

Paint4Net: visualisation toolbox for COBRA

Author(s):

Andrejs Klotins, Biosystems Group, Department of Computer Systems, Latvia University of Agriculture, Lielā iela 2, LV-3001 Jelgava, Latvia.

Egils Stalidzans, Institute of Microbiology and Biotechnology, University of Latvia, Jelgavas iela 1, LV-1004, Latvia.

Reviewer(s):

Agne Pētņeva, Institute of Microbiology and Biotechnology, University of Latvia, Jelgavas iela 1, LV-1004, Latvia.

Almut Heinken, Luxembourg Centre for Systems Biomedicine, University of Luxembourg, 8 avenue du Sierg, Belvaux, L-1207, Luxembourg.

IMPORTANT NOTE

Paint4Net uses Bioinformatics Toolbox to generate visualization layout, however it is not supported in *mls* causing an error during function execution. Thus the functions involving visualization were run in regular MATLAB command window and each visualization layout was saved as a static figure and placed in the *mls* tutorial, while the corresponding functions were run in *mls* with visualization feature turned off (input argument *drawMap* was set to 'false') to get outputs (without visualization) in *mls* without crashing. Be aware of this issue when you are running the functions in *mls*. All *drawMap* input arguments are set to 'true' in *mls*. Change it to 'false' to avoid an error.

INTRODUCTION

A visual analysis of reconstructions and large stoichiometric models with elastic change of the visualization scope and representation methods becomes increasingly important due to the rapidly growing size and number of available reconstructions.

The Paint4Net is a COBRA Toolbox extension for automatic generation of a hypergraph layout of defined scope with the steady state rates of reaction fluxes of stoichiometric models. Directionalities and fluxes of reactions are constantly represented in the visualization while detailed information about reaction (ID, name and synonyms, and formula) and metabolite (ID, name and synonyms, and charged formula) appears placing the cursor on the item of interest. Additionally Paint4Net functionality can be used to: (1) get lists of involved metabolites and dead-end metabolites of the visualized part of the network, (2) exclude (filter) particular metabolites from representation, (3) find isolated parts of a network and (4) find running cycles when all of the substrates are cut down. Layout pictures can be saved in various formats and easily distributed.

Two functions with their arguments are used in the Paint4Net to define the scope of visualization: (1) `[involvedMet, deadEnds] = drawBy rxn(model, rxns, drawMap, direction, initiator, excludeMet, flux)` – to define scope by a list of reactions and (2) `[directions, involvedMet, deadEnds] = drawBy met(model, metId, drawMap, radius, direction, excludeMet, flux)` – to define the metabolite of interest to see linked reactions within radius of, for instance, 3 reactions.

The function `drawBy rxn` has 7 input arguments: (1) *model* – stands for stoichiometric reconstruction or model with constraints, (2) *rxns* – stands for a list of the reactions of interest for analysis, (3) *drawMap* (optional) – stands for request to generate visualization ('true' or 'false', default is 'false'), (4) *direction* (optional) – stands for algorithm visualization mode ('true', 'false', 'prod' or 'both') in order to visualize structure (true) of reconstructions without FBA data or visualize substrates (sub), products (prod) or substrates and products (both) for models with flux constraints and FBA data (default is 'true'), (5) *initiator* (optional) – stands for metabolite of interest to be used by function `drawBy met` (default is empty), (6) *excludeMet* (optional) – stands for a list of the excludable metabolites as a filter and (7) *flux* (optional) – stands for vector of FBA data of reactions flux distribution (default is vector of 1's characters if flux is not calculated). The last 5 arguments are optional and can be unset. The function `drawBy met` has 3 outputs: (1) *involvedMet* – stands for a list of involved metabolites depending on the input arguments and (2) *deadEnds* – stands for a list of dead-end metabolites depending on the input arguments. The function `drawBy met` has 7 input arguments: (1) *model* – stands for stoichiometric reconstruction or model with constraints, (2) *metId* – stands for an input for metabolite of interest for analysis, (3) *drawMap* (optional) – stands for request to generate visualization ('true' or 'false', default is 'false'), (4) *radius* (optional) – stands for distance indicator between metabolite of interest and involved reactions (default is 1), (5) *direction* (optional) – stands for algorithm visualization mode ('true', 'false', 'prod' or 'both') in order to visualize structure (true) of reconstructions without FBA data or visualize substrates (sub), products (prod) or substrates and products (both) for models with flux constraints and FBA data (default is 'true'), (6) *excludeMet* (optional) – stands for a list of the excludable metabolites as a filter and (7) *flux* (optional) – stands for vector of FBA data of reactions flux distribution (default is vector of 1's characters for no flux). The last 5 arguments are optional and can be unset. The function `drawBy met` has 3 outputs: (1) *directions* – stands for a list of involved reactions depending on the input arguments, (2) *involvedMet* – stands for a list of involved metabolites depending on the input arguments and (3) *deadEnds* – stands for a list of dead-end metabolites depending on the input arguments.

The layout of the network is generated by Bioinformatics Toolbox using hierarchical (default), radial or equilibrium layout engine algorithm, which can be changed in the menu item "tool" of the layout. The automatically generated layout stays the same as long as the scope of representation and directionality of reactions remain unchanged. The layout of automatically generated network can be changed by dragging metabolites or reactions for analysis or publishing needs. Still the new layout cannot be saved for next visualizations. The visual representation of information in the layout (see Fig. 1) allows quick assessment of running processes generally.

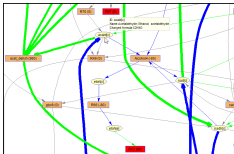


Fig. 1 Zoomed fragment of a large model. Reaction nodes (rectangle) contain IDs and flux rates. Metabolite nodes (ellipse) are marked by IDs. Forward and reverse fluxes (arrows) are green and blue correspondingly, zero fluxes are grey. The thickness of an arrow is proportional to the size of flux. Rectangles of blocked and exchange reactions are marked by red. One node at a time can be chosen by a cursor to see detailed information: (1) ID, name, synonyms and equation for a reaction and (2) ID, name, synonyms and charged formula for a metabolite. Metabolite acetylglutamate selected by cursor for displaying of detailed information. Creation of this figure is described in the "Anticipated results" section, Step 8.

Detailed information about any reaction and metabolite can be requested right on the visualization choosing the node of interest by cursor. To facilitate the analysis the PaintNet creates a list of involved metabolites and a list of the dead-end metabolites (metabolite cannot be produced or consumed caused by gap – missing reaction – in the model) (Thiele and Palsson, 2012) in the visualized part of the reaction network. That is a convenient feature analyzing larger visualizations where it becomes hard to be sure that a particular metabolite is or is not involved just by search of the abbreviation in the picture. The dead-end metabolites are detected according to the S matrix (Palsson, 2008) of the biochemical model and the rates of reactions at steady state according to FBA. Flux rates below 10^{-6} mmol g⁻¹ h⁻¹ are considered to be zero flux rates. To reduce the density of visualization a metabolite filter can be used. Stoichiometric model may have some particular metabolites (H₂O, CO₂, ATP, ADP, NAD⁺, NADPH, NADP⁺ etc.), which are more involved in the metabolism, than other ones. It leads to a very tight interconnectivity and makes it harder to visualize larger models. For this reason both functions of the PaintNet have special argument for the list of excludable metabolites. Each excluded metabolite reduces the number of intersecting curves in the visualization of the model. Debugging of a model can be facilitated in several ways. The visualization shows the interconnections between reactions and metabolites and isolated parts of a network appear very clearly. There are three reasons why isolated reactions can be present in the model: (1) human made mistakes in the model while generating the model, (2) intentionally left gaps in case of no interest in specific metabolic process or (3) knowledge gaps in case of missing biochemical knowledge for the organism. Isolated parts can be related to metabolic dead-ends in the model (Thiele and Palsson, 2012). The PaintNet is able to find the dead-end metabolites and represent them in the visualization (red ellipse). Blocked reactions which cannot carry the fluxes (Thiele and Palsson, 2012) can appear in the model as intentionally blocked reactions (gene knockouts, flux constraints set to 0), however often there are cases when the model is stoichiometrically unbalanced or lower and upper bounds are set incorrectly. The PaintNet can deal with this problem as well by showing the nodes of the blocked reactions in the visualization by red rectangles. The true power of visualization reflects when the user has to find Type II-extreme pathways (Thiele and Palsson, 2012) in the model. Those are stoichiometrically balanced cycles which can have fluxes even when all the substrate sources are cut down. According to FBA data the PaintNet visualizes the network and then it is relatively easy to distinguish cycles by nonzero fluxes marked by green and blue arrows, which in most cases are caused by insufficient constraints of the flux and/or directionality.

PaintNet has been used in the development of stoichiometric models of *Zymomonas mobilis* (Periquin et al., 2012), *Methanococcus marisnigellus* (Richards et al., 2018) and *Kluyveromyces fragilis* (Periquin et al., 2017). PaintNet was used for to develop draft model, to identify network reactions connecting missing outputs to inputs and/or to produce figures for the published manuscript (Aulich and Thiele, 2012), (Dzenderko et al., 2017), (Cortador et al., 2018). It is used also for analysis and published figures in other studies (Kousa et al., 2014). PaintNet is mentioned also as valuable supplement to other COBRA Toolboxes like CRCA (Miao and Veenendaal, 2014). PaintNet has been used also for analysis of networks during building of modeling software (Rove et al., 2012) and supporting software tools (Rubina and Stalitz, 2012), (Mellatou and Stalitz, 2012). PaintNet is also used in a number of doctoral theses in different countries.

MATERIALS

No materials are needed as PaintNet is a software product.

EQUIPMENT

The equipment has to be able to run MATLAB.

EQUIPMENT SETUP

The COBRA toolbox and the Monodomics toolbox are required to use the PaintNet.

PROCEDURE

The PaintNet v1.3 contains two main commands for the visualization purposes:

- `[involvedMetS, deadEnds] = draw_by_run(model, xms, drawMap, direction, initialMet, excludableMet, flux)`
- `[directionFluxes, involvedMetS, deadEnds] = draw_by_map(model, metObj, drawMap, radius, direction, excludableMet, flux)`

Application of command `draw_by_run`

`draw_by_run` can be performed using option **A** in case COBRA model optimization results have to be visualized or option **B** if the interconnections between metabolites in COBRA file have to be visualized (see Fig. 2).

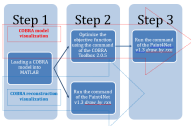


Fig. 2. The scenarios of an application of the command `draw_by_run`.

Before starting the tutorial, initialize the Cobra Toolbox if necessary and set a LP solver.

in initialization box



Constraint-Based Reconstruction and Analysis
The COBRA Toolbox - 2023
Documentation:
<https://cobratoolbox.github.io/cobraToolbox/>

```
% Checking if git is installed ... Done.
% Checking if the repository is tracked using git ... Done.
% Checking if curl is installed ... Done.
% Checking if remote can be reached ... Done.
% Initializing and updating submodules ... Done.
% Adding all the files of The COBRA Toolbox ... Done.
% Define CMake output... set to exp.
% Retrieving models ... Done.
% TranslateSBML is installed and working properly.
% Configuring solver environment variables ...
- [set] SBML_CXX_PATH: C:\Program Files\IBM\ILOG\CPLEX_Studio126\cplex\win64\x86_64_wintel
- [set] GURUBI_PATH : -- set this path manually after installing the solver ( see Installation )
- [set] TROPICAL_PATH: C:\sw\sw64
- [set] MUMPS_PATH : -- set this path manually after installing the solver ( see Installation )
Done.
% Checking available solvers and solver interfaces ... Done.
% Setting default solvers ... Done.
% Saving the MATLAB path ... Done.
- The MATLAB path was saved in the default location.
```

% Summary of available solvers and solver interfaces

Solver	LP	MILP	QP	MQSP	NLP
cplex_direct	active	0	0	0	0
dgblines	active	0	-	-	-
glpk	active	1	1	-	-
gurobi	active	1	1	1	1
linx_glpk	active	0	0	0	-
matlab	active	1	-	-	1
moose	active	0	0	0	-
pico	active	1	-	1	-
quadlines	active	0	-	-	0
linxlab_glpk	active	1	1	1	1
qng	passive	-	-	1	-
linxlab_simgp	passive	-	-	-	1
gurobi_legacy	legacy	0	0	0	0
linx_legacy	legacy	0	-	-	-
lp_solve	legacy	1	-	-	-
scip	legacy	0	0	0	0
Total	-	6	3	4	2

% Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.

```
% You can solve LP problems using: 'glpk' - 'gurobi' - 'matlab' - 'pico' - 'linxlab_glpk' - 'lp_solve'
% You can solve MILP problems using: 'glpk' - 'gurobi' - 'linxlab_glpk'
% You can solve QP problems using: 'gurobi' - 'pico' - 'linxlab_glpk' - 'qng'
% You can solve MQSP problems using: 'gurobi' - 'linxlab_glpk'
% You can solve NLP problems using: 'matlab' - 'linxlab_simgp'
```

```
% Checking for available updates ...
% The COBRA Toolbox is up-to-date.
```

```
changeCobraSolver('gurobi','all',1);
%changeCobraSolver('glpk','all',1);
```

```
% Solver for LP problems has been set to glpk.
% Solver for MILP problems has been set to glpk.
% Solver glpk not supported for problems of type MQSP. Currently used: gurobi
% Solver glpk not supported for problems of type NLP. Currently used: matlab
% Solver glpk not supported for problems of type QP. Currently used: gurobi
```

A. COBRA model visualization

L is a COBRA model has to be loaded into MATLAB. The sample model is available at https://www.ebi.ac.uk/ebipublic/files/COBRA_models/2023/COBRA_2023.zip. Download the file and extract it in a folder of your choice. You can also use your own model.

```
model = xCModel('test_model.xlsx')
```

```
gpc12 flp(s) -> glp(s)
gpc21 atp(s) + gls-0(s) -> glp(s) + adp(s) + h(s)
car12 atp(s) + fru(s) -> flp(s) + adp(s) + h(s)
e22 glp(s) + nadp(s) -> h(s) + gdp(s) + nadh(s)
glp1 glp(s) + nad(s) -> h(s) + gdp(s) + nadh(s)
e23 gdp(s) + h2o(s) -> h(s) + pp1(s)
e24 pp1(s) -> h2o(s) + dgp(s)
R2B1 nadp(s) + pp1(s) -> nadph(s) + e2(s) + rufp-0(s)
ppp1 rufp-0(s) + atp(s) -> flp(s) + glp(s)
ppp1 rufp-0(s) + flp(s) -> flp(s) + xfp(s)
car12 rufp-0(s) -> rufp-0(s)
R2B1 rufp-0(s) -> rufp-0(s)
```

```

g0p01 rmapg[a] := rmapg[a]
g0p02 dpg[a] := var g0p[a] + ppr[a]
g0p03 g0p[a] := var dmap[a]
g0p04 nadd[a] + g0p[a] + p[a] := var h[a] + nadd[a] + 21dpg[a]
g0p05 atp[a] + 3pg[a] := var atp[a] + 21dpg[a]
g0p06 3pg[a] := var 3pg[a]
g0p07 2pg[a] := var 2pg[a]
g0p08 2pg[a] + var h2a[a] + ppg[a]
g0p09 atp[a] + ppr[a] := var atp[a] + h[a] + ppg[a]
ppr000 h[a] + g0p[a] := var aa2[a] + aaat0[a]
aa01 nadd[a] + ppr[a] + aa[a] := var nadd[a] + aa2[a] + aaacc0[a]
aa02aa00 nadd[a] + etah[a] := var h[a] + nadd[a] + aa02aa0[a]
aa01_a00p nadd[a] + h2a[a] + aaat0[a] := var 2 h[a] + nadd[a] + aa[a]
aa02aa0 nadd[a] + aa[a] := var h[a] + nadd[a] + ppr[a]
R000 la0-0[a] + q[a] := var ppr[a] + q02[a]
R020 h[a] + nadd[a] + q[a] := var nadd[a] + q02[a]
R000 h[a] + nadd[a] + q[a] := var nadd[a] + q02[a]
R000 2 q02[a] + aa2[a] := var 2 h2a[a] + 2 q[a]
R000 q02[a] + aa0[a] := var q[a] + aaacc[a]
la002 h2a[a] + aaacc[a] + aaacc[a] := var h[a] + aa[a] + aa2[a]
la002 aa2[a] := var aa[a] + aa0[a]
la002 aa2[a] := var h2a[a] + aa-aaacc0aa[a]
la002 h2a[a] + aa-aaacc0aa[a] := var la0-0[a]
la001 nadd[a] + aa0-0[a] := var nadd[a] + aa2[a] + atp[a]
R000 h[a] + nadd[a] + atp[a] + aa2[a] := var nadd[a] + h2a[a] + g0a-1[a]
g0a00 na0-1[a] := var h2a[a] + aa0[a]
g0a00 na0[a] + na0-1[a] := var nadd[a] + aa2[a] + ppr[a]
g0a00a07 p[a] + aa[a] := var ppg[a] + aa0[a]
g0p00 na0[a] + na0-1[a] := var h[a] + nadd[a] + dmap[a]
R000 h2a[a] + 3pg[a] := var p[a] + g0p0-0[a]
g0p002 atp[a] + g0p0-0[a] := var 3pg[a] + h[a] + 3pg[a]
R020 atp[a] + g0p0[a] := var atp[a] + h[a] + g0p0p[a]
R000 h2a[a] + g0p0p[a] := var p[a] + g0p0[a]
R000 h2a[a] + dmap[a] := var p[a] + aa0[a]
ppg0la02 xyl-0[a] := var xyl0-0[a]
ppg0la02 atp[a] + xyl0-0[a] := var atp[a] + h[a] + aa0p-0[a]
ppg0la02 g0p[a] + x0p[a] := var 3pg[a] + aa0p-0[a]
R000 atp[a] + h2a[a] := var atp[a] + h[a] + p[a]
R000 h[a] + ppr[a] + aaat0[a] := var aa2[a] + aaacc[a]
ppp00 h2a[a] + g0a[a] := var h[a] + g0acc[a]
aa00001 nadd[a] + g0a[a] := var h[a] + nadd[a] + 2dpg0acc[a]
R020 g0a-0[a] + fra[a] := var g0a[a] + aa0-0[a]
R000 g0a-0[a] + q[a] := var g0a[a] + aa0-0[a]
R000 aa0[a] := var aa0p[a] + aa0[a]
R000 g0acc[a] := var aa0p[a] + aa0[a]
R000 g0a-0[a] := var
R007 fra[a] := var
R000 xyl0-0[a] := var
R000 g0p0[a] := var
R000 aa0[a] := var
R000 g0p0-0[a] := var
R002 2dpg0acc[a] := var
R003 g0a-1[a] := var
R004 aa0[a] := var
R000 aa2[a] := var
R000 etah[a] := var
R007 aa2[a] := var
R000 la0-0[a] := var
R000 aa[a] := var
R000 aaat0[a] := var
R020 aa0[a] := var
R020 dmap[a] := var
R020 h2a[a] + aa2[a] := var h[a] + aaat0[a]
R020 aa0-0[a] := var

```

[illegible]

```

setK0G22: (50+1 on())
setInChIString: (50+1 on())
setPubChemID: (50+1 on())
setChEMBLID: (50+1 on())
description: "levi_model.v1a"

```

II. optimisation of the objective function by using the COBRA toolbox command `optimizeCbModel` where the argument `model` corresponds to the COBRA model in the MATLAB workspace.

```

PBAccolution = optimizeCbModel(model)

```

Warnings: & value of class "non.networks.mde.mdein.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warnings: & value of class "non.networks.mde.mdein.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

```

PBAccolution =
  obj: [70x1 double]
  rxns: [70x1 double]
  durs: [50x1 double]
  solvers: 'glpk'
  algorithms: 'default'
  vstart: 2
  wstart: 5
  lines: 0.000
  basis: {}
    w: [70x1 double]
    r: 0
    y: [50x1 double]
    w: [70x1 double]
    w: [70x1 double]

```

This step ensures that a vector of the steady state fluxes `x` will be available for the command `draw_by_rxn`.

```

PBAccolution.x

```

Warnings: & value of class "non.networks.mde.mdein.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warnings: & value of class "non.networks.mde.mdein.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

ans =

```

0
0
0
-1000
1000
0
100
0
0
0
0

```

III. execution of the command `[resolvedMet, deadIndex] = draw_by_rxn(model, rxns, drawMap, direction, initialMet, excludedMet, flux)`

```

[resolvedMet, deadIndex] = draw_by_rxn(model, model.rxns, 'true', 'false', {}, {}, PBAccolution.x)

```

Warnings: & value of class "non.networks.mde.mdein.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warnings: & value of class "non.networks.mde.mdein.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

```

Start_time = 2017_7_14_12_38_38.880
Starting parallel pool (parpool) using the 'local' profile ... Diagram object with 120 nodes and 256 edges.
End_time = 2017_7_14_12_38_04.178
Total_time = 0_0_0_0_0_3.713
InitialMet_metis =
  'Rtp[x]'
  'Glp[x]'
  'Atp[x]'
  'Gls-Glx]'
  'Adp[x]'
  'h[x]'
  'Fou[x]'
  'nadp[x]'
  'Glp[x]'
  'Glp[x]'
  'nadph[x]'
  'nadp[x]'
  'nadh[x]'
  'h2o[x]'

```

```

*ag1(x)
*ag2(x)
*ag3(x)
*ag4(x)
*ag5(x)
*ag6(x)
*ag7(x)
*ag8(x)
*ag9(x)
*ag10(x)
*ag11(x)
*ag12(x)
*ag13(x)
*ag14(x)
*ag15(x)
*ag16(x)
*ag17(x)
*ag18(x)
*ag19(x)
*ag20(x)
*ag21(x)
*ag22(x)
*ag23(x)
*ag24(x)
*ag25(x)
*ag26(x)
*ag27(x)
*ag28(x)
*ag29(x)
*ag30(x)
*ag31(x)
*ag32(x)
*ag33(x)
*ag34(x)
*ag35(x)
*ag36(x)
*ag37(x)
*ag38(x)
*ag39(x)
*ag40(x)
*ag41(x)
*ag42(x)
*ag43(x)
*ag44(x)
*ag45(x)
*ag46(x)
*ag47(x)
*ag48(x)
*ag49(x)
*ag50(x)
*ag51(x)
*ag52(x)
*ag53(x)
*ag54(x)
*ag55(x)
*ag56(x)
*ag57(x)
*ag58(x)
*ag59(x)
*ag60(x)
*ag61(x)
*ag62(x)
*ag63(x)
*ag64(x)
*ag65(x)
*ag66(x)
*ag67(x)
*ag68(x)
*ag69(x)
*ag70(x)
*ag71(x)
*ag72(x)
*ag73(x)
*ag74(x)
*ag75(x)
*ag76(x)
*ag77(x)
*ag78(x)
*ag79(x)
*ag80(x)
*ag81(x)
*ag82(x)
*ag83(x)
*ag84(x)
*ag85(x)
*ag86(x)
*ag87(x)
*ag88(x)
*ag89(x)
*ag90(x)
*ag91(x)
*ag92(x)
*ag93(x)
*ag94(x)
*ag95(x)
*ag96(x)
*ag97(x)
*ag98(x)
*ag99(x)
*ag100(x)

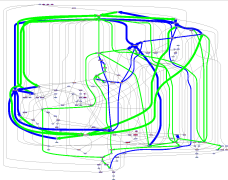
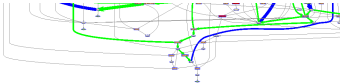
```

```

Dead_paths =
*ag1(x)
*ag2(x)
*ag3(x)

```





B. COBRA reconstruction visualization

1. a COBRA model has to be loaded into MATLAB.

```
model = sbModel('test_model.sbo')
```

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdr.mdrin.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

operation is the indexed value `SizeT`, but in a future release, it will be an error.

```

gfp12 ffp[s]    ~r gfp[s]
gfp25 atp[s] + gfu-0[s] ~r gfp[s] + adp[s] + h[s]
csh12 atp[s] + fra[s]    ~r ffp[s] + adp[s] + h[s]
ed2 gfp[s] + nady[s]    ~r h[s] + gfp[s] + nadyh[s]
ghp1 gfp[s] + nad[s]    ~r h[s] + gfp[s] + nadh[s]
ed1 gfp[s] + h2a[s]    ~r h[s] + pgt[s]
ed1 pgt[s]    ~r h2a[s] + dpg[s]
R03 nady[s] + pgt[s] ~r nadyh[s] + csh2[s] + rufp-0[s]
ppp1 nufp-0[s] + atp[s] ~r ffp[s] + gfp[s]
ppp1 nufp-0[s] + rfp[s] ~r gfp[s] + rfp[s]
csh2 rufp-0[s] ~r nufp-0[s]
R04 rufp-0[s] ~r rfp[s]

ppp1 dpg[s] ~r gfp[s] + ppr[s]
gfp12 gfp[s] ~r dthp[s]
gfp12 nad[s] + gfp[s] + pi[s] ~r h[s] + nadh[s] + 13dpg[s]
gfp12 atp[s] + 3pg[s] ~r adp[s] + 13dpg[s]
gfp12 2pg[s] ~r npp[s]

gfp121 2pg[s] ~r h2a[s] + ppg[s]
gfp14 atp[s] + ppr[s] ~r adp[s] + h[s] + ppg[s]
ppr_dou h[s] + ppr[s] ~r csh2[s] + asah[s]
ed1 nad[s] + ppr[s] + csh[s] ~r nadh[s] + csh2[s] + accash[s]
A1cshah14 nad[s] + etah[s] ~r h[s] + nadh[s] + asah[s]
acsh_dshg nad[s] + h2a[s] + asah[s] ~r 2 h[s] + nadh[s] + as[s]
acsh_dshg nady[s] + h2a[s] + asah[s] ~r 2 h[s] + nadyh[s] + as[s]
cshah1 nad[s] + lac-0[s] ~r h[s] + nadh[s] + ppr[s]

R26 lac-0[s] + q[s] ~r ppr[s] + qh2[s]
R27 h[s] + nadh[s] + q[s] ~r nad[s] + qh2[s]
R28 h[s] + nadyh[s] + q[s] ~r nady[s] + qh2[s]
R29 2 qh2[s] + a2[s] ~r 2 h2a[s] + 2 q[s]
R301 qh2[s] + fun[s] ~r q[s] + uash[s]
tsh21 h2a[s] + accash[s] + ana[s] ~r h[s] + ana[s] + csh[s]
tsh2 csh[s] ~r ac[s] + ana[s]
tsh21 csh[s] ~r h2a[s] + csh-accumulate[s]
tsh21 h2a[s] + csh-accumulate[s] ~r csh-0[s]
tsh21 nady[s] + csh-0[s] ~r nadyh[s] + csh2[s] + atq[s]
R33 h[s] + nadyh[s] + atq[s] + shd[s] ~r nady[s] + h2a[s] + gfu-4[s]
tsh22 nsh-4[s] ~r h2a[s] + fun[s]
gfp10 nad[s] + nsh-4[s] ~r nadh[s] + csh2[s] + ppr[s]
shdshah7 pi[s] + ana[s] ~r ppg[s] + hshd[s]
ppp1 nady[s] + gfp1p[s] ~r h[s] + nadyh[s] + dthp[s]
R06 h2a[s] + hpg[s] ~r pi[s] + gfp-0[s]
Carshah127 atp[s] + gfp-0[s] ~r adp[s] + h[s] + 3pg[s]
R32 atp[s] + gfp1[s] ~r adp[s] + h[s] + gfp1p[s]
R33 h2a[s] + gfp1p[s] ~r pi[s] + gfp1[s]
R34 h2a[s] + dthp[s] ~r pi[s] + dth[s]
ppp1ac2 up-0[s] ~r up-0[s]
ppp1ac1 atp[s] + up-0[s] ~r adp[s] + h[s] + nufp-0[s]
ppp1 gfp[s] + rfp[s] ~r ffp[s] + atp[s]
R08 atp[s] + h2a[s] ~r adp[s] + h[s] + pi[s]
R09 h[s] + ppr[s] + asah[s] ~r csh2[s] + accash[s]
ppp1 h2a[s] + gfu[s] ~r h[s] + gfu[s]
cshah17 nady[s] + gfu[s] ~r h[s] + nadyh[s] + 2dpgfun[s]
R32 gfu-0[s] + fra[s] ~r gfu[s] + shd-0[s]
R33 gfu-0[s] + q[s] ~r qh2[s] + gfu[s]
ppp1 atp[s] + gfu[s] ~r adp[s] + h[s] + pgt[s]
R35 gfu[s] ~r
R36 gfu-0[s] ~r
R37 fra[s] ~r
R38 up-0[s] ~r
R39 gfu[s] ~r
R40 ana[s] ~r
R41 gfu-0[s] ~r
R42 2dpgfun[s] ~r
R43 gfu-4[s] ~r
R44 shd[s] ~r
R45 a2[s] ~r
R46 etah[s] ~r
R47 csh2[s] ~r
R48 lac-0[s] ~r
R49 ac[s] ~r
R50 asah[s] ~r
R51 accash[s] ~r
R52 dth[s] ~r
R53 h2a[s] + csh2[s] ~r h[s] + hshd[s]
R54 shd-0[s] ~r

model =
    rami: (7b1 ceU)
    li: (8b75 double)
    lsi: (7b1 double)
    shi: (7b1 double)
    ci: (7b1 double)

    nsh: (8b1 ceU)
    li: (8b1 double)

    rufu: (7b1 ceU)
    gfu: (8b1 ceU)
    -----

```



```

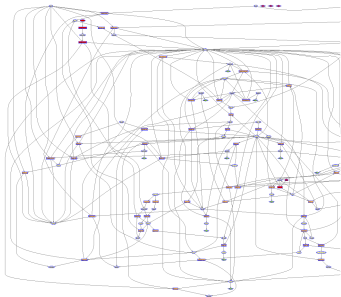
'xyle-Q(x)'
'mim(x)'
'g1(x)'
'g2(x)'
'2abg1(x)'
'ab-Q(x)'

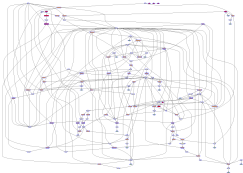
```

```

deadEnds =
'g1-Q(x)'
'g2(x)'
'xyl-Q(x)'

```





In this case in the brackets in the rectangles are shown characters, indicating that fluxes at steady state were not calculated. This approach is useful in case of reconstruction visualization where fluxes are not calculated. The arrows in the end of the edges point on the forward directions of the reactions in the COBRA model.

Input arguments of the command `draw_by_run`

The command `draw_by_run` has several input arguments – `model`, `ons`, `drawMap`, `direction`, `initialMet`, `excludeMet`, and `flux`. The last 5 arguments are optional; it means that the algorithm of the command `draw_by_run` uses default values of those arguments, so user can ignore them if additional functionality is not actual.

1. argument `model`

This argument stands for a COBRA model in the MATLAB workspace.

6. argument `rule`

This argument represents a list of reactions from a COBRA model separated by a comma. The layout of map depends on this list, as a result if new abbreviation is added or some deleted in the list the layout will change as well. To prevent layout change by potential mistakes and save time by not creating the list every time from scratch it is possible to create a cell type vector in the MATLAB workspace that contains the static abbreviations of the reactions in the COBRA model and use it as argument `rule`.

```
rule = {'glyc12', 'glyc23', 'carb02', 'w02'}
```

```

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings & value of class "var.methodss.mde.mdein.X2Indefinite()" was
indeed with no subscripts specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

var = c(
  "var123" "var234" "var345" "var4"

```

```
[involvedNode, deadNode] = draw_by_node(node, row)
```

```

Warnings: Value of class "com.mathworks.mde.index.XIndexOptions" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Warnings: Value of class "com.mathworks.mde.index.XIndexOptions" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.

Start_time = 2017_7_14_12_38_35.760

Starting parallel pool (parpool) using the "local" profile ...
End_time = 2017_7_14_12_38_36.001
Total_time = 0_0_0_0_0_0.240

functionResults =
    'fip(a)'
    'fip(a)'
    'uip(a)'
    'gip=Q(a)'
    'uip(a)'
    'h(a)'
    'fip(a)'
    'uip(a)'
    'uip(a)'
    'gip(a)'
    'uip(a)'

dealResults =
    'gip=Q(a)'
    'fip(a)'
    'uip(a)'
    'gip(a)'
    'uip(a)'

```

Another example is a list of reaction in the CCSSRA model which can be accessed by `model.rxn`. It illustrates all CCSSRA model

```
[involvedness, demand] = drawByRejection, model.mcmc)
```

Warnings: 1 value of class 'class "unlabeled" not within X20000000' was
 ignored with no subscripts specified. Currently the result of this
 operation is the ignored value 0.000000, but in a future release, it will
 be an error.

Warnings: 1 value of class 'class "unlabeled" not within X20000000' was
 ignored with no subscripts specified. Currently the result of this
 operation is the ignored value 0.000000, but in a future release, it will
 be an error.

Warnings: 1 value of class 'class "unlabeled" not within X20000000' was
 ignored with no subscripts specified. Currently the result of this


```

'max':
'tan-0[x]'
'q[x]'
'q2[x]'
'a2[x]'
'fuu[x]'
'uuu[x]'
'uau[x]'
'u11[x]'
'uuu-acumilate[x]'
'tan-0[x]'
'ahq[x]'
'ahd[x]'
'g1u-1[x]'
'mu1-1[x]'
'uuu0[x]'
'g1y2u[x]'
'g1y-0[x]'
'g1y[x]'
'dhu[x]'
'xyl-0[x]'
'xylu-0[x]'
'muu[x]'
'g1u[x]'
'g1uu[x]'
'2dglu[x]'
'ahd-0[x]'

```

```

deadEnds =
'g1u-0[x]'
'a2[x]'
'xyl-0[x]'

```

III. optional argument direction

It is a string type variable that can take value of 'lstruc', 'sub', 'prod' or 'both' (default is 'lstruc') indicating a direction for the algorithm of the command draw_by_out.

```
[Involved_nets, Dead_end] = draw_by_out(nodes, mode_name, 'false', 'struc')
```

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings & value of class "iron.networks.mdn.mdnin.XCedInflam" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Start_time = 2017_7_10_12_18_16.388

Starting parallel pool (parpool) using the 'local' profile

End_time = 2017_7_10_12_18_16.532

Total_time = 0_0_0_0_0_1.143

```

Involved_nets =
'fhp[x]'
'ghp[x]'
'ahq[x]'
'g1u-0[x]'
'ahp[x]'
'h[x]'
'fuu[x]'
'uau[x]'
'g1hp[x]'

```




This metabolite is represented as green ellipse on the map (see Fig. 3) and this feature is essential for the command `draw_by_net` of the `PaintNet` v1.3 because the command `draw_by_net` is calling out the command `draw_by_on` and passing the argument `initialMet`.



Fig. 3. The metabolite `apc` as initial metabolite on the map of the COBRA model.

III. optional argument `excludedMet`

This argument represents a list of metabolites (default is empty) that will be excluded from the visualization map of the COBRA model in form of the abbreviations of the metabolites separated by a comma. To save time by not loading the list every time from scratch it is possible to create a cell-type vector in the MATLAB workspace that contains the static abbreviations of the metabolites in the COBRA model and use it as argument `excludedMet`.

```
excludedMet = {'apc', 'adp', 'adp2', 'h2o'}
```




Fig. 4. An example of the map of the COBRA model. Full scope.



Fig. 5. An example of the map of the COBRA model. The metabolite h_2 is excluded from map which reduce the number of edges by 26.

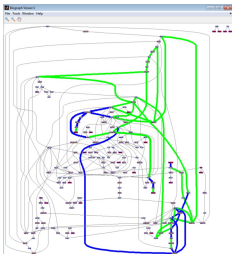


Fig. 8. An example of the map of the COBRA model. The metabolites $\eta[c]$ and $\text{hdg}[c]$ are excluded from map which reduce the number of edges by $26 \times 17 = 442$.

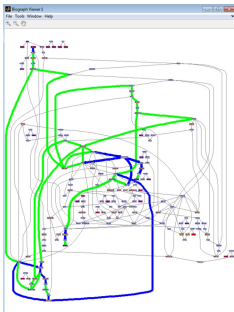


Fig. 7. An example of the map of the COBRA model. The metabolites $\eta[c]$, $\text{tzh}[c]$ and $\text{apj}[c]$ are excluded from map which reduce the number of edges by $20+17+6=43$.

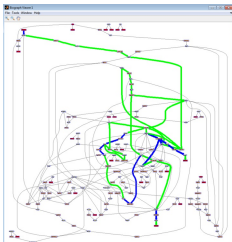


Fig. 8. An example of the map of the COBRA model. The metabolites $\eta[c]$, $\text{f2b}[c]$, $\text{ap}[c]$ and $\text{aap}[c]$ are excluded from map which reduce the number of edges by $28+17+8+8=61$.

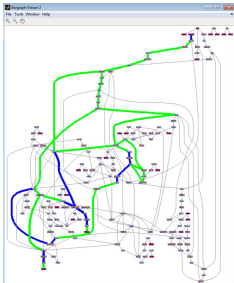


Fig. 9. An example of the map of the COBRA model. The metabolites $\eta[c]$, $\eta[h[c]]$, $\text{asp}[c]$, $\text{asp}[h]$ and $\text{nad}[c]$ are excluded from map which reduce the number of edges by $28+17+9+9+8=63$.

1000, optional argument flux

It is a double-type Nx1 size-vector of fluxes of reactions where N is number of reactions (default is vector of 5). This vector is calculated during the optimization of the objective function and can be accessed through the result of the optimization command.

伊藤昌弘と山田隆弘 主編 978-4-479-26100-0

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rsc.networks.mdr.mdrain.XCellModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

ans =

```

0
0
0
-1880
1880
0
180
0
0
0

```

Output of the command draw_by_rsc

The command draw_by_rsc has two output vectors in the result: involvedMetes and drainIds.

1. the vector involvedMetes

It is a cell type vector that contains a list of the involved metabolites in the specified reactions (see Fig. 10).

involved_mets		
	1	2
1	hco2	
2	gln	
3	gln	
4	gln	
5	gln	
6	hco2	
7	hco2	
8	malp	
9	gln	
10	malp	
11	malp	
12	malp	
13	hco2	
14	gln	
15	gln	
16	malp	
17	malp	
18	malp	
19	gln	
20	gln	
21	gln	
22	gln	
23	malp	
24	malp	
25	malp	
26	hco2	
27	hco2	

Fig. 10. An example of the list of the involved metabolites.

ii. the vector `deadEnds`

It is also a cell-type vector but it contains a list of the dead-end metabolites in the specified reactions (see Fig. 11).

DeadEnds	Cell
1	fruc
2	pyr-fuc

Fig. 11. An example of the list of the dead-end metabolites.

Application of command `draw_by_met`

`draw_by_met` can be used can be performed using **option A** in case COBRA model with optimization results have to be visualized or **option B** if the interconnections between metabolites in COBRA file have to be visualized (see Fig. 12).

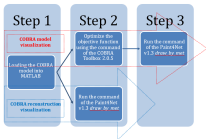


Fig. 12. The scenarios of an application of the command `draw_by_met`.

A. COBRA model visualization

1. a COBRA model has to be loaded into MATLAB.

```
model = xlbioset('test_model.xls');
```

Warning: A value of class "non.networks.mdm.mdm.XCellNetItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "non.networks.mdm.mdm.XCellNetItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```

gln12 flp(s) ~> gfp(s)
gln15 atp(s) + gls-0(s) ~> gfp(s) + atp(s) + h(s)
car12 atp(s) + fru(s) ~> flp(s) + atp(s) + h(s)
e02 gfp(s) + nadp(s) ~> h(s) + gfp(s) + nadh(s)
gfp1 gfp(s) + nad(s) ~> h(s) + gfp(s) + nadh(s)
e01 gfp(s) + h2o(s) ~> h(s) + py(s)
e02 py(s) ~> h2o(s) + dgp(s)
R00 nadp(s) + py(s) ~> nadh(s) + co2(s) + ruf-0(s)
pyr1 ruf-0(s) + atp(s) ~> flp(s) + gfp(s)
pyr2 ruf-0(s) + flp(s) ~> gfp(s) + stp(s)
car12 ruf-0(s) ~> ruf-0(s)
R00 ruf-0(s) ~> flp(s)
pyr3 dgp(s) ~> gfp(s) + pyr(s)
gln15 gfp(s) ~> dhp(s)
gln12 nad(s) + gfp(s) + pi(s) ~> h(s) + nadh(s) + 13dgp(s)
gln15 atp(s) + 3pg(s) ~> atp(s) + 13dgp(s)
gln17 2pg(s) ~> 3pg(s)
gln15 2pg(s) ~> h2o(s) + pgp(s)
gln15 atp(s) + pyr(s) ~> atp(s) + h(s) + pgp(s)
pyr_dms h(s) + pyr(s) ~> co2(s) + aso(s)
  
```

```

R01 nad[s] = pyr[s] + cna[s] -> nadh[s] + cna2[s] + acnaa[s]
R02 nadh[s] + etah[s] -> h[s] + nadh[s] + acald[s]
acna_dhpy nad[s] + h2a[s] + acald[s] -> 2 h[s] + nadh[s] + ac[s]
acna_dhpy nadp[s] + h2a[s] + acald[s] -> 2 h[s] + nadph[s] + ac[s]
cnaoh nad[s] + lac-0[s] -> h[s] + nadh[s] + pyr[s]
R26 lac-0[s] + q[s] -> pyr[s] + qh2[s]
R27 h[s] + nadh[s] + q[s] -> nad[s] + qh2[s]
R28 h[s] + nadph[s] + q[s] -> nadp[s] + qh2[s]
R29 2 qh2[s] + a2[h] -> 2 h2a[s] + 2 q[s]
R30 qh2[s] + fun[s] -> q[s] + succ[s]
trc12 h2a[s] + acnaa[s] + ana[s] -> h[s] + ana[s] + cit[s]
trc7 cit[s] -> ac[s] + ana[s]
trc11 cit[s] -> h2a[s] + cit-acuminate[s]
trc12 h2a[s] + cit-acuminate[s] -> succ-0[s]
trc11 nadp[s] + succ-0[s] -> nadph[s] + succ2[s] + atp[s]
R35 h[s] + nadph[s] + atp[s] + shd[s] -> nadp[s] + h2a[s] + glu-4[s]
trc10 succ-4[s] -> h2a[s] + fun[s]
gls08 nad[s] + mal-4[s] -> nadh[s] + succ2[s] + pyr[s]
atp0ark7 pi[s] + ana[s] -> prep[s] + hna3[s]
gls3 nadp[s] + glp3p[s] -> h[s] + nadph[s] + atp[s]
R48 h2a[s] + 3pg[s] -> pi[s] + glp3-0[s]
Cofact127 atp[s] + glp3-0[s] -> adp[s] + h[s] + 3pg[s]
R52 atp[s] + glp3[s] -> adp[s] + h[s] + glp3p[s]
R53 h2a[s] + glp3p[s] -> pi[s] + glp3[s]
R54 h2a[s] + atp[s] -> pi[s] + dha[s]
perg1ac2 xyl-0[s] -> xyl-0[s]
perg1ac1 atp[s] + xyl-0[s] -> adp[s] + h[s] + xylp-0[s]
pyr0 glp[s] + x3p[s] -> 3pg[s] -> adp[s] + atp[s]
R49 atp[s] + h2a[s] -> adp[s] + h[s] + pi[s]
R49 h[s] + pyr[s] + acald[s] -> succ2[s] + acald[s]
pyr0 h2a[s] + glu[s] -> h[s] + gluco[s]
sucmet3 nadp[s] + gluco[s] -> h[s] + nadph[s] + 2ahg[con]
R52 glu-0[s] + fru[s] -> glu-4[s] + shd-0[s]
R53 glu-0[s] + q[s] -> qh2[s] + glu-4[s]
pyr0 atp[s] + gluco[s] -> adp[s] + h[s] + pg1[s]
R55 gluco[s] ->
R56 glu-0[s] ->
R57 fru[s] ->
R58 xyl-0[s] ->
R59 glp[s] ->
R60 succ[s] ->
R61 glp3-0[s] ->
R62 2ahg[con] ->
R63 glu-4[s] ->
R64 shd[s] ->
R65 a2[s] ->
R66 etah[s] ->
R67 succ2[s] ->
R68 lac-0[s] ->
R69 ac[s] ->
R70 acald[s] ->
R71 acna[s] ->
R72 dha[s] ->
R73 h2a[s] + succ2[s] -> h[s] + hna3[s]
R74 shd-0[s] ->
model =
  names: (70x1 colU)
  li: (60x75 double)
  lbs: (70x1 double)
  ub: (70x1 double)
  ci: (70x1 double)
  neto: (60x1 colU)
  li: (60x1 double)
  rules: (70x1 colU)
  genes: (60x1 colU)
  cnames: -1
  cnames: (60x1 char)
  rxnGeneOid: (70x60 double)
  rxnNames: (70x1 colU)
  subSystem: (70x1 colU)
  netNames: (60x1 colU)
  grNames: (70x1 colU)
  confidenceScores: (70x1 double)
  rxnOidNames: (70x1 colU)
  rxnOidNames: (70x1 colU)
  rxnOidNames: (70x1 colU)
  comp: "s"
  compNames: ("Cytosol or Cytosol, Extra-organism or Cytosol, Extra-organism, Periplasm")
  netFormulas: (60x1 colU)
  netCharges: (60x1 double)
  netIons: (60x1 colU)
  netK0022: (60x1 colU)
  netCh310rings: (60x1 colU)
  netPulsChemE0: (60x1 colU)
  netC00222: (60x1 colU)
  descriptions: "lnc1_model.xls"

```

```
PRASolution = optimizeMode(model)
```

Warning: A value of class "non.networks.mdr.indain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "non.networks.mdr.indain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
PRASolution =  
  fulls [75x1 double]  
  obj 0  
  roots [75x1 double]  
  durs [58x1 double]  
  labels "gph"  
  algorithm "default"  
  state 1  
  origState 5  
  times 0.0070  
  basis {}  
  w [75x1 double]  
  r 0  
  y [58x1 double]  
  u [75x1 double]  
  v [75x1 double]
```

This step ensures that a vector of the steady state fluxes x will be available for the command `draw_by_net`.

```
PRASolution.x
```

Warning: A value of class "non.networks.mdr.indain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "non.networks.mdr.indain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
ans =
```

```
0  
0  
0  
-1000  
1000  
0  
100  
0  
0  
0  
0
```

At execution of the command `[directionPars, involvedMetS, deadEnds] = draw_by_net(model, metRdx, drawMap, radius, direction, excludeMetS, flag)`.

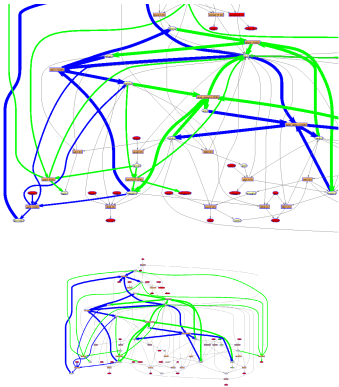
```
[directionPars, involvedMetS, deadEnds] = draw_by_net(model, {'etab(c)'} , "true", 2, 'true', {''} , PRASolution.x)
```

Warning: A value of class "non.networks.mdr.indain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "non.networks.mdr.indain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
Start_line = 2017_7_14_12_38_7_437  
Starting parallel pool (parpool) using the 'local' profile --- Biograph object with 75 nodes and 100 edges.  
End_line = 2017_7_14_12_38_10_093  
Total_time = 0_0_0_0_3_000  
directionPars =
```

```
'Alcohol12'  
'800'  
'glyc11'  
'carb12'  
'e02'  
'glut1'  
'e03'  
'glyc14'  
'glyc15'  
'pyr_005'  
'e05'  
'acet_004y'  
'acet_004y1'  
'carb01'  
'027'  
'028'  
'luc12'  
'031'  
'glyc8'  
'glu13'  
'Default12'  
'042'  
'pent1013'  
-----
```

B. COBRA reconstruction visualization

Load a COBRA model has to be loaded into MATLAB.

```
model = xlcobra('test_model.xls')
```

Warning: & value of class "var.networks.mln.index.XCellidItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "var.networks.mln.index.XCellidItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "var.networks.mln.index.XCellidItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "var.networks.mln.index.XCellidItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "var.networks.mln.index.XCellidItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mlm.index.XEmbedding" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
gfc12 ffp[s] ~v gfp[s]
gfc13 atp[s] + gfc0[s] ~v gfp[s] + atp[s] + h[s]
cac12 atp[s] + fru[s] ~v ffp[s] + atp[s] + h[s]
e02 gfp[s] + nady[s] ~v h[s] + gfp[s] + nady[s]
gfp0 gfp[s] + nad[s] ~v h[s] + gfp[s] + nad[s]
e01 gfp[s] + h2u[s] ~v h[s] + pgf[s]
e02 pgf[s] ~v h2u[s] + dgp[s]
R03 nady[s] + pgf[s] ~v nady[s] + e02[s] + rufp0[s]
pgf0 rufp0[s] + atp[s] ~v ffp[s] + gfp[s]
pgf1 rufp0[s] + rfp[s] ~v gfp[s] + sfp[s]
cac02 rufp0[s] ~v rufp0[s]
R04 rufp0[s] ~v rfp[s]
pgp0 dgp[s] ~v gfp[s] + ppr[s]
gfc0E gfp[s] ~v dmap[s]
gfc01 nad[s] + gfp[s] + pi[s] ~v h[s] + nad[s] + l3dpg[s]
gfc04 atp[s] + 3pg[s] ~v atp[s] + l3dpg[s]
gfc07 3pg[s] ~v 3pg[s]
gfc03 3pg[s] ~v h2u[s] + ppg[s]
gfc04 atp[s] + ppr[s] ~v atp[s] + h[s] + ppg[s]
ppr0es h[s] + ppr[s] ~v e02[s] + asaf[s]
e01 nad[s] + ppr[s] + e0a[s] ~v nad[s] + e02[s] + asoan[s]
l3asaf0 nad[s] + stah[s] ~v h[s] + nad[s] + asaf[s]
asaf_dhgy nad[s] + h2u[s] + asaf[s] ~v 2 h[s] + nad[s] + as[s]
asaf_dhgy nady[s] + h2u[s] + asaf[s] ~v 2 h[s] + nady[s] + as[s]
cac01 nad[s] + lac0[s] ~v h[s] + nad[s] + ppr[s]
R20 lac0[s] + q[s] ~v ppr[s] + q02[s]
R21 h[s] + nad[s] + q[s] ~v nad[s] + q02[s]
R20 h[s] + nady[s] + q[s] ~v nady[s] + q02[s]
R20 2 q02[s] + e2[s] ~v 2 h2u[s] + 2 q[s]
R01 q02[s] + fua[s] ~v q[s] + asaf[s]
l3af0 h2u[s] + asoan[s] + asaf[s] ~v h[s] + e0a[s] + e10[s]
l3af e10[s] ~v as[s] + asaf[s]
l3af0 e10[s] ~v h2u[s] + e10-asaf[s]
l3af0 h2u[s] + e10-asaf[s] ~v e100[s]
l3af1 nady[s] + e100[s] ~v nady[s] + e02[s] + atp[s]
R01 h[s] + nady[s] + atp[s] + e04[s] ~v nady[s] + h2u[s] + gfc0[s]
l3af0 e04[s] + asaf[s] + fua[s]
gfc0E nad[s] + asaf[s] ~v nad[s] + e02[s] + ppr[s]
af0ac07 pi[s] + e0a[s] ~v ppr[s] + h01[s]
R01 nady[s] + gfc03[s] ~v h[s] + nady[s] + dmap[s]
R01 h2u[s] + 3pg[s] ~v pi[s] + gfc0[s]
C01ac07 atp[s] + gfc0[s] ~v atp[s] + h[s] + gfc03[s]
R02 atp[s] + gfc03[s] ~v atp[s] + h[s] + gfc03[s]
R02 h2u[s] + gfc03[s] ~v pi[s] + gfc0[s]
R04 h2u[s] + dmap[s] ~v pi[s] + dmap[s]
pgg1ac2 sfp0[s] ~v sfp0[s]
pgg1ac1 atp[s] + sfp0[s] ~v atp[s] + h[s] + sfp0[s]
pgp0 gfp[s] + sfp[s] ~v ffp[s] + atp[s]
R01 atp[s] + h2u[s] ~v atp[s] + h[s] + pi[s]
R01 h[s] + ppr[s] + asaf[s] ~v e02[s] + asaf[s]
pgp0 h2u[s] + gfc0[s] ~v h[s] + gfc0[s]
asaf07 nady[s] + gfc0[s] ~v h[s] + nady[s] + 2dpg[s]
R02 gfc0[s] + fru[s] ~v gfc0[s] + e10[s]
R01 gfc0[s] + q[s] ~v q02[s] + gfc0[s]
pgp0 atp[s] + gfc0[s] ~v atp[s] + h[s] + pgf[s]
R01 gfc0[s] ~v
R01 gfc0[s] ~v
... ..
```



```

R01 t1x[x] ~v
R02 aq1-Q[x] ~v
R03 g1y[x] ~v
R04 xax[x] ~v
R05 g1y-R[x] ~v
R06 2ahg[x] ~v
R07 g1w-L[x] ~v
R08 shd[x] ~v
R09 a2[x] ~v
R10 etah[x] ~v
R11 aa2[x] ~v
R12 lau-Q[x] ~v
R13 ac[x] ~v
R14 acaal[x] ~v
R15 acac[x] ~v
R16 dha[x] ~v
R17 h2e[x] + aa2[x] ~v h[x] + haa2[x]
R18 shd-Q[x] ~v

model =
  names: (Tb1: cell)
  li: (Sb1: double)
  lka: (Tb1: double)
  uhi: (Tb1: double)
  ci: (Tb1: double)
  acfci: (Sb1: cell)
  li: (Sb1: double)
  rules: (Tb1: cell)
  gpmci: (Sb1: cell)
  acpmci: -1
  cpmci: (Sb1: cell)
  ractonefci: (Tb1: double)
  ractlami: (Tb1: cell)
  ractlami: (Tb1: cell)
  ractlami: (Tb1: cell)
  ractlami: (Sb1: cell)
  ractlami: (Tb1: cell)
  confoundlami: (Tb1: double)
  ractlami: (Tb1: cell)
  ractlami: (Tb1: cell)
  ractlami: (Tb1: cell)
  compi: "a"
  compi: ("Cytosol or Cytosol, Extra-organism or Cytosol, Extra-organism, Periplase")
  netFormci: (Sb1: cell)
  netChargci: (Sb1: double)
  netlami: (Sb1: cell)
  netK0022: (Sb1: cell)
  netInChIci: (Sb1: cell)
  netPubChIci: (Sb1: cell)
  netK0022: (Sb1: cell)
  descriptions: "level_model.v1a"

```

ii. execution of the command `[directionFlow, involvedMets, deadEnds] = draw_by_net(model, netNode, drawType, radius, direction, excludeMets, flux)`

```
[directionFlow, involvedMets, deadEnds] = draw_by_net(model, {'t1x[a]'}, "flow", 3)
```

Warning: A value of class "com.networks.mde.mdein.XCellFlow" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "com.networks.mde.mdein.XCellFlow" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Start_time = 2017_7_14_12_18_12.816

Starting parallel post (parpool) using the 'local' profile ... Biograph object with 76 nodes and 103 edges.

End_time = 2017_7_14_12_18_18.817

Total_time = 8_8_8_8_8.000

directionFlow =

't1x[a]'

'R01'

'g1y21'

'carh12'

'ed'

'g1y4'

'ed'

'g1y12'

'g1y14'

'pyr_dec'

'ed'

'acel_dabg'

'acel_dabg'

'carh12'

'R27'

'R28'

't1x12'

'R33'

'g1y18'

'g1y13'

'Defac127'

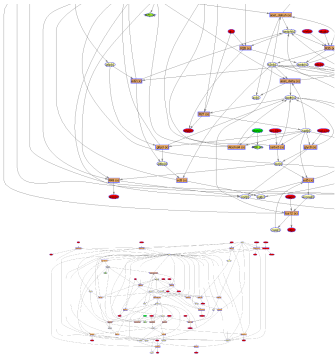
'R42'

'pmg1a13'

'R48'

'R49'

'pyrR'



In this case in the brackets in the rectangles are shown characters, indicating that fluxes at steady state were not calculated. This approach is useful in case of reconstruction visualization where fluxes are not calculated.

Input arguments of the command `draw_by_net`

The command `draw_by_net` has several input arguments – `model`, `metIds`, `drawMap`, `radius`, `direction`, `excludesIds`, and `flux`. The last 5 are optional, it means that the algorithm of the command `draw_by_net` uses default values of those arguments, so user can ignore them if additional functionality is not necessary.

1. argument `model`

This argument stands for a COBRA model in the MATLAB workspace.

2. argument `metIds`

It is a cell type variable that can take a value that represents the abbreviation of a metabolite in a COBRA model. This argument is the start point for the algorithm of the command `draw_by_net` for visualization.

```
[directedEdges, involvedMetIds, deadMetIds] = draw_by_net(model, {'ethanol'})
```



```

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnMdlm" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.
Warnings & value of class "rnn.networks.mdn.mdnin.XCnnMdlm" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.
Warnings & value of class "rnn.networks.mdn.mdnin.XCnnMdlm" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.
Warnings & value of class "rnn.networks.mdn.mdnin.XCnnMdlm" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.
Warnings & value of class "rnn.networks.mdn.mdnin.XCnnMdlm" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.
Warnings & value of class "rnn.networks.mdn.mdnin.XCnnMdlm" was
indexed with no subscript specified. Currently the result of this
operation is the indexed value itself, but in a future release, it will
be an error.
Start_time = 2017_7_14_12_38_17.888
Starting parallel_junk (parpool) using the "local" profile .... Rnigraph object with 7 nodes and 8 edges.
End_time = 2017_7_14_12_38_17.748
Total_time = 0_0_0_0_0_0.66
directionEdges =
'klnshd4'
'888'

localMdlmEris =
'k[x]'
'nat[x]'
'natd[x]'
'natat[x]'
'etab[x]'

globalEris =
'k[x]'
'nat[x]'
'natd[x]'
'natat[x]'

```



The argument radius indicates the depth of an analysis of the initial metabolite (the argument `metABdr`) and it is tightly connected to the optional argument `direction`. For example, if user is interested in the substrates of ethanol, the user can analyse substrates step-by-step starting from the first reactions where the argument radius is equal to 1 and moving to the next reactions by increasing the value of the argument radius (see Fig. 13 and Fig. 14).

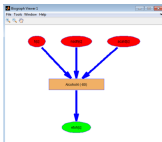


Fig. 13. Example where the argument radius = 1 (distance = one reaction from initial metabolite *ethanol*). In the reaction *AlcoholH* the metabolites *h3c*, *h2nch3* and *acetylch3* are consumed and the metabolite *ethanol* is produced. The flux rate is -80 mmol/g-17h-1 that indicates that reaction is going backwards.

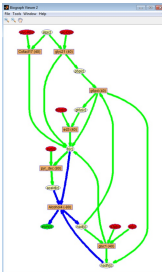


Fig. 14. Example where the argument radius = 2 (distance = two reactions from initial metabolite *ethanol*). The metabolites *glyc-6p*, *glc-6p*, *ndp*, *pyr*, *gdp* and *pip* are consumed and the metabolite *ethanol* is produced.

The algorithm of the command *draw_by_mer* interconnects all involved metabolites according to the stoichiometric matrix of a COBRA model. The important point to understand correctly is imbalance of the rates of fluxes in case of partial network. The algorithm of the command *draw_by_mer* shows the rates of fluxes in the brackets in the rectangles according to steady state, but in case of visualization of partial network out of all reactions are seen which leads to imbalance for some metabolites. For example, in the Fig. 14 the metabolite *h3c* is produced in 3 reactions (*Cotact77*, *glyc77*, *eth*, *gdp* and *glyc7*) with total $5^{\circ}480,000 \text{ mmol/g-17h-1}$ but it is consumed in 2 reactions (*pyr_decard* *AlcoholH*) with only $2^{\circ}80 = 160 \text{ mmol/g-17h-1}$.

III. optional argument *direction*

It is a string-type variable that can take value of 'isub', 'sub', 'prod' or 'both' (default is 'isub') indicating a direction for the algorithm of the command `draw_by_met`. In case of 'isub' (structure) the algorithm visualizes all metabolites connected to the specified reactions in the argument `rxns`. The key feature of this function is visualization of all specified reactions not taking in account a steady state fluxes in that way representing the structure of the COBRA model. In case of 'sub' (substrate) the algorithm visualizes only those metabolites which are substrates for the specified reactions in the argument `rxns`. This time the algorithm is using a stoichiometric matrix and the steady state fluxes to determine direction of each reaction. The algorithm is using an assumption that only those fluxes which ones are smaller than -10^{-9} mmol/g-1h or greater than 10^{-9} mmol/g-1h are non-zero fluxes. In case of 'prod' (products) the algorithm visualizes only those metabolites which are products for the specified reactions in the argument `rxns` but in case of 'both' the algorithm visualizes both - substrates and products - for the specified reactions in the argument `rxns`. For both cases the algorithm is using the same rules regarding to calculation of the directions for each reaction as for case of 'sub'.

III. optional argument `excludedMets`

This argument represents a list of metabolites (default is empty) that will be excluded from the visualization map of the COBRA model in form of the abbreviations of the metabolites separated by a comma.

III. optional argument `flux`

It is a double-type float size-vector of fluxes of reactions where N is number of reactions (default is vector of 1). This vector is calculated during the optimization of the objective function and can be accessed through the result of the optimization command by `PMResOpt.res`.

Output of the command `draw_by_met`

The command `draw_by_met` has three output vectors in the result: `involvedRxns`, `involvedMets`, and `deadEnds`.

I. vector `involvedRxns`

It is a cell-type vector that contains a list of the involved reactions according to the set of input arguments (see Fig. 18).

	1	2
1	Alcohol	
2	Cefepime	
3	cef	
4	glycol	
5	glycol	
6	glycol	
7	pyruvate	
8	pyruvate	
9	pyruvate	
10	pyruvate	

Fig. 18. An example of the list of the involved reactions.

II. vector `involvedMets`

It is a cell-type vector that contains a list of the involved metabolites in the specified reactions (see Fig. 19).

III. vector `deadEnds`

It is also a cell-type vector but it contains a list of the dead-end metabolites in the specified reactions (see Fig. 17).

TROUBLESHOOTING

Problem: output vectors are valid, but visualization layout is not generated.

Possible reason: the input argument `drawMap` is not provided properly.

Solution: It is a logical-type variable that can take value of 'true' or 'false' (default is 'false') indicating whether to visualize the COBRA model or not. Please pay attention to single quotes around the argument.

TIMING

Timing is given as `Start_time`, `End_time` and `Total_time` for every `PaintNet` function call. It may vary based on equipment computing power.

ANTICIPATED RESULTS

1. Load a model.

```
model = xls2model('test_model.xls')
```

Warning: & value of class "java.math.BigDecimal" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0.0, but in a future release, it will be an error.

Warning: & value of class "java.math.BigDecimal" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0.0, but in a future release, it will be an error.

Warning: & value of class "java.math.BigDecimal" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0.0, but in a future release, it will be an error.

Warning: & value of class "java.math.BigDecimal" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0.0, but in a future release, it will be an error.

Warning: & value of class "java.math.BigDecimal" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0.0, but in a future release, it will be an error.

Warning: & value of class "java.math.BigDecimal" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0.0, but in a future release, it will be an error.

indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "non.mathworks.mde.index.XCodedFrom" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "non.mathworks.mde.index.XCodedFrom" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "non.mathworks.mde.index.XCodedFrom" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "non.mathworks.mde.index.XCodedFrom" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "non.mathworks.mde.index.XCodedFrom" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "non.mathworks.mde.index.XCodedFrom" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
gfp02 ffp[s] ~> gfp[s]
gfp02 atp[s] + gfu-0[s] ~> gfp[s] + atp[s] + h[s]
ca0612 atp[s] + fru[s] ~> ffp[s] + atp[s] + h[s]
ca2 gfp[s] + nady[s] ~> h[s] + gfp[s] + nady[s]
gfp0 gfp[s] + nad[s] ~> h[s] + gfp[s] + nad[s]
ca2 gfp[s] + h2a[s] ~> h[s] + pgf[s]
ca2 pgf[s] ~> h2a[s] + dpp[s]
R03 nady[s] + pgf[s] ~> nady[s] + ca2[s] + rufp-0[s]
ppp4 rufp-0[s] + atp[s] ~> ffp[s] + gfp[s]
ppp3 rufp-0[s] + ffp[s] ~> gfp[s] + sfp[s]
ca062 rufp-0[s] ~> rufp[s]
R03 rufp-0[s] ~> ffp[s]
ppp0 dpp[s] ~> gfp[s] + ppp[s]
gfp03 gfp[s] ~> dthap[s]
gfp01 nad[s] + gfp[s] + pi[s] ~> h[s] + nad[s] + 13dpp[s]
gfp04 atp[s] + 3pg[s] ~> atp[s] + 13dpp[s]
gfp07 2pg[s] ~> 2pg[s]
gfp03 2pg[s] ~> h2a[s] + ppp[s]
gfp04 atp[s] + ppp[s] ~> atp[s] + h[s] + ppp[s]
ppp_0m h[s] + ppp[s] ~> ca2[s] + aa04[s]
ca0 nad[s] + ppp[s] + caa[s] ~> nad[s] + ca2[s] + aa0aa[s]
15ca0a12 nad[s] + fru[s] ~> h[s] + nad[s] + aa04[s]
aa0_0m nad[s] + h2a[s] + aa04[s] ~> 2 h[s] + nad[s] + aa[s]
aa0_0m nad[s] + h2a[s] + aa04[s] ~> 2 h[s] + nad[s] + aa[s]
ca0a1 nad[s] + 1a-0[s] ~> h[s] + nad[s] + ppp[s]
R26 1a-0[s] + q[s] ~> ppp[s] + q02[s]
R27 h[s] + nad[s] + q[s] ~> nad[s] + q02[s]
R28 h[s] + nad[s] + q[s] ~> nady[s] + q02[s]
R29 2 q02[s] + ca2[s] ~> 2 h2a[s] + 2 q[s]
R03 q02[s] + fru[s] ~> q[s] + fru[s]
1ca12 h2a[s] + aa0aa[s] + aa[s] ~> h[s] + aa[s] + ca[s]
1ca7 ca[s] ~> aa[s] + nad[s]
1ca12 ca[s] ~> h2a[s] + ca-aa0aa[s]
1ca12 h2a[s] + ca-aa0aa[s] ~> ca[s] + 1a-0[s]
1ca1 nady[s] + ca[s] ~> nad[s] + ca2[s] + atp[s]
R03 h[s] + nad[s] + atp[s] + ad[s] ~> nady[s] + h2a[s] + gfu-4[s]
1ca10 na0-4[s] ~> h2a[s] + fru[s]
gfp08 nad[s] + na0-4[s] ~> nad[s] + ca2[s] + ppp[s]
a0na06 pi[s] + aa[s] ~> ppp[s] + fru[s]
gfp1 nady[s] + gfp03p[s] ~> h[s] + nad[s] + dthap[s]
R03 h2a[s] + 2pg[s] ~> pi[s] + gfp-0[s]
Ca0a117 atp[s] + gfp-0[s] ~> atp[s] + h[s] + 2pg[s]
R02 atp[s] + gfp[s] ~> atp[s] + h[s] + gfp03p[s]
R03 h2a[s] + gfp03p[s] ~> pi[s] + gfp[s]
R04 h2a[s] + dthap[s] ~> pi[s] + dthap[s]
ppp0a12 xfp-0[s] ~> xfp-0[s]
ppp0a12 atp[s] + xfp-0[s] ~> atp[s] + h[s] + xfp-0[s]
ppp0 gfu-0[s] + sfp[s] ~> ffp[s] + atp[s]
R03 atp[s] + h2a[s] ~> atp[s] + h[s] + pi[s]
R03 h[s] + ppp[s] + aa04[s] ~> ca2[s] + aa[s]
ppp0 h2a[s] + gfu[s] ~> h[s] + gfu[s]
ca0a07 nady[s] + gfu[s] ~> h[s] + nad[s] + 2dpp[s]
R02 gfu-0[s] + fru[s] ~> gfu[s] + h2a[s]
R03 gfu-0[s] + q[s] ~> q02[s] + gfu[s]
ppp2 atp[s] + gfu[s] ~> atp[s] + h[s] + pgf[s]
R03 gfu[s] ~>
R03 gfu-0[s] ~>
R07 fru[s] ~>
R08 xfp-0[s] ~>
R09 gfu[s] ~>
R08 aa0a1 ~>
```

```

R62 gfypr-R[x] ~>
R63 2ahg(x) ~>
R64 gfyw-L[x] ~>
R65 ohd[x] ~>
R66 a2[x] ~>
R68 atah[x] ~>
R67 aa2[x] ~>
R68 lau-Q[x] ~>
R69 aa[x] ~>
R70 acald[x] ~>
R71 acia[x] ~>
R72 dha[x] ~>
R73 h2a[x] = aa2[x] ~> h[x] + heat[x]
R74 ahb-Q[x] ~>

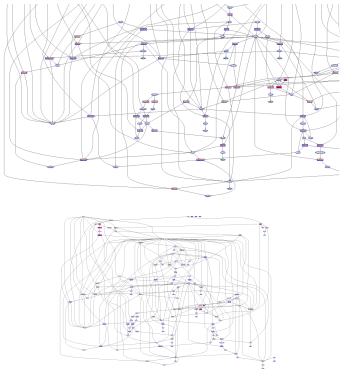
model =
    rxns: (75x1 cell)
    ls: (68x75 double)
    lks: (75x1 double)
    ubi: (75x1 double)
    ci: (75x1 double)
    netci: (68x1 cell)
    ls: (68x1 double)
    rules: (75x1 cell)
    genes: (6x1 cell)
    enzymes: -1
    enzymes: (68x1 char)
    rxnCoeffs: (75x6 double)
    rxnNames: (75x1 cell)
    subSystems: (75x1 cell)
    netNames: (68x1 cell)
    grRules: (75x1 cell)
    confidenceScores: (75x1 double)
    randChans: (75x1 cell)
    rxnChans: (75x1 cell)
    rxnReactions: (75x1 cell)
    names:
        'a'
    compNames: ('Cytosol or Cytosol, Extra-organism or Cytosol, Extra-organism, Periplasm')
    netFormulas: (68x1 cell)
    netCharges: (68x1 double)
    netIons: (68x1 cell)
    netO22: (68x1 cell)
    netCo22: (68x1 cell)
    netPubChans: (68x1 cell)
    netO22: (68x1 cell)
    descriptions: 'level_model.v1s'

```

2. Find involved and dead-end metabolites in the whole model without visualization and without FBA data (assuming all reaction rates are 0).

The model must be loaded before (see step 1). The first two arguments are used for the function `[involvedMet, deadEnd] = draw_by_rxn(model, rxns, drawMap, direction, initialMet, excludeMet, flux)`, the rest will take default values. The expected involved metabolites are all 68 metabolites in the model and the list of them will be created in the MATLAB workspace as variable `involvedMet`. The expected dead-end metabolites are `gly-C[0]`, `lcy-C[0]` and `lyt-C[0]`. The list of dead-end metabolites will be created in the MATLAB workspace as variable `deadEnd`.

```
[involvedMet, deadEnd] = draw_by_rxn(model, model.rxns)
```

6. Visualize the reactions of interest without FBA data (assuming all reaction rates are 1).

The model must be loaded before (see step 1). The list of reactions of interest must be created before (see step 2). The first three arguments are used for the function `[involvedMetts, deadEnds] = draw_by_rxn(model, rxns, drawType, direction, initialMet, excludeMetts, flux)`, the rest will take default values. Besides the list of involved metabolites and the list of dead-end metabolites the hypograph layout will be generated by PaintNet using the Bioinformatics Toolbox. The reaction nodes will contain a for flux rates. All interconnecting edges will be in gray because of no FBA data.

```
[involvedMetts, deadEnds] = draw_by_rxn(model, rxns, "true")
```

Warning: & value of class "van.networks.mdx.mdxin.XCellNetSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdx.mdxin.XCellNetSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdx.mdxin.XCellNetSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdx.mdxin.XCellNetSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0, but in a future release, it will be an error.

Warning: & value of class "van.networks.mdx.mdxin.XCellNetSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0, but in a future release, it will be an error.

specimen with no submargin specimens. Currently one result on this specimen is the indexed value 514217, but in a future release, it will be an error.

Warning: A value of class "com.mathworks.mde.toolbox.XIndexedData" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "xml.math.mml.mde.mmlm.XMLElement" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "non.mathworks.mde.mdeuin.XMdeuiView" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: A value of class "xml.xpath.result.xml.xpath.XMLElement" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warnings: A value of class "van.methworks.mde.unchain.XEdmObliv" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Start_time = 2017_7_14_12_39_16.000

Starting parallel pool (parpool) using the 'local' profile Biograph objects with 38 nodes and 75 edges.

End Date: 2012-7-14 12:09:29 EDT

Total time = 0 0 0 0 0 2,000

James L. Campbell is an

```

"Flip(x)"
"Flip(y)"
"Flip(z)"
"g[x-G](x)"
"andp(x)"
"b(x)"
"Form(x)"
"nandp(x)"
"nandp(y)"
"nandp(z)"
"b2a(x)"
"pg1(x)"
"dlap(x)"
"nand2(x)"
"rFlip-G(x)"
"nandp-G(x)"
"nandp(x)"
"g3p(x)"
"r3p(x)"
"n3p(x)"
"gyr(x)"
"dlap(x)"
"p1(x)"
"13lap(x)"
"2pg(x)"
"2pg(x)"
"pgp(x)"
"p1a(x)"

```

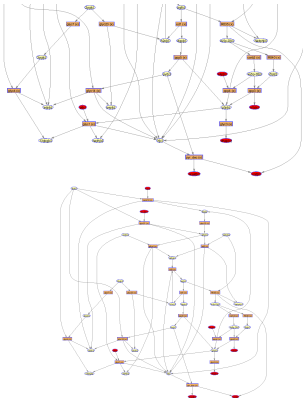
© 2004 Blackwell

```

'g'(x)=0[x]
'f'(x)=
'x2(x)=
'x4p(x)=
'x7p(x)=
'allp(x)=
'p4(x)=
'p4(x)=

```





7. Perform FBA.

The FBA results will be stored in the MATLAB workspace as variable `FBAresults`.

```
FBAresults = solveOptTSS2reCMode(1)(mode 1)
```

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
FBAdataflow =
  flux1 [75x1 double]
  wk1 0
  rxn1 [75x1 double]
  dw1 [50x1 double]
  w1 [75x1 double]
  algorithm 'default'
  state 1
  ninitials 5
  times 0.0001
  basis {}
  w1 [75x1 double]
  r1 0
  y1 [50x1 double]
  w1 [75x1 double]
  w1 [75x1 double]
```

8. Visualize the model with FBA data.

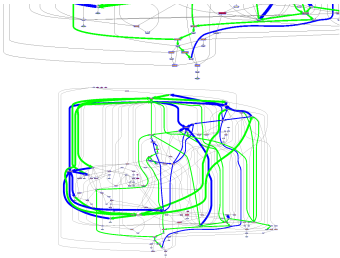
The model must be loaded before (see step 1). The FBA must be performed before (see step 7). Besides the list of FB involved metabolites and the list of these (not 4 like in the step 7 because of FBA data) dead-end metabolites the hypergraph layout will be generated by Paintlet using the Bioinformatics Toolbox. The reaction nodes will contain flux rates according to FBA data. Interconnecting nodes will be in corresponding colors according to flux rates for each reaction.

```
[Solved_flux, dead_end] = draw_by_flux(model, model.rxn, 'true', 'struc', {}, {'-'}, FBAdataflow.w)
```

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCedimFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

```
Start_time = 2017_7_14_12_38_317
Starting parallel pool (parpool) using the 'local' profile ... Biograph object with 135 nodes and 254 edges.
End_time = 2017_7_14_12_39_007
Total_time = 0_0_0_0_0_0_77
Solved_flux =
  'flp1(x)'
  'gfp1(x)'
  'atp1(x)'
  'g1c-0(x)'
  'adp1(x)'
  'h1(x)'
  'fou1(x)'
  'nadh1(x)'
  'g1fp1(x)'
  'g1fp1(x)'
  'nadh1(x)'
  'nad1(x)'
  'nadh1(x)'
  'h2a1(x)'
  'ac1(x)'
  'ac1(x)'
```

5. Visualize the reactions of interest with FBA data.

The model must be loaded before (see step 1). The list of reactions of interest must be created before (see step 2). The FBA must be performed before (see step 3). Besides the list of involved metabolites and the list of dead-end metabolites the hypergraph layout will be generated by PaintNet using the Bioinformatics ToolBox. The reaction nodes will contain flux rates according to FBA data. Interconnecting nodes will be in corresponding colors according to flux rates for each reaction.

```
[Involved_mets, Dead_mets] = draw_by_rxn(model, rxns, 'true', 'false', {''}, {''}, PMASolution.a)
```

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "ros.networks.mdr.mdrin.XCellModel" was indexed with no subscripts specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

for an error.

Start_time = 2017_7_10_12_38_10.428

Starting parallel post (parpost) using the 'local' profile Ringraph object with 58 nodes and 75 edges.

End_time = 2017_7_10_12_38_38.826

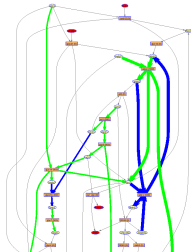
Total_time = 0_0_0_0_0_2.408

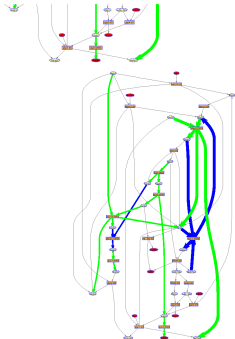
Isolated_nodes =

```
'fhp(x)'  
'gls(x)'  
'atp(x)'  
'gls-Q(x)'  
'adp(x)'  
'a(x)'  
'fou(x)'  
'sou(x)'  
'gldp(x)'  
'soudph(x)'  
'sout(x)'  
'soudh(x)'  
'kds(x)'  
'pgl(x)'  
'dps(x)'  
'cnd(x)'  
'vdpr-Q(x)'  
'vdpr-Q(x)'  
'vdp(x)'  
'gls(x)'  
'vlp(x)'  
'vlp(x)'  
'vlp(x)'  
'gpr(x)'  
'dhp(x)'  
'gpi(x)'  
'lchmg(x)'  
'dpp(x)'  
'dpp(x)'  
'gmp(x)'  
'sacat(x)'
```

Dead_nodes =

```
'gls-Q(x)'  
'fou(x)'  
'cnd(x)'  
'vdp(x)'  
'vlp(x)'  
'dhp(x)'  
'gpi(x)'  
'sacat(x)'
```





10. Create a list of the excludable metabolites.

The list with `apc`, `aad`, `adp` and `adp` has excludable metabolites will be created in the MATLAB workspace as variable `excludMet`.

```
excludMet = {'apc', 'aad', 'adp', 'adp'}
```

[illegible]

11. Visualize the model with excluded metabolites without FBA data.

The model must be loaded before (see step 1). The list of excludable metabolites must be created before (see step 10). The first six arguments are used for the function `[involveMetabs, deactivate] = draw_by_node(model, node, drawing, direction, initiator, excludables, flux)`; the last will take default value. Besides the list of involved metabolites, the list of dead-end metabolites the hightygraph layout will be generated by `ParNet` using the `libNetworks` toolbox. The reaction nodes will contain "r" for fluxes. The generated layout will not contain any of metabolites declared in the list of excludable metabolites `excludables`. All interconnecting edges will be in gray because of no FBA data.

```
[Dissolved_solid, Dead_solid] = draw_by_run(model, model.run, 'True', 'True', {''}, excludedData)
```

```
Warning: <value> of class "com.mathworks.mde.toolbox.X2mdeMdlSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.
```

```
Warning: <value> of class "com.mathworks.mde.toolbox.X2mdeMdlSim" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.
```

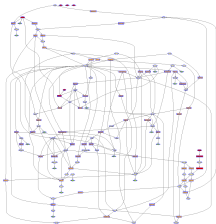
```
Start_time = 2017_7_14_12_38_37.560
```

```
Starting parallel pool (parpool) using the 'local' profile .... Diagram
```

```
End_time = 2017_7_14_12_39_41.991
```

```
Total_time = 0_0_0_0_0_4.287
```

```
Included_mats =  
    'fig(x)'  
    'gfig(x)'  
    'gfig-Q(x)'  
    'fvec(x)'  
    'nmap(x)'  
    'grate(x)'  
    'nmapb(x)'  
    'nadb(x)'  
    'k2a(x)'  
    'pgl(x)'  
    'dpp(x)'  
    'cst(x)'  
    'rule-Q(x)'  
    'rule-Q(x)'  
    'ndp(x)'  
    'gls(x)'  
    'rVec(x)'  
    'cMap(x)'  
    'ypr(x)'  
    'dmap(x)'  
    'pi(x)'  
    'tMap(x)'  
    'Dpp(x)'  
    'Dpp(x)'  
    'jpp(x)
```

12. Visualize the model with excluded metabolites with FBA data.

The model must be loaded before (see step 1). The FBA must be performed before (see step 7). The list of excludable metabolites must be created before (see step 10). Besides the list of involved metabolites and the list of dead-end metabolites the hypergraph layout will be generated by PaintNet using the Bioinformatics Toolbox. The reaction nodes will contain flux rates according to FBA data. Interconnecting nodes will be in corresponding colors according to flux rates for each reaction. The generated layout will not contain any of metabolites declared in the list of excludable metabolites `excludables`.

```
[Solved_verts, Dead_ends] = draw_by_rxn(nodes, model_rxn, 'true', 'struc', {'-'}, excludables, visualization)
```

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

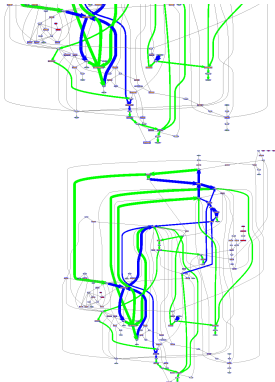
Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Warnings & value of class "icon.networks.mdx.mdxin.XCedInflm" was indexed with no subscript specified. Currently the result of this operation is the indexed value 0x001f, but in a future release, it will be an error.

Start_time = 2017_7_14_12_38_42-760

Starting parallel pool (parpool) using the 'local' profile ... Biograph object with 131 nodes and 262 edges.

End_time = 2017_7_14_12_38_46-847



13. Find involved reactions, involved metabolites and dead-end metabolite in the radius of 2 reactions from the metabolite of interest (in this case etoh[4]) without visualization, without FBA data.

[illegible]

```
[directions, involvedetc, deaths] = draw_by_attribute(['econ(c)'], 'false', 2)
```

Warnings & value of class "com.sun.xml.xpath.XPathImpl" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an array.

Warning: A value of class "com.mathworks.mde.editor.XEditorView" was indexed with no subscript specified. Currently the result of this

operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Warning: & value of class "rnn.networks.mdn.mdnin.XCodedDfAm" was indexed with no subscript specified. Currently the result of this operation is the indexed value itself, but in a future release, it will be an error.

Start_time = 2017_7_14_12_08_337

Starting parallel post (parpost) using the 'local' profile

End_time = 2017_7_14_12_08_044

Total_time = 0_0_0_0_0_047

directionNames =

```
'XlocMid1'
'000'
'g1p121'
'car12'
'm2'
'g1p4'
'm3'
'g1p12'
'g1p14'
'gpr_dmc'
'm3'
'acel_delyb'
'acel_delyb'
'car14'
'027'
'028'
'loc12'
'030'
'g1p18'
'gpr13'
'Car14127'
'042'
'pmpg1413'
'040'
'049'
'gpr18'
'acel127'
'gpr12'
'039'
'033'
```

locationDfAm =

```
'Filp(x)'
'g1p(x)'
'g1p(x)'
'atp(x)'
'g1p-Q(x)'
'ndp(x)'
'0(x)'
'Fru(x)'
'ndp(x)'
'g1p(x)'
'ndp(x)'
'ndp(x)'
'ndp(x)'
'ndp(x)'
'ndp(x)'
'ndp(x)'
'ndp(x)'
'ndp-Q(x)'
'g1p(x)'
'gpr(x)'
'dhsp(x)'
'pi(x)'
'134p(x)'
'gpr(x)'
'gpr(x)'
```

```

'acutal[x]'
'acu[x]'
'acua[x]'
'atalk[x]'
'au[x]'
'tau-0[x]'
'g[x]'
'g2[x]'
'max[x]'
'ail[x]'
'ahg[x]'
'ahd[x]'
'glu-L[x]'
'mat-L[x]'
'houd[x]'
'glyco-0[x]'
'glyco-0[x]'
'glyco-0[x]'
'glyco[x]'
'xyl-0[x]'
'min[x]'
'gla[x]'
'glu[x]'
'Dahgla[x]'

```

```

deadEnds =
'fap[x]'
'glu-0[x]'
'fru[x]'
'mutp-0[x]'
'gfp[x]'
'ahap[x]'
'Dahg[x]'
'hyp[x]'
'gms[x]'
'tau-0[x]'
'g[x]'
'g2[x]'
'max[x]'
'ail[x]'
'ahg[x]'
'ahd[x]'
'glu-L[x]'
'mat-L[x]'
'houd[x]'
'glyco-0[x]'
'glyco[x]'
'xyl-0[x]'
'min[x]'
'gla[x]'
'Dahgla[x]'

```

15. Visualize the part of the model in the radius of 2 reactions from the metabolite of interest (in this case `etoh[c]`) without FBA data.

The model must be loaded before (see step 1). The first four arguments are used for the function `(directionFluxes, involvedMet, deadEnds) = draw_by_net(model, metName, deadMap, radius, direction, excludeMet, flux)`, the rest will take default values. Besides the list of involved reactions, the list of involved metabolites and the list of dead-end metabolites the hypograph layout will be generated by PaintNet using the Bioinformatics Toolbox. The reaction nodes will contain a for flux rates. All interconnecting edges will be in gray because of no FBA data.

```
[directionFluxes, involvedMet, deadEnds] = draw_by_net(model, {'etoh[c]'}, 'true', 2)
```

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was indexed with no subscript specified. Currently the result of this operation is the indexed value `iseIf`, but in a future release, it will be an error.

Warning: & value of class "non.networks.mdr.mdrain.XCodedItem" was

indexed with no subscript specified. Currently the result of this operation is the indexed value `size17`, but in a future release, it will be an error.

Warning: & value of class "non.methodski.msh.mshwin.XCmshMshWin" was indexed with no subscript specified. Currently the result of this operation is the indexed value `size17`, but in a future release, it will be an error.

Warning: & value of class "non.methodski.msh.mshwin.XCmshMshWin" was indexed with no subscript specified. Currently the result of this operation is the indexed value `size17`, but in a future release, it will be an error.

Warning: & value of class "non.methodski.msh.mshwin.XCmshMshWin" was indexed with no subscript specified. Currently the result of this operation is the indexed value `size17`, but in a future release, it will be an error.

Start_time = 2017_7_14_12_30_06.993

Starting parallel post (parpost) using the "local" profile --- Biograph object with 78 nodes and 140 edges.

End_time = 2017_7_14_12_30_08.82

Total_time = 0_0_0_0_0_3.827

directionalities =

```
"Bisecting"
"glpc21"
"carb12"
"ed2"
"glpc4"
"ed3"
"glpc1"
"glpc14"
"pyr_dms"
"ed5"
"acet_dshy"
"acet_dshy2"
"carb15"
"R27"
"R28"
"loc12"
"R31"
"glpc8"
"glpc13"
"CoFac1127"
"R42"
"peng1ac3"
"R48"
"R49"
"pyr6"
"acetac12"
"pyr2"
"R56"
"R57"
```

linearizedMeth =

```
"Flp(x)"
"glpc(x)"
"atp(x)"
"glc-0(x)"
"adp(x)"
"b(x)"
"fruc(x)"
"acdh(x)"
"glhsp(x)"
"acdhph(x)"
"acat(x)"
"acdh(x)"
"R2a(x)"
"pg1(x)"
"ac2(x)"
"aclyp-0(x)"
"glpc(x)"
"pyr(x)"
"dhsp(x)"
"pi(x)"
"LDhsp(x)"
"dhsp(x)"
"pgp(x)"
"acacat(x)"
"acac(x)"
"acetac(x)"
"acdh(x)"
"ac(x)"
"lac-0(x)"
"ag(x)"
"ph2(x)"
"acac(x)"
"acil(x)"
"adp(x)"
"acil(x)"
"glpc-4(x)"
"acil-4(x)"
"acac(x)"
"glpcac(x)"
"glpc-0(x)"
"glpc(x)"
"acacacil(x)"
```

```

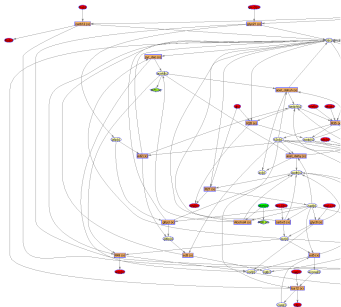
    'sgn(x)'
    'asin(x)'
    'g1a(x)'
    'g1m(x)'
    '2dmg1m(x)'

```

```

deadEnds =
    'fhp(x)'
    'g1a-0(x)'
    'fms(x)'
    'outp-0(x)'
    'g1a(x)'
    'dhp(x)'
    '23dmg(x)'
    '2mg(x)'
    'pmp(x)'
    'cas-0(x)'
    'g(x)'
    'g12(x)'
    'cas(x)'
    'c11(x)'
    'dmg(x)'
    'hd(x)'
    'g1a-4(x)'
    'm1-4(x)'
    'hmd(x)'
    'g1p-0(x)'
    'g1p(x)'
    'outp-0(x)'
    'cas(x)'
    'g1a(x)'
    '2dmg1m(x)'

```





15. Visualize the part of the model in the radius of 2 reactions from the metabolite of interest (in this case *etohc*) with FBA data.

The model must be loaded before (see step 1). The FBA must be performed before (see step 7). Besides the list of involved reactions, the list of involved metabolites and the list of dead-end metabolites the typegraph layout will be generated by PaintMet using the Bioinformatics ToolBox. The reaction nodes will contain flux rates according to FBA data. Interconnecting colors will be in corresponding colors according to flux rates for each reaction.

```
[directedEdges, involvedMetIs, deadEnds] = draw_by_net(model, {'etohc'}, 'true', 2, 'etohc', {''}, PRASolution.x)
```

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Warning: & value of class "yon.networks.mdr.mdr.XCedndfAm" was indred with no subscript specified. Currently the result of this operation is the indred value 0x0, but in a future release, it will be an error.

Start_time = 2017_7_14_12_38_38.382

Starting parallel pool (parpool) using the 'local' profile --- Biograph object with 78 nodes and 100 edges.

End_time = 2017_7_14_12_38_31.283

Total_time = 8_8_8_8_8_8.888

directedEdges =

'etohc'

'888'

'gtp21'

'carb12'

'm2'

'gtpd'

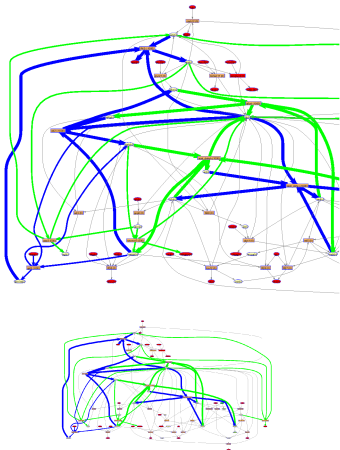
'm3'

'gtp1'


```

'met(s)'
'glt(s)'
'2dhglt(s)'

```



18. Visualize substrates in the radius of 2 reactions from the metabolite of interest (in this case *ent(s)*) with FBA data.

The model must be loaded before (see step 1). The FBA must be performed before (see step 7). Besides the list of involved reactions, the list of involved metabolites and the list of dead-end metabolites the hypergraph layout will be generated by PaintFlttr using the Bioinformatics Toolbox where only substrates for *glt(s)* in the radius of 2 reactions will be visualized. The reaction nodes will contain flux rates according to FBA data. Interconnecting nodes will be in corresponding colors according to flux rates for each reaction.

```
[directionFrom, involvedNode, deadEnd] = draw_by_net(model, {'gtp[x]'}, 'True', 2, 'col', {''}, PPMaturation.x)
```

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Warnings & value of class "rnn.networks.mdn.mdnin.XCnnModel" was indexed with no subscript specified. Currently the result of this operation is the indexed value `listIf`, but in a future release, it will be an error.

Start_time = 2017_7_14_12_33_367

Starting parallel post (parpost) using the "local" profile Biograph object with 17 nodes and 20 edges.

End_time = 2017_7_14_12_34_488

Total_time = 0_0_0_0_0_822

directionFrom =

```
'ed2'
'gtpd'
'gtpu14'
'acti_delay'
'sensor'
'gppd'
```

involvedNode =

```
'gtp[x]'
'atp[x]'
'h[x]'
'ndp[x]'
'gtp[x]'
'ndp[x]'
'acti[x]'
'h2a[x]'
'gpr[x]'
'acta[x]'
'glsa[x]'
```

deadEnds =

```
'acti[x]'
'h2a[x]'
'gpr[x]'
'acta[x]'
```



indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

`Start_time = 2017_7_14_12_39_34.388`

Starting parallel pool (parpool) using the 'local' profile ... Biograph object with 22 nodes and 28 edges.

`End_time = 2017_7_14_12_39_35.744`

`Total_time = 0_0_0_0_0_0.055`

`directionalFlow =`

```
'g1p14'  
'gpp2'  
'ed2'  
'ed1'  
'g1p23'  
'acut_duty'  
'gpl3'  
'ed2'
```

`localFlowIn =`

```
'g1p14'  
'gpp2'  
'acut_duty'  
'g1p14'  
'acut_duty'  
'acut_duty'  
'gpl3'  
'gpp2'  
'gpp2'  
'gpp2'  
'acut_duty'  
'g1p23'  
'g1p14'
```

`localFlow =`

```
'g1p14'  
'acut_duty'  
'gpp2'  
'gpp2'  
'acut_duty'  
'g1p14'
```

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

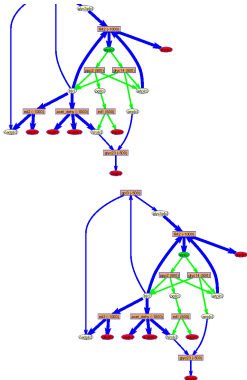
Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.

Warning: & value of class "van.neuhof.mdr.mdrin.XCodedFlow" was indexed with no subscript specified. Currently the result of this operation is the indexed value `sizeIf`, but in a future release, it will be an error.





Acknowledgments

Authors are thankful to Ines Thiele who inspired to make a simple visualization software for COBRA models.

This work is funded by a project of European Structural Fund Nr. 2008/020/11DP1.1.1.2.018/NPVA/VWA/128 "Latvian Interdisciplinary Interuniversity Scientific Group of Systems Biology" www.systbio.lv.

REFERENCES

- Avrich, M.K., Thiele, I., 2012. Contextualization Procedure and Modeling of Monocyte Specific TLR Signaling. *PLoS One* 7, e48976. doi:10.1371/journal.pone.0048976
- Comadori, C.A., Rod'iguez, V., Andrews, B.A., Asenjo, J.A., 2016. Genome-scale reconstruction of *Salinispora tropica* C90-640 metabolism to study strain-specific adaptation. *Antonie Van Leeuwenhoek* 108, 1079–1090. doi:10.1007/s10482-016-0361-9
- Demchenko, A., Alberdi, I.R., Altemann, M., Allen, S.E., Kaljuzhnyaya, M.G., 2017. Fatty Acid Biosynthesis Pathways in *Methylobaculum buysseae* S0(R1). *Front. Microbiol.* 7. doi:10.3389/fmicb.2016.02167
- Koussa, J., Chalabandeh, A., Salehi-Astiani, K., 2014. Computational Approaches for Microbial Bioutil Optimization: A Review. *Biomol. Res. Int.* 2014, 1–12. doi:10.1155/2014/649493
- Mao, L., Vervaeke, W.S., 2014. COBRA: a COBRA toolbox extension for model-driven discovery and analysis. *Bioinformatics* 30, 584–585. doi:10.1093/bioinformatics/btt723
- Mettavalli, J., Stalder, E., 2013. Analysis of synthetic metabolic pathways solution space, in: 2013 International Conference on System Science and Engineering

Pallaś, B.D., 2008. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, New York.

Pentjuŝ, A., Ōdina, I., Kozlomiņš, A., Fell, D.A., Stalidzans, E., Kalneriņš, U., 2013. Biotechnological potential of respiring *Zymomonas mobilis*: a stoichiometric analysis of its central metabolism. *J. Biotechnol.* 183, 1–12. doi:10.1016/j.biotech.2013.02.018

Pentjuŝ, A., Stalidzans, E., Ļepins, J., Kokina, A., Marjoniņa, J., Zīenariņš, P., Moza, I., Scheldtaka, R., Hartman, H., Podman, M.D., Fell, D.A., Vīgants, A., 2017. Model-based biotechnological potential analysis of *Kluyveromyces marxianus* central metabolism. *J. Ind. Microbiol. Biotechnol.* doi:10.1007/s10296-017-1946-6

Richards, M.A., Liu, T.J., Zhang, J., Ragotzke, S.W., Leigh, J.A., Price, N.D., 2016. Exploring Hydrogenotrophic Methanogenesis: a Genome Scale Metabolic Reconstruction of *Methanosaeta marisnigrae*. *J. Bacteriol.* 198, 3379–3390. doi:10.1128/JB.00671-16

Rovs, Z., Mednis, M., Ōdina, I., 2013. Biochemical networks comparison tool, in: 38th International Scientific Conference on Applied Information and Communication Technologies. Jelgava, Latvia, pp. 309–311.

Rubina, T., Stalidzans, E., 2013. GENESA — A software tool for evolution modeling of biochemical networks' structure, in: 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI). IEEE, pp. 385–390. doi:10.1109/CINTI.2013.6706218

Thiele, I., Palsson, B.O., 2010. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.* 5, 93–121. doi:10.1038/nprot.2009.203