User Manual

The package is provided as a zip archive FindPath.zip and requires Matlab R2007a or higher with Cobra-Toolbox, EFMTool (http://www.csb.ethz.ch/tools/efmtool). It contains the set of functions to design and rank synthetic pathways for new metabolic conversion process. To illustrate the methodology, the example of xylose as a new substrate for the yeast *Saccharomyces cerevisiae* will be used.

2.1 Installation

There is no constraint on the unzipped folder location, nevertheless the path to the package should be specified either in matlab path definition (GUI) or with the *addpath()* command line. The archive contains 5 folders.

- The data folder contains the host CBModel (SBML) and the SAR database (CSV).
- The documentation folder contains the following files: example.m file,Xylose-DB-Model.xls and a copy findPath.log.
- The FindPath functions folder contains the matlab functions to design and rank the pathways
- The MakeModel folder contains functions to convert a CSV file into a CBModel
- The results folder contains the results while running the example file

2.2 Utilization

2.2.1 Inputs formats

FindPath can use directly Cobra-toolbox CBModel , SBML or CSV as input formats. The Cobra-toolbox CBModel is the matlab model structure used by the Cobra-toolbox. It is composed by several fields like the stoichiometric matrix, the reversibility and more (see the Cobra-toolbox documentation). Using the Cobra-toolbox SBML import function the user can also provide the SBML file containing the constraint based models. Finally The CSV file must be converted before using a set of functions provided in the makeModel tool of the Cobra Toolbox. Generally, the host metabolic constraint based model is provided in Cobra-toolbox CBModel or SBML formats while the substrate-associated reaction (SAR) model is provided in CSV format.

2.2.2 Construction of the SAR database and model

The objective of Substrate-Associated Reaction (SAR) database is to minimize the number of reactions found in available metabolic databases (e.g. MetaCyc, Kegg) to a subset of relevant reactions which are closely related to the compound of interest (CI). SAR database contains all reactions which directly use the CI as substrate or product (i.e. seed reactions) plus all the reactions that use or produce metabolites used by the seed reactions (i.e. seed extended). The seed extended can contains up to the second or the third neighbors. If whished by the user all the metabolic pathways that used reactions from the seed extended can be added.

2.2.2.1 SAR database

If user does not possess a tool to automatically create a SAR database, she/he can apply the following method.

<u>Collect all the reactions:</u>

Using MetaCyc (http://www.metacyc.org/), the name of the CI can be directly entered into the searching bar. A list of all the pathways involved in the metabolic conversion of the CI is given. For xylose, four pathways are given. For each pathways, the list of reactions can be directly stored to an excel sheet. For instance, the pathway “xylose degradation I” contains two reactions. By clicking on the first reaction, we access to a list of enzyme and genes. By clicking on one of the enzyme (e.g.: xylose isomerase), the detail of the reaction is given “ a D-xylopyranose <=> D-xylulose”. The reaction can be selected and stored in an excel sheet by drag and drop or copy/paste. By repeating this step for all the reactions of each pathway, the SAR database can be built.

Standardize all the reactions:

To avoid redundancy or miss prediction between the SAR database and the host metabolic model, the name of the reactions and the name of the metabolites shared between the SAR database and the metabolic host model must have the same ID. If not, we suggest that the user change the name and ID in the SAR database rather than in the host model. The BIGG database (http://bigg.ucsd.edu/bigg/home.pl) or the SBML file of the host metabolic model can be used to translate the Metacyc ID into genome scale metabolic model ID.

For instance the reaction:” a D-xylopyranose <=> D-xylulose” will be translated into : “xyl-D <=> xylu-D”, Reaction ID = R1

When reaction and /or metabolite are not present in the BIGG database or in the metabolic host model, the user has to create new names for metabolite and reaction that are not already used in the host model.

2.2.2.2 SAR model

One main difference between the SAR model and database is the mathematical aspect of the model. The SAR model should respect the metabolic constrain-based model rules:

1. Balanced reactions: the mass balance and stoichiometric of the reaction must be correct
2. Redundant reaction should be removed
3. Generic reaction must be instanced
4. Reversibility of the reaction should be given
5. Input, output metabolites and cofactors should be complemented

Point 1 to 3 should be check and made while the SAR database is created. Point 4 can be done by looking at the symbol of the reaction in Metacyc; the <=> symbol means reversible reactions and -> irreversible reactions. For point 5, input metabolite corresponds to the CI and should be added as follow: ->CI[e] with its transporter: CI[e]<=>cCI[c]. Output metabolites correspond to the host metabolites. For both output metabolites and cofactors (e.g. nad, nadp, atp..), an artificial exchange fluxes and must be added as follow: “metabolite[c] <=> “. Be careful to put space between each metabolites and symbols (ie. +, ->,] <=>).

Below is the illustration of how the excel sheet and must be formatted. This file must be saved in CSV format. In our example: xylose (xyl-D) = input metabolite; xylulose 5 phosphate (xu5p-D) = one of a ouput metabolite; nadp = one of a cofactor. The complete database and model for xylose is provided (Xylose-DB-Model.xls in documentation folder).

RXN ID	RXN NAME	EQUATION	REVERSIBILITY
R1	Reaction 1	xyl-D[c] <=> xylu-D[c]	1
R2	Reaction 2	xylu-D[c] + atp[c] -> xu5p-D[c] + adp[c] + h[c]	0

EX_xyl(e)	xylIn	-> xyl-D[e]	0
Xyltex	Xyl transport via diffusion (extracellular to periplasm)	xyl-D[c] <=> xyl-D[e]	1
EX_xu5p(e)	xu5pOut	xu5p[c] <=>	1
EX_nadp(e)	nadpOut	nadp[c] <=>	1

- RXN ID: The short name of the reactions
- RXN NAME: Name of the reaction
- EQUATION: Equation of the reaction
    - Compartment are specified by adding a tag between brackets: [c] means cytosol and [e] means extracellular.
- REVERSIBILITY: the reversibility of the reactions (1 reversible, 0 irreversible)

Except the fluxes of the input metabolites, all the artificial exchanges fluxes must be put in the ignored reaction list (i.e options.ignoreRxns in the example file). This is done to avoid artificial increase of the size of the newly designed pathways and potential miss prediction of the pathway efficiency if some of those metabolites are not present in the host metabolism.

2.2.3 Example: Xylose consumption in S. cerevisiae

An example file (example.m in documentation folder) is provided in the package and can be taken as a template for further utilization. It is composed of 3 blocks, paths end files definitions, variables initialization and the FindPath run command line. By running this example, it will illustrate the use of FindPath as an unified workflow to predict and select the best pathways involved in the conversion of xylose which is not naturally consumed by *S. cerevisiae*.

- If neither FindPath nor models folders are defined in the matlab path definition, this can be done using the following lines:

```
%FindPath path
FindPathPath='/myPath/FindPath/functions';
addpath(decapathPath);

%ModelPath
modelPath='/myPath/FindPath/data';
addpath(modelPath);

%name of the host CBM
modelName= iMM904_flux.xml';  % genome scale model of  S. cerevisiae from BMC Syst Biol. 2010 Dec 29;4:178. doi: 10.1186/1752-0509-4-178.

% name of the SAR database
dbCSVName='dbModel.csv'; % SAR model containing all the reactions involved in xylose metabolism
```

- The database model in CSV format is converted using the makeModel tool:

```
%CSV to model path
cvs2cbm='/myPath/FindPath/MakeModel';
addpath(cvs2cbm);
dbModel=makeCBModelFromCSV(dbCSVName,';','dbModel');
```

- The model variables are initiated as follow:

```
oriPath=pwd;
cd(modelPath)
 model=readCbModel(modelName);
 dbModel=makeCBModelFromCSV(dbCSVName,';','dbModel');
cd(oriPath);
```

- The option variables are initiated as follow:

```
options=struct();
options.metIn='xyl-D[e]';  % input metabolite
options.metOut={'xu5p-D[c]','glx[c]','akg[c]'};  % output metabolites
options.maxLength=10;    % maximal length of solution pathways
options.maxPwy=15;  % maximal number  of solution  pathways to be exported in SBML
options.resPath='/myPath/FindPath/resultsXyl'; % path to file were results in SBML format will be stored
options.namePwy='test';  % name of the result SBML  files
options.weightEfficiency=1;  % weight for the efficiency score
options.weightLength=1; % weight for the length score
options.ignoreRxns={'EX_atp(e)','EX_adp(e)','EX_xu5p-D(e)','EX_h(e)','EX_h(e)','EX_akg(e)','EX_pyr(e)','EX_oea(e)','EX_rea(e)','EX_glx(e)','EX_nad(e)','EX_nadp(e)','EX_nadph(e)','EX_nadh(e)'};  % write all the artificial exchange reactions that has to be removed for the efficiency score calculation
```

- The calculation is launched using the following command line:

```
[pwys,score]=FindPath(model,dbModel,EFMTools,options);
```

The resulting files will be written in the /myPath/Findpath/results' folder, with this nomenclature test_X.sbml,   where X is the rank of the pathway.

- The following command lines at the beginning and at the end of the example file return the elapsed time of the complete code.

```
t=clock;

Time=etime(clock, t)
```

At the end of the run:

A log file is generated. This file indicates:

- The total number of pathways that have been found from the SAR database
- The total number of pathway that have been kept based on the metabolite composition and the length filters
- The number of solution pathway to be exported in SBML
- The details for each solution pathway: length, efficiency, reaction, equation. Solution

pathways are ranked according to both length and efficiency scores that have been weighted according to the user (see detail section 2.2.4).

A copy of the log file (findPath.log) obtained by running the example can be found in the documentation folder.

The variables pwys contains the list of the different pathways ranked according to both length and efficiency. The score variable contains the score of each pathway ranked in the same order (e.g; score in score[1] is the score for the pathway in pwys{1}).

2.2.4 Pathways ranking

FindPath compute both pathway length and pathway efficiency. To find a good compromise between both scoring criteria, a weighted score function has been implemented. This allows to consider equally both criteria or to give a priority on one of them.

The score function is:

$$score = w_{length}\frac{pathway\ length}{\min(pathways\ Length)} + w_{efficiency} Optimization\ value$$

Where w_{lenght} is the weight for the length and $w_{efficiency}$ the weight of the optimization results. As the optimization value is most of the time below 1, we decided to divide the length of the pathway by the minimal length among all the pathways to obtain a ration value which is in the same order scale as the optimal value. Then findPath sorts the pathways by decreasing score to give the final pathways order. Weights are given in options.weightEfficiency and options.weightLength (see section 2.2.3).

2.2.5 Functions

A set of functions has been implemented for the FindPath package. Additionally MakeModel functions are provided for converting CSV file into CBM model structure from the Cobra-Toolbox formalism and SBML file.

2.2.5.1 FindPath functions

<u>FindPath</u>

[pwys,score]=FindPath(model,dbModel,EFMToolPath,options)

Design pathways from a set of reactions

INPUTS

mode	CBM host for metabolic modifications (sbml file or model object)
dbModel	Path to the EFMTool package
EFMToolPath	Set of reactions to design the new pathway (sbml file or model object)
options	Structure with fields
rxnIn	Reactions starting the metabolic conversion process
rxnMust	Reactions included in the metabolic conversion process
rxnOut	Reactions ending the metabolic conversion process

rxnForb	Reactions to avoid in the metabolic conversion process
metIn	Metabolites starting the metabolic conversion process
metMust	Metabolites included in the metabolic conversion process
metOut	Metabolites ending the metabolic conversion process
metForb	Metabolites to avoid in the metabolic conversion process
maxLength	Maximal number of reactions in the pathways
namePwy	Name for pathways model
maxPwy	Maximal number of solution pathways
ignoreRxns	List of reactions to not take into account
weightEfficiency	Weight of the pathway efficiency in the final ranking
weightLength	Weight of the pathway length in the final ranking

OUTPUTS

pwys	List of pathways
score	Score of the pathways

addPwy

pwyModel=addPwy(model,pwy,dbModel)

Add the pathway in a model

INPUTS

model	CBM host for metabolic modifications (sbml file or model object)
pwy	List of reactions or sbml file
dbModel	CBM model if pwy is a list of reactions

OUTPUT

pwyModel	Model structure with the new pathway

findMetsOfRxn

mets = findMetsOfRxn(model, rxns)

Find metabolites involved in reaction(s)

INPUTS

model	CBModel
rxns	Reaction name (possibly a cell of several reaction names)

OUTPUT

mets	List of metabolites

findPwyFromPwyIds

pwys=findPwyFromPwyIds(mnet,pwyIds)

Extract pathways from the pathway IDs

INPUTS

mnet	Structure results from EFMTools
pwyIds	List of the pathway Ids

OUTPUT

pwys	List of pathways

findPwysLength

[pwys,lengthPwys]=findPwysLength(pwys)

Find the length of pathways

INPUT

pwys	List of pathways

OUTPUTS

pwys	Pathways sorted by length
lengthPwys	List of pathways length

findPwyWithIntersectRxns

pwyIds=findPwyWithIntersectRxns(mnet,rxns)

Find the intersection of pathways having specific reactions

INPUTS

mnet	Structure results from EFMTools
rxns	List of reactions

OUTPUT

pwyIds	List of pathway ID

findPwyWithLength

pwyIds=findPwyWithLength(mnet,maxLength)

Find pathways having a length under or equal to the given value

INPUTS

mnet	Structure results from EFMTools
maxLength	Maximal size of the pathways

OUTPUTS

pwyIds	List of pathways ID

findPwyWithRxns

pwyIds=findPwyWithRxns(mnet,rxns)

Find the intersection of pathways having specific reactions

INPUTS

mnet	Structure results from EFMTools
rxns	List of reactions

OUTPUTS

pwyIds	List of pathway ID

findSubOfRxn

mets = findSubOfRxn(model, rxns)

Find metabolites substrate involved in reaction(s)

INPUTS

model	CBModel
rxns	List of reactions

OUTPUT

mets	List of metabolites

2.2.5.2 MakeModel functions

The makeModel tool is a set of functions to import data from a csv file and to convert it to a CBM structure and a sbml file. It use as an external functions csvimport.m and creatCBModel.m to be used as independent tool.

makeCBModelFromCSV

makeCBModelFromCSV(fileName,delimiter,modelName,metExt,lbExt,ubExt)

Create a model from a CSV file

INPUTS

filename	Name of the CSV file
delimiter	Delimiter of the CSV file
modelName	Name of the output CBModel file
metExt	List of external metabolites
lbExt	Lower bound for the exchange flux of external metabolites
ubExt	Upper bound for the exchange flux of external metabolites

OUPUT

Model	A CBModel

findExtMet

mets=findExtMet(model,suffix)

Find External metabolites based on a suffix

INPUTS

model	A CBModel
suffix	The suffix of the external metabolites

OUTPUT

Mets	List of external metabolites