
```

initCobraToolbox(false) % false, as we don't want to update
changeCobraSolver ('gurobi', 'all');
global CBTDIR
modelFileName = 'Recon2.0model.mat';
modelDirectory = getDistributedModelFolder(modelFileName); %Look up
    the folder for the distributed Models.
modelFileName= [modelDirectory filesep modelFileName]; % Get the full
    path. Necessary to be sure, that the right model is loaded
model = readCbModel(modelFileName);

[selExc, selUpt] = findExcRxns(model);
uptakes = model.rxns(selUpt);
subuptakeModel = extractSubNetwork(model, uptakes);
hiCarbonRxns = findCarbonRxns(subuptakeModel,1);
modelalter = changeRxnBounds(model, hiCarbonRxns, 0, 'b');
energySources = {'EX_adp'; 'EX_amp(e)'; 'EX_atp(e)'; 'EX_co2(e)';...
    'EX_coa(e)'; 'EX_fad(e)'; 'EX_fe2(e)'; 'EX_fe3(e)'; 'EX_gdp(e)';...
    'EX_gmp(e)'; 'EX_gtp(e)'; 'EX_h(e)'; 'EX_h2o(e)'; 'EX_h2o2(e)';...
    'EX_nad(e)'; 'EX_nadp(e)'; 'EX_no(e)'; 'EX_no2(e)'; 'EX_o2s(e)'};
modelalter = changeRxnBounds (modelalter, energySources, 0, 'l');
% modelfva1 represents aerobic condition
modelfva1 = modelalter;
modelfva1 = changeRxnBounds(modelfva1, 'EX_glc(e)', -20, 'l');
modelfva1 = changeRxnBounds(modelfva1, 'EX_o2(e)', -1000, 'l');
% modelfva2 represents anaerobic condition
modelfva2 = modelalter;
modelfva2 = changeRxnBounds(modelfva2, 'EX_glc(e)', -20, 'l');
modelfva2 = changeRxnBounds(modelfva2, 'EX_o2(e)', 0, 'l');
% [minFlux, maxFlux, Vmin, Vmax] = fluxVariability(model,...
% optPercentage,osenseStr, rxnNameList, verbFlag, allowLoops, method);
% Selecting several reactions of the model that we want to analyse
    with FVA
rxnsList = {'DM_atp_c_'; 'ACOAHi'; 'ALCD21_D'; 'LALDO'; 'ME2m';...
    'AKGDm'; 'PGI'; 'PGM'; 'r0062'};
% Run FVA analysis for the model with the constraints that simulates
    aerobic conditions:
[minFlux1, maxFlux1, Vmin1, Vmax1] = fluxVariability(modelfva1,
    100, 'max', rxnsList)
% Run FVA analysis for the model with the constraints that
% simulates anaerobic conditions:
[minFlux2, maxFlux2, Vmin2, Vmax2] = fluxVariability(modelfva2, [],
    [], rxnsList)
ymax1 = maxFlux1;
ymin1 = minFlux1;
ymax2 = maxFlux2;
ymin2 = minFlux2;

maxf = table(ymax1, ymax2)
minf = table(ymin1, ymin2)
maxfxs = table2cell(maxf);
minfxs = table2cell(minf);

```

```

figure
plot1 = bar(cell2mat(maxfxs(1:end, :)));
hold on
plot2 = bar(cell2mat(minfxs(1:end, :)));
hold off
xticklabels({'DM_atp_c_', 'ACOAHi', 'ALCD21__D', 'LALDO',...
            'ME2m', 'AKGDm', 'PGI', 'PGM', 'r0062'})
set(gca, 'XTickLabelRotation', -80);
yticks([-1000 -800 -600 -400 -200 0 200 400 600 800 1000])
xlabel('Reactions from the models')
ylabel('Fluxes')
legend({'Aerobic', 'Anaerobic'}, 'Location', 'southwest')
title('Variations in fluxes in the aerobic and anaerobic conditions')
changeCobraSolver('ibm_cplex', 'all', 1);
[minFluxF1, maxFluxF1, optsol, ret, fbasol, fvamin, fvamax,
 statussolmin, statussolmax] = fastFVA(modelfva1);
[minFluxF2, maxFluxF2, optsol2, ret2, fbasol2, fvamin2, fvamax2,...
 statussolmin2, statussolmax2] = fastFVA(modelfva2);
ymaxf1 = maxFluxF1;
yminf1 = minFluxF1;
ymaxf2 = maxFluxF2;
yminf2 = minFluxF2;

maxf = table(ymaxf1, ymaxf2);
minf = table(yminf1, yminf2);

maxf = table2cell(maxf);
minf = table2cell(minf);

figure
plot3 = bar(cell2mat(maxf(1:end, :)));
hold on
plot4 = bar(cell2mat(minf(1:end, :)));
hold off
xticks([0 2000 4000 6000 8000 10600])
yticks([-1000 -800 -600 -400 -200 0 200 400 600 800 1000])
xlabel('All reactions in the model')
ylabel('Fluxes')
legend({'Aerobic', 'Anaerobic'})
title('Variations in fluxes in the aerobic and anaerobic conditions')

```

/ ____ / ____ \ / ____ \ / ____ \ / ____ \ / ____ \ / *Constraint-Based*
Reconstruction and Analysis
 / *The COBRA*
Toolbox - 2021
 / *Documentation:*
 / ____ / ____ \ / ____ \ / ____ \ / ____ \ / ____ \ / *http://*
opencobra.github.io/cobratoolbox

```

> Checking if git is installed ... Done (version: 2.20.1).
> 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 (this may take a while)...
Done.
> Adding all the files of The COBRA Toolbox ... Done.
> Define CB map output... set to svg.
> TranslateSBML is installed and working properly.
> Configuring solver environment variables ...
  - [----] ILOG_CPLEX_PATH: --> set this path manually after
installing the solver ( see https://opencobra.github.io/cobratoolbox/docs/solvers.html )
  - [----] GUROBI_PATH: --> set this path manually after installing
the solver ( see https://opencobra.github.io/cobratoolbox/docs/solvers.html )
  - [----] TOMLAB_PATH: --> set this path manually after installing
the solver ( see https://opencobra.github.io/cobratoolbox/docs/solvers.html )
  - [----] MOSEK_PATH: --> set this path manually after installing
the solver ( see https://opencobra.github.io/cobratoolbox/docs/solvers.html )
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

```

Support	LP	MILP	QP	MIQP	NLP	
<hr/>						
gurobi	active		0	0	0	0
ibm_cplex	active		0	0	0	0
tomlab_cplex	active		0	0	0	0
glpk	active		1	1	-	-
mosek	active		0	-	0	-
matlab	active		1	-	-	1
pdco	active		1	-	1	-
quadMinos	active		1	-	-	-
dqqMinos	active		1	-	1	-
cplex_direct	active		0	0	0	-
cplexlp	active		0	-	-	-
qpng	passive		-	-	1	-
tomlab_snopt	passive		-	-	-	0
lp_solve	legacy		1	-	-	-
<hr/>						
Total	-		6	1	3	0

```

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

```

```
> You can solve LP problems using: 'glpk' - 'pdco'
> You can solve MILP problems using: 'glpk'
> You can solve QP problems using: 'pdco'
> You can solve MIQP problems using:
> You can solve NLP problems using:

> Checking for available updates ... skipped

Error using changeCobraSolver (line 271)
changeCobraSolver: The global variable `GUROBI_PATH` is not set.
Please follow the instructions on https://opencobra.github.io/cobratoolbox/docs/solvers.html in order to set the environment
variables properly.

Error in tutorial_FVA (line 2)
changeCobraSolver('ibm_cplex','LP');
```

Published with MATLAB® R2019b