Yexp(12,1)=1.480000e-002;Y	exp(12,2)=9.900000e-003;	exp(12,3)=8.270000e-00	13;Yexp(12,4	
18 - lexp(13,1)=9.600000e-001;Yexp(13,1)=1.480000e-002;Y	exp(13,2)=9.910000e-003;)	exp(13,3)=8.2/UUUUe-UU	/3;Yexp(13,4	
20	sxp(14,2)=3.320000e=003,1	exp(14,3)=0.200000e=00	5, Texp(14, 1	
21 %X1.cyt				
22 - yi(:,1)=interp1(t,y(:,1),Texp,'spline');				
23 %X2.cyt				
24 - yi(:,2)=interp1(t,y(:,2),Texp,'spline');				
25 $\%$ X3.Cyt 28 - vi(: 3)-intern1(t v(: 3) Texp 'enline'):				
27 %X4.cvt				
<pre>28 - yi(:,4)=interp1(t,y(:,4),Texp,'spline');</pre>				
29				
30 - numberOfExperiment = 14;				
31 - numberOfFitParam = 4;				
32 - FTOT I=I:humberOfFitParam				
34 - if Yexp(i,i) == 0	- if Yeyn(i i) 0			
35 - fitness = fitness + (yi(j,i)-Yexp(j,i))	`2;			
36 — else				
37 - fitness = fitness + ((yi(j,i)-Yexp(j,i))/Yexp(j,i))^2;			
38 – end				
33 - end				
40end 41				
K			•	
CADLIVE_param.m × CADLIVE_getFitness.m ×				
	CADLIVE_getFitness	行 1 列	1 上書き .::	

Fig.32 CADLIVE_getFitness.m input as SSE

2.4 Execution

By clicking the "Execute" button, the GA search is executed. During the optimization, the progress bar is displayed (**Fig.33**). The number indicates (number of generation calculated) / (maximum number of generation).

🛃 GA Progress		- • •
	77/200	
		Cancel



When the optimization finishes, the graph for changes in the fitness with respect to generation (**Fig.34**), the value of best fitness, and the simulation result with the optimized parameters are displayed (**Fig.35**), and OptimizedParameter.mat and CADLIVE_OptParam.m are output. OptimizedParameter.mat is the data of the parameters optimized by GA, and CADLIVE_OptParam.m is the file for calling OptimizedParameter.mat. When the content in CADLIVE_OptParam.m is copied into CADLIVE_param.m, users can simulate the model with the optimized parameters.



Fig.34 Fitness graph

sults Show graph Settin	ng Reset] [Show Make_S_	Param Result	Show Eigen Values	Export CSV	Export MATLAB
PARAMETER SURVE	EY No.	-	Save for in	put			QMPS
Results	y[1].X1.cyt	y[2].X2.cyt	y[3].X3.cyt	y[4].X4.cyt			
0.000E+000	0	0	0	0			
2.000E-002	9.5308e-04	1.5904e-04	6.1070e-06	8.0101e-08			=
4.000E-002	0.0017	4.7764e-04	3.6052e-05	9.1060e-07			
6.000E-002	0.0024	8.5836e-04	9.7921e-05	3.7530e-06			
8.000E-002	0.0030	0.0013	1.9236e-04	1.0053e-05			
1.000E-001	0.0036	0.0017	3.1757e-04	2.1214e-05			
1.200E-001	0.0041	0.0021	4.7082e-04	3.8535e-05			
1.400E-001	0.0046	0.0025	6.4899e-04	6.3169e-05			
1.600E-001	0.0051	0.0029	8.4901e-04	9.6143e-05			
1.800E-001	0.0056	0.0033	0.0011	1.3836e-04			
2 000E-001	0.0060	0.0037	0.0013	1.9057e-04			-

Fig.35 Result for the simulation with optimized parameters

3 TPS

Two-Phase Search (TPS) smoothly combines a random search with an evolutionary algorithm to achieve both nonbiased and high-speed searches for a large parameter space (**Fig.36**). Use of QMPS with the TPS reveals the mechanism of how a particular architecture is related to robustness in complex regulations.



Fig.36 Schematic diagrams of the TPS method (Fig.1 in [6])

A: A flow chart for TPS that consists of a random search (the first phase) and a search by GAs (the second phase). B: How to create the initial populations for the second phase search by GAs. AE, AEC and RIG indicate the allowable error, the allowable error for the coarse solution, and the region of the initial population for the search by GAs, respectively.

3.1 Start

Execute "CADLIVE_SetTPS" on the MATLAB command window to start the TPS, the "CADLIVE_SetTPS" window is displayed (**Fig.37**). In "Dynamic Analysis", users can select whether QMPS is executed because QMPS may require a long calculation time. In "Steady-state Analysis", QMPS is always calculated.

ADLIVE_SetTPS		
Edit TPS Condition		Analysis Type
Number of solution :	100	Dynamic Analysis
AE :	1e-5	Edit Search Parameters
GA condition		Edit parameters
AEC :	1e-3	- Edit Fitness Function
RRIG : Number of maximum generation :	0.2	Manual
Number of population :	5	Upload
Number of children generated :	5	
Search type :	log 👻	Edit QMPS
Alpha in UNDX :	0.5	Calculate ? 🛛 🖉
Beta in UNDX :	0.35	Delta : 0.001
		Target function
Quit		Execute

Fig.37 CADLIVE_SetTPS window

AE: Allowable error, AEC: Allowable error for the coarse solution, RRIG: Relative value of the region of the initial population for the search by GAs (RIG) to the search region of each parameter.

* To execute the TPS, users need to input the initial values and parameters in section 1.6. The parameters can be edited at the "Edit parameters" button.

Parameter	Values
Number of solution	Integer ≥ 1
AE	Real value ≥ 0
AEC	Real value \geq AE
RRIG	Real value > 0
Number of maximum generation	Integer ≥ 1
Termination condition	Real value ≥ 0
Number of population	Integer ≥ 1
Number of children generated	Integer ≥ 1
Search type	log, real
Alpha in UNDX	Real value > 0
Beta in UNDX	Real value > 0

Table2 Values that users are allowed to set with respect to each parameter.

Here, we set the objective function as follows:

objective function =
$$\left(\frac{X4(t) - 5 \times 10^{-3}}{5 \times 10^{-3}}\right)^2$$
,

where X4(t) is the value of X4 on the end time. The target functions are the values of X1, X2, X3 and X4 on the end time, shown as **Fig.21**.

3.2 QMPS on TPS

QMPS on TPS may require a long calculation time because it calculates QMPS with respect to many plausible solutions obtained by TPS. When QMPS is calculated, users check the checkbox beside "Calculate ?". The "Delta" and "Target function" are described in section 1.7.1. In "Steady-state Analysis", QMPS on TPS is always calculated.

3.3 Execution

By clicking the "Execute" button, TPS is executed. During the process, the progress bar is displayed (**Fig.38**). The number indicates (number of solution calculated) / (number of solution). When the calculation for TPS finishes, two files (TPSresultinfo.dat (**Fig.39**), TPSresult.mat) are output. When QMPS is calculated, the progress bar for QMPS is displayed (**Fig.40**). The number indicates (number of solution calculated) / (number of solution) and (number of single parameter sensitivity calculated) / (number of parameter) for QMPS. When the calculation for QMPS finishes, the file TPStoQMPS.mat is output and the cumulative frequency distribution for QMPS is displayed (**Fig.41**).



Fig.38 Progress of TPS

* If suitable parameters with respect to the fitness function are not searched or the TPS fails, users can check the fitness on the MATLAB command window, and can change the values of AE and AEC.

🗓 デスクトップ¥QMPStest¥StraightChain_dynamic¥TPSresultinfo.dat - sakura 1.6.5.0 💿 💿 💽
ファイル(E) 編集(E) 変換(C) 検索(S) ツール(I) 設定(Q) ウィンドウ(W) ヘルプ(H)
□ ジ ジ マ 日 日 ・
19
2 Number of solutions setting: 100+
3 AE: 1.00000e=005↓ 4 AEC: 1.000000e=03↓
5 RRIG: 0.200000.
b Number of maximum generation: 100↓ 7 Number of population: 5↓
8 Number of children generated: 5+
a bearbn type. Tog+ TO Alpha. 0.5000000 +
11 Beta: 0.350000↓
13 \$\$Perfomances →
14 (Calculation time: 0 hour 20 min 24 sec↓ 15 Number of trials for TPS: 14490↓
16 EVA: 29640+
17 Number of solutions obtained: 10U+ 18 Number of solutions obtained in RS: 10+
19 Number of solutions obtained in GA: 90 +
20 Wumber of solutions obtained in first generation of GA: U+ 21 +
22 \$\$search parameters+
24 JQ(1) 8.865361e-001 9.565007e-001 4.537395e+000 6.156828e-002 4.7(1516e-01) 5.00000e+000 5.000000e-002 +
25 0(2)^ 1.085986e+000^ 1.137881e+000^ 5.885496e+000^ 7.151516e-002^ 6.420540e-001^ 6.000000e+000^ 6.000000e-002^ +
27 0(4) 1.565038+000 1.14338+000 7.553053+000 1.1188518-001 8.0000008+000 6.0000008-002 +
28 [9(5)] 1.120620e+000^ 1.185880e+000^ 8.183079e+000^ 1.317065e-001^ 7.473011e-001^ 9.000000e+000^ 9.000000e-002^ +
30 Kmich(2) 5.991500e-003 6.622959e-003 2.844204e-002 3.181860e-004 3.015191e-003 3.000000e-002 3.000000e-004 +
31 Kmich(3) 6.329837e-003 6.2010/6e-003 3.57/7864e-002 4.210945e-004 4.4594410e-003 4.000000e-002 4.000000e-004 + 32 Kmich(4) 9.768647e-003 1.020768e-002 4.962771e-002 5.251092e-004 5.5556582e-003 5.000000e-002 5.0000000e-004 +
33 kmich(5)^ 1.867815e-002^ 1.261703e-002^ 4.864755e-002^ 1.341962e-003^ 1.359117e-002^ 6.000000e-002^ 6.000000e-004^ +
<u>↓</u> € ►
1 行 13 桁 LF 0a UTF-7 REC 挿入

Fig.39 Result for TPS

The file TPSresultinfo.dat is written condition and performances for TPS, and statistics with regard to search parameters.

TPSresult.mat

[Structure variable "searchParam"]

Field	Meanings
paramName	Parameter name
paramIndex	Index of paramName
upperBound	Upper bound of parameter search
lowerBound	Lower bound of parameter search
newValue	Ignore

[Structure variable "solution"]

Field	Meanings
fitness	Value of fitness
paramValue	Vector of the parameter with respect to index of
	searchParam
type	Method when the result is obtained
	'RS': Random search
	'GA': Genetic algorithm
	'GAinit': First generation of GA

TPStoQMPS Progress	- • -
70/100(1/10)	
	Cancel

Fug.40 Progress of QMPS on TPS



Fig.41 Cumulative frequency distribution of QMPS

TPStoQMPS.mat

Variable	Meanings
QMPS	Value of QMPS
tag	Target number when "Dynamic Analysis"
	Name of variable when "Steady-state Analysis"

4 Execution on command line

Users can use the simulator, GA, TPS and QMPS without GUI.

4.1 CADLIVE_simulator

This is the command for executing simulation.

Preparations:

CADLIVE_initial.m, CADLIVE_param.m, CADLIVE_fvec.m and CADLIVE_fjac.m in the current folder.

Command:

SimResult = CADLIVE_simulator();

Return:

SimResult : Simulation result (structure variable)

[Structure variable "SimResult"]

Field	Meanings				
Result	Success or fail (true or false)				
Result_data	Structure variable of simulation result according to				
	analysis type				
	The details are described in follow Table.				
Y	Final values after simulation				
Y_pre	Initial values				
tag	Name of each dependent variable				
Eigen	Eigen values				
S_Param	Parameters for S-system				

[Structure variable "Result_data" in dynamic analysis]

Field	Meanings
time	Time
Y	Dynamics of each dependent variable

[Structure variable "Result_data" in steady-state analysis]

Field	Meanings
sensitivity	Sensitivity to each parameter and QMPS for the change of
	"NR_SENS_CW" of "CADLIVE_CTL" in section 1.6.1

[Structure variable "Result_data" in S-system analysis]

Field	Meanings
STDValueFlux	STD Values and Fluxes
LogGainMetabo	Logarithmic Gains of Metabolites
SensMetaboRateConst	Sensitivities of Metabolites with respect to
	rate constants
SensMetaboKinOrder	Sensitivities of Metabolites with respect to
	kinetic orders
LogGainsFlux	Logarithmic Gains of Fluxes
SensFluxRateConst	Sensitivities of Fluxes with respect to rate
	constants
SensFluxKinOrder	Sensitivities of Fluxes with respect to kinetic
	orders

When users simulate with change of initial values and/or parameters iteratively, users can use the following command.

Preparations:

CADLIVE_initial.m, CADLIVE_param.m, CADLIVE_fvec.m and CADLIVE_fjac.m in the current folder.

Command:

SimResult = CADLIVE_simulatorForIteration(CADLIVE_CTL, Y_START, param, event);

Return:

SimResult : Simulation result (structure variable)

Example:

```
[CADLIVE_CTL Y_START]=CADLIVE_initial();
[param, event]=CADLIVE_param();
for i=1:5
    Y_START(1).value = 0.1*i;
SimResult(i) = CADLIVE_simulatorForIteration(CADLIVE_CTL, Y_START, param,
event);
end
```

4.2 CADLIVE_myGAcommand

This is the command for executing GA. If the search by GA is successful, OptimizedParameter.mat and CADLIVE_OptParam.m are output.

Preparations:

CADLIVE_initial.m, CADLIVE_param.m, CADLIVE_fvec.m, CADLIVE_fjac.m and CADLIVE_getFitness.m in the current folder.

Command:

[SimResult fitness] = CADLIVE_myGAcommand(...

n_generation, n_population, n_children, allowable_error, searchType, alpha, beta);

Arguments:

n_generation	:	Number of maximum generation	; Integer ≥ 1
n_population	:	Number of population	; Integer ≥ 1
n_children	:	Number of children generated	; Integer ≥ 1
allowable_error	:	Termination condition	; Real value ≥ 0
searchType	:	Search type	; log, real
alpha	:	Alpha in UNDX	; Real value > 0
beta	:	Beta in UNDX	; Real value > 0

Return:

SimResult	:	Simulation result with optimized parameters
fitness	:	Fitness values

Example:

[SimResult fitness]= CADLIVE_myGAcommand(100, 5, 5, 1e-20, 'log', 0.5, 0.35);

4.3 CADLIVE_myTPScommand

This is the command for executing TPS. If the search by TPS is successful, two files (TPSresultinfo.dat (**Fig.39**), TPSresult.mat) are output.

Preparations:

CADLIVE_initial.m, CADLIVE_param.m, CADLIVE_fvec.m, CADLIVE_fjac.m and CADLIVE_getFitness.m in the current folder.

Command:

[solution searchParam info flag] = CADLIVE_myTPScommand(term_n_solution, ... AE, AEC, RRIG,...

n_generation, n_population, n_children, ...

searchType, alpha, beta);

Arguments:

term_n_solution	:	Number of solution	; Integer \geq	1	
AE	:	Allowable error for random search	; Real value	\geq	0
AEC	:	Allowable error for the coarse solution	; Real value	\geq	AE
RRIG	:	Relative value of the region of the initial	; Real value	\geq	0
		population for the search by GAs (RIG) to the			
		search region of each parameter			
n_generation	:	Number of maximum generation	; Integer \geq	1	
n_population	:	Number of population	; Integer \geq	1	
n_children	:	Number of children generated	; Integer \geq	1	
searchType	:	Search type	; log, real		
alpha	:	Alpha in UNDX	; Real value 2	> 0	
beta	:	Beta in UNDX	; Real value 2	> 0	

Returns:

solution	:	Result (structure variable)
searchParam	:	Condition of search parameter (structure variable)
info	:	Information of the result (structure variable)
flag	:	Success or failed (flag=0 or flag <0)

Example:

[solution searchParam info flag] = CADLIVE_myTPScommand(100, 1e-3, 1.01, 0.2,... 100, 5, 5, 'log', 0.5, 0.35);

Field	Meanings	
fitness	Value of fitness	
paramValue	Values of search parameters	
type	Search method obtained the result;	
	'RS', 'GA', 'GAinit'	

[Structure variable "solution"]

[Structure variable "searchParam"]

Field	Meanings			
paramName	Parameter name			
paramIndex	ndex of paramName			
upperBound	Upper bound of parameter search			
lowerBound	Lower bound of parameter search			
newValue	Ignore			

[[]Structure variable "info"]

Field	Meanings
n_trail	Number of trials for TPS
n_rs	Number of parameter set obtained
	in random search
n_solution	Total number of parameter set
	obtained by TPS
n_coarse_solution	Number of trials by GA in TPS
n_solution_obtained_in_GA	Number of parameter set obtained
	in GA
n_solution_obtained_in_first_generation	Number of parameter set obtained
	in first generation of GA
n_evalution	Number of evaluations for fitness
	function (EVA)

*The indices of "paramValue" of solution correspond to those of "searchParam".

4.4 CADLIVE_DispFigure

This is the command for displaying a figure of the result in dynamic analysis. CADLIVE_DispFigure.m is output in the current folder when "Dynamic analysis" is successful (section 1.7.1). When users execute the simulation on the MATLAB command window, the resultant simulation can be displayed by using this command (**Fig.42**).

Preparations:

SimResult = CADLIVE_simulator(); or load a file output by "Export MATLAB"

Command:

CADLIVE_DispFigure(timeStep, y , y_tag);

Arguments:

timeStep	:	Time
У	:	Dynamics of each dependent variable
y_tag	:	Names of each dependent variable

Example:

CADLIVE_DispFigure(SimResult.Result_data.time, SimResult.Result_data.y, ... SimResult.tag);



Fig.42 CADLIVE_DispFigure

4.5CADLIVE_DispCumulFreqQMPS, CADLIVE_DispHistFreqQMPS, CADLIVE_DispHistCumulFreqQMPS

These are the commands for displaying the QMPSs on TPS. In CADLIVE_SetTPS (GUI) in section 3, it displays the cumulative frequency distribution of QMPS on a certain state for the mathematical model by the command CADLIVE_DispCumulFreqQMPS. Using these commands on command line, it can simultaneously display the distributions for QMPSs among some models (e.g. wild-type model and knockout models or non-branching model and branching models), where robustness of them can be characterized by their QMPSs.

CADLIVE_DispCumulFreqQMPS displays a cumulative frequency distribution with all data points for the values of QMPS (**Fig.43**). CADLIVE_DispHistFreqQMPS displays a histogram for a frequency distribution (**Fig.44**). CADLIVE_DispHistCumulFreqQMPS displays a cumulative frequency distribution whose data points are smoothed by the values of the bin for histogram instead of the actual values of QMPS (**Fig.45**).

Preparations:

load TPStoQMPS.mat

Commands:

CADLIVE_DispCumulFreqQMPS(QMPS, tag); CADLIVE_DispHistFreqQMPS(QMPS, tag, nbin, lb, ub); CADLIVE_DispHistCumulFreqQMPS(QMPS, tag, nbin, lb, ub);

Arguments:

QMPS	:	QMPS; each column is corresponding to QMPS for each target
tag	:	Name of target function
nbin	:	Number of classes for the histograms
lb	:	Lower bound of the histograms
ub	:	Upper bound of the histograms

Example:

cd (the folder for the StraightChain model);

load TPStoQMPS.mat;

Q(:,1)=QMPS(:,1); t(1)=cellstr('Straight');

cd (the folder for the model having two branches);

load TPStoQMPS.mat;

Q(:,2)=QMPS(:,1); t(2)=cellstr('Branch2');

cd (the folder for the model having three branches)

load TPStoQMPS.mat;

Q(:,3)=QMPS(:,1); t(3)=cellstr('Branch3');

cd (the folder for the model having five branches)

load TPStoQMPS.mat;

Q(:,4)=QMPS(:,1); t(4)=cellstr('Branch5');

CADLIVE_DispCumulFreqQMPS(Q, t);

CADLIVE_DispHistFreqQMPS(Q, t, 20, 6, 11);

CADLIVE_DispHistCumulFreqQMPS(Q, t, 20, 6, 11);



Fig.43 CADLIVE_DispCumulFreqQMPS



Fig.45 CADLIVE_DispHistCumulFreqQMPS



Fig.44 CADLIVE_DispHistFreqQMPS

4.6 CADLIVE_optimtool

This is the command for executing GAs by using the optimization toolbox (<u>www.mathworks.co.jp/jp/help/gads/genetic-algorithm.html</u>). To use this command, users buy the optimization toolbox provided by Mathworks Inc. If the search by GAs is successful, OptimizedParameter.mat and CADLIVE_OptParam.m are output.

Preparations:

CADLIVE_initial.m, CADLIVE_param.m, CADLIVE_fvec.m, CADLIVE_fjac.m and CADLIVE_getFitness.m in the current folder.

Command:

[SimResult fitness] = CADLIVE_optimtool(options)

Arguments:

options	:	Option	according	to	optimization	toolbox			
		(http://www.mathworks.co.jp/jp/help/gads/genetic-algorithm-options.html)							
		No argument is set options = gaoptimset(@ga).							

Return:

SimResult	:	Simulation result with optimized parameters

fitness : Best fitness value

Example:

[SimResult fitness] = CADLIVE_optimtool();

References

[1] Hiroyuki Kurata, Nana Matoba, Natsumi Shimizu, CADLIVE for constructing a large-scale biochemical network based on a simulation-directed notation and its application to yeast cell cycle, *Nucleic Acids Res.*31: 4071-4084, 2003

[2] Hiroyuki Kurata, Kouichi Masaki, Yoshiyuki Sumida, Rei Iwasaki, CADLIVE Dynamic Simulator: Direct Link of Biochemical Networks to Dynamic Models, *Genome Res.*, 15: 590-600, 2005

[3] Hiroyuki Kurata, Kentaro Inoue, Kazuhiro Maeda, Koichi Masaki, Yuki Shimokawa, Quanyu Zhao, Extended CADLIVE: a novel graphical notation for designing a biochemical network map that enables computational pathway analysis, *Nucleic Acids Res.* 35(20):e134, 2007

[4] Kentaro Inoue, Sayaka Tomeda, Shinpei Tonami, Yuki Shimokawa, Masayo Ono, Hiroyuki Kurata, CADLIVE Converter for constructing a biochemical network map, *Biochemical Engineering Journal*, 54(3): 200-206, 2011

[5] Kentaro Inoue, Kazuhiro Maeda, Yuki Kato, Shinpei Tonami, Shogo Takagi, Hiroyuki Kurata, CADLIVE Optimizer: Web-based Parameter Estimation for Dynamic Models, *Source Code for Biology and Medicine*, 7:9, 2012

[6] Kazuhiro Maeda, Hiroyuki Kurata, Two-phase search (TPS) method: Nonbiased and high-speed parameter search for dynamic models of biochemical networks, *IPSJ Transaction on Bioinformatics* 2:2-14, 2009

[7] Kazuhiro Maeda, Hiroyuki Kurata, Quasi-multiparameter sensitivity measure for robustness analysis of complex biochemical networks *J Theor Biol* 272:174–186, 2011