FastMM Manual

--------Written by GongHua Li, PhD

Version 1.0.0

Mail to: ligonghua@mail.kiz.ac.cn or

ligonghuabio@gmail.com

1.What is FastMM?

FastMM was designed to fast implement constraint based metabolic modeling. It contains nine core programms: FBA, FVA, singleGeneKO, singleGeneKI, doubleGeneKO, singleMetKO, doubleMetKO, . All of the nine core programms are written by C and precomplied in linux. FastMM is under GPL licence, thus, it is completely free for all users.

In addition, we also developed Matlab interface for fast reconstruction, modeling and sampling.

The matlab interface was multithread and makes FastMM has the ability to solve large-scale constraint-based metabolic model analysis. For example, we can perform a large-scale reconstruction, personal target s, personal antimotabilites and biomarker analysis from hundreds to thousands TCGA samples.

2. Why do we use FastMM?

Recently, the most popular toolbox for constraint-based metabolic model analysis is COBRA[1]. Usually, the COBRA can used to solve the following three biology problem. 1) understand metabolism related disease mechanisms by MCMC sampling. 2) inferring new potential drug targets by single or multiple gene knockout analysis (Folger, et al., 2011), 3) inferring potential biomarkers by flux variability analysis (FVA) (Shlomi, et al., 2009), 4)inferring potential antimetabolites by single or multiple metabolite knockout analysis (Agren, et al., 2014).

However, although the COBRA is easy use and popular, but it is inefficient for metabolic modeling. For example, the whole genome single gene knockout of consistent Recon2.03(5317 reactions, 1853 genes) to find the essential genes for cell growth spend **1070.1** seconds. However, using FastMM, it just spend **12.3** seconds. Of course, they returned the consistent results, both FastMM and COBRA identified three essential genes(686, 54675, and 259230) that is required for cell growth of consistent Recon2.03. FastMM is **86 times faster** than COBRA.

To find the synergistic lethal genes using whole genome double gene knockout method from consistent Recon2.03(5317 reactions, 1853 genes)[2, 3], FastMM spend 2.7 hours. But after 3 days later, COBRA2.0 still do not complete the task. The FastMM usually has **60~90 times faster** than COBRA.

All the tests were performed on the server with the system of Ubuntu 14.04lts, CPU of Intel Xeon X5650 2.67GHz and the memory of 16GB, and used 1 threading.

Here is the detailed comparsion of FastMM and COBRA 2.0:

Table 1. Comparison of the time cost of metabolic modeling between FastMM and other software using GUROBI solver.

Applications	COBRA 2.0	Cobrapy	fastFVA	Fast-SL	FastMM
FVA	15494 s	99 s	**85 s	\	***80 s
SingleGeneKO	1143 s	15 s	\	21 s	9 s

DoubleGeneKO	>7 d	*2324 s	\	3161 s	260 s
SingleMetKO	1368 s	\	\	\	8 s
DoubleMetKO	>7 d	\	\	\	578 s
MCMC	2179 s	\	\	\	254 s

Note: The consistent general human metabolic model Recon 2.03 (5317 reaction, 2960 metabolites, and 2194 genes) was used in the analyses. The symbols of "s", "h", and "d" represent seconds, hours, and days, respectively. MCMC sampling uses the parameter of points of 2000, and steps of 1000. All FastMM applications used one CPU cores, other software used default parameters. \: not available for the corresponding applications. *: Cobrapy used 4 CPU cores to perform double gene knockout. **: fastFVA do not support the GUROBI solver, the time cost (85 s) was calculated using GLPK solver. ***: FastMM consumed 80 and 107 seconds to conduct flux variability analysis using GUROBI and GLPK solver , respectively.

3. Installation

There are two different ways to installation

3.1 Use precompiled binaries.

In matlab, users just type:

```
Install
```

3.2 Compile from source code

Make sure your platform include gcc and g++ environment. Download the linux version of FastMM from **http://bsb.kiz.ac.cn/site_media/download/FastMM/FastMM.tar.gz** and then type the following in the command window:

```
tar -zxf FastMM.tar.gz

cd FastMM

./install
```

It will take some minutes. If it success, you will find six core programs in ./bin subdictionary: FBA, FVA, singleGeneKO, singleGeneKI, doubleGeneKO, singleMetKO, doubleMetKO.

Note: in this version, we have precompiled the Linux version of FastMCMC, to compile the FastMCMC in MAC, please contact the author: ligonghua@mail.kiz.ac.cn. The win32/64 version of FastMCMC was not supported.

3.3. Dependency

The core programs of FastMM are independent. However, the matlab interface of FastMM need the following 4 additional software:

1). Install MATLAB. MATLAB is the software environment for engineers and scientists. You can get detailed installation information by visiting http://www.mathworks.com/products/matlab/

2). Install Cobra toolbox in Matlab. Cobra toolbox is a matlab toolbox that used to implement metabolic modeling. Users can download and install this toolbox from: https://opencobra.github.io/cobratoolbox/

3). Install Gurobi optimizer MATLAB interface. Please download and install the latest Gurobi optimizer and set MATLAB interface. Users can download and install this toolbox from: https://www.gurobi.com/

4). Install Cplex optimizer MATLAB interface. Users can download the cplex from http://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition/

4. How to use

We will use the matlab interface to show how to use FastMM

4.1 One-command use

For one-command usage, the users do not need strong programming ability and do not need strong metabolic modeling background, users just need to set up the parameters in the file "pars.txt", and type one command in matlab:

```
FastMM
```

The detailed parameters information could be found in "pars.txt". Briefly, to perform personalized metabolic modeling, users just need set the path of gene expression matrix and the path of consistent general metaoblic model, such as consistRecon2_v3.mat

Then all of the results will saved in the ./out subdirectory, these results including:
Model reconstruction results: saved in ./out/reconstructed_model.txt
FVA results: saved in ./out/FVAmin.txt and ./out/FVAmax.txt
single Gene knockout results: ./out/singleGeneKO.txt
single metabolite knockout results: ./out/singleMetKO.txt
mcmc results: the mcmc sampling results are saved in the ./out/mcmc/ subdirectory

4.2 Advanced use

Beside the "one-command" method for metabolic modeling, FastMM also provide the advanced use for various applications:

In general, all of the core FastMM can be directly used in matlab as following:

output = function_name(model)

For example, for single gene konckout analysis, we can use it the following command in matlab:

```
Output = FastMM_singleGeneKO(model)
```

All of the core programs have the bellow options, such as **FastMM_singleGeneKO:**

-m: model name, which is required for all the core programs. For example the model of “consRecon2” is automatically generated from the consistent generic human model Recon2.03 in matlab[3].

-t: type or the sense of the optimization. min or max represents to minimization or maximization of the objective function. default is max.

-f: objective function file(optimal). Default is biomass_reaction, users can also define the objective function (such as ATP production, and uptake/secrete reactions). FastMM accepts multiple objective functions.

-c: additional constraint file. For example, when we apply FVA, we may need to fix the biomass reaction to maximum or at least 0.6 maximum.

4.2.1 User defining objective function.

Beside biomass_reaction, many other biological reactions may also need to be optimized, such as ATP production, uptake/secret production. Thus user defining objective function is widely used in knockout analysis.

In FastMM, the number of objective function is not limited. Thus, the extremely cases is that one can use all of the reactions as the objective function, and find how each gene knockout effect every reaction. The objective function file is very simple: each line contains an objective function
For example: if we need to find the optimization of ATP production, dATPm production, and dCTP production in consRecon2. We can write a text file, suppose named “objectives.txt”, then this file will contain 3 lines:

```
DM_atp_c_
DM_datp_m_
DM_dctp_m_
```

Then, we can use FastMM_singleGeneKO:

```
Output = FastMM_singleGeneKO(model,'-f objectives.txt')
```

4.2.2 User defining additional constraint

As mentioned above, we may need additional constraint for knockout analysis. For example, we may want to know when fix the ability of cell growth, whether knockout genes can affect the production of ATP, dATP and dCTP. We can set the cutoff of growth rate as 0.6 compared to wide type. Then we can write a text file, suppose names “constraint.txt”, which contains one line:

```
biomass_reaction  0.6
```

We now can use all of the advanced option of singleGeneKO:

```
Output = FastMM_singleGeneKO(model,'-f objectives.txt -c constraint.txt')
```

4.3 Output interpretation

4.5.1 FVA output

Using FastMM FVA through the command of *[fluxmin, fluxmax] = FastMM_FVA(model)*
The outputs of *fluxmin, fluxmax* are the minimal and maximal flux vector for each reaction of the model, respectively.

4.3.2 knockout output

Using FastMM knockout analysis through the commands of *output = FastMM_...KO(model).* The outputs is a matrix with >= 3 columns. The first and second column is the index of pairs of knockout genes or metabolites in the input model. Other columns is the objective function values. For example, if we just use one objective function, such as the 'biomass_reaction'. The output will contain 3 column. The first and second column is the index of genes or metabolites. The third column is the maximum of biomass_reaction in different genes/metabolites knock out. Index of 0 represents no knockout conducting.

4.3.3 MCMC output

Using the FastMM MCMC through the commands of outmodel = FastMM_MCMC(model,...). The outmodel is a structure that contains the "points" field. This points is a n × s matrix. Where n is the number of reactions in input model, s is the points user defined. Note: in this version(1.0.0), only Linux version of MCMC was supported .

5. Example and multithreading

Because FastMM effectively covered the limitation of the time cost of genome-wide knockout analysis of COBRA 2.0, the most important advantage of FastMM is to implement large scale genome-wide metabolic modeling. We now apply FastMM to 528 TCGA individual samples to infer the individual cancer metabolic profiles.

5.1 Reconstruction

Many methods can be used to reconstruct tissue specific metabolic model, such as MBA, and fastcore. In this example, we used fastcore to reconstruct individual metabolic model, since it makes a well balance between the time cost and accuracy.

The procedure of reconstruction was implemented using the matlab function of "reconstruction_by_fastcore", we constructed the 528 individual TCGA lung cancer models using the matlab function . There are 58 normal samples are 470 lung cancer samples. In this procedure, we select the cutoff of RESM as 75, because RESM is about 25 times larger than RPKM[4], where RPKM >3 was consider as expressed or highly expressed []. Actually, there are about 10000~12000 genes with the

RESM >75, which is consistent with Ramskold's result[5].

The reconstruction can be implemented very simple in matlab:

```
# matlab wrap FastMM_doubleGeneKO_multi.m was specifically for multiple
# threading for multiple sample double gene knockout analysis

rxnsmatrix = reconstruction_by_fastcore(consmodel,expr,75);
```

Figure 1 shows the distribution of number of reactions of 58 normal and 470 cancer lung metabolic models. Overall, the number of reaction of lung cancer metabolic models is larger than that of normal lung models (p = 0.02, t-test).

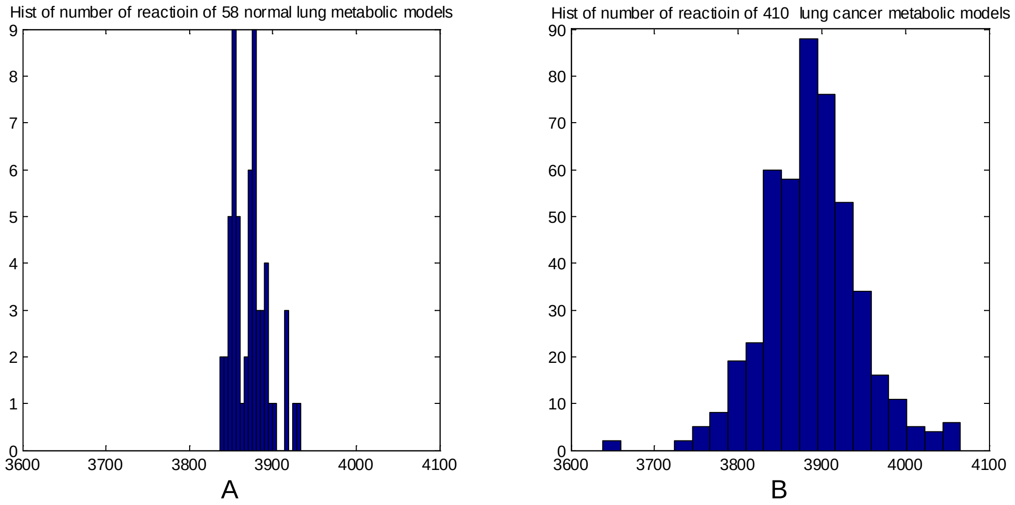

Figure 1: Histogram plot of reactions of normal (A) and cancer (B).

Figure 2. Overall, the number of metabolites of lung cancer metabolic models is larger than that of normal lung models (p <0.01, t-test).

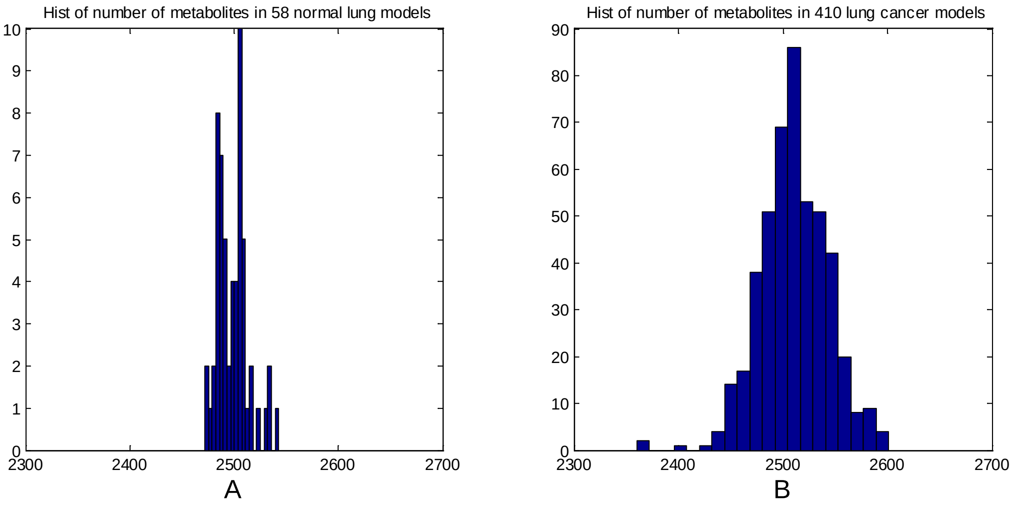

Figure 2: Histogram plot of metabolites of normal (A) and cancer (B).

5.2 Target genes: Large scale genome wide single and double gene knockout analysis

Because the program doubleGeneKO contains both single and double gene knockout results, we can implement single and double gene knockout analysis by doubelGeneKO.

We recommend that users use multiple threading to implement large scale genome wide knockout analysis. For example, we can use 8 threading to implement single & double gene knockout analysis using matlab interface, this procedure is very simple:

```
# matlab wrap for multi threading double gene knockout:
# FastMM_doubleGeneKO_multi.m
# usage: out = FastMM_doubleGeneKO_multi(model,rxnsmatrix,numCPU)
# eg. in this case:

GeneKOout = FastMM_doubleGeneKO_multi(consmodel,rxnsmatrix,8)
```

The result was showed in Figure 3.

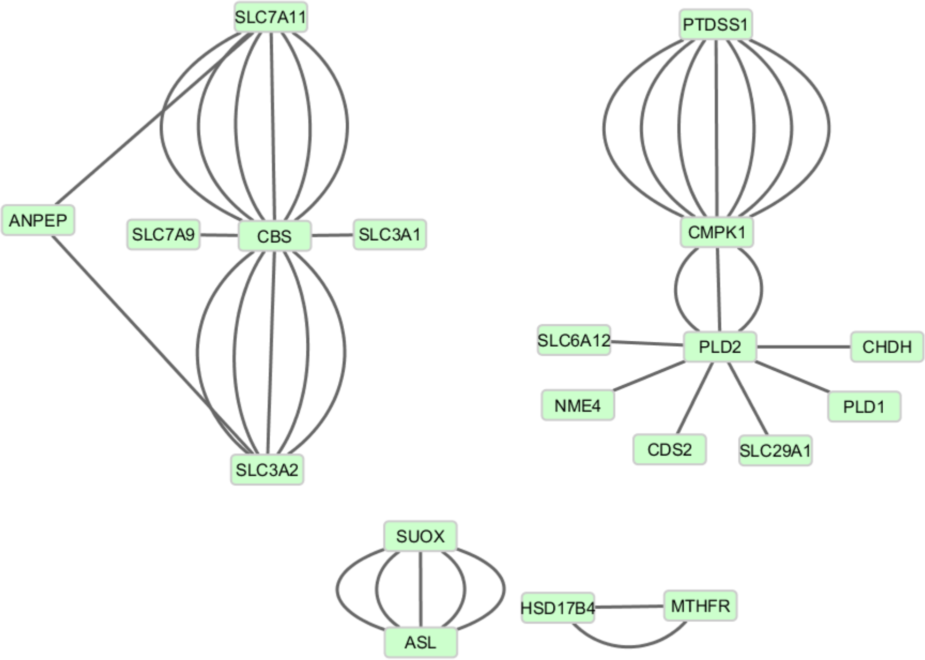

Figure 3: lung cancer specific synergistic lethal genes. Node represents specific cancer specific gene which is neither a lethal gene nor one of synergistic lethal genes in all normal lung tissue. Edge represents the linked genes are lung cancer specific synergistic lethal.

5.3 Biomarkers: secreting reactions

Now we can use FastMM to analysis metabolite biomarker and used to target environment.
First, we need to know which metabolites are secreted into the environment. Using FVA and constrain the cell growth: use the multithreading matlab wrap(58 normal, and 470 cancer samples, 8 threading):

```
## matlab wrap program for multithreading FVA: FastMM_FVA_multi.m
## Usage:  [fluxmin,fluxmax] =  FastMM_FVA_multi(model,rxnsmatrix,numCPU)
## Example in this case :

[fluxmin,fluxmax] =  FastMM_FVA_multi(consmodel,rxnsmatrix,8)
```

The specific secret reaction shows in Table 1

Table 1: secret reactions in lung cancer samples

Secret reaction	**# of Occurance**
EX_lac_D(e)	118
EX_fuc13galacglcgal14acglcgalgluside_hs(e)	89

EX_fucacngal14acglcgalgluside_hs(e)	89
EX_fucfuc12gal14acglcgalgluside_hs(e)	89
EX_fucfuc132galacglcgal14acglcgalgluside_hs(e)	89
EX_fucfucfucgalacglcgal14acglcgalgluside_hs(e)	89
EX_fucgal14acglcgalgluside_hs(e)	89
EX_4hdebrisoquine(e)	88
EX_24nph(e)	69
EX_gp1calpha_hs(e)	66
EX_13_cis_retnglc(e)	60
EX_retn(e)	50
EX_bildglcur(e)	46
EX_estradiolglc(e)	46
EX_retnglc(e)	46
EX_CLPND(e)	39
EX_retfa(e)	39
EX_ala_B(e)	31
EX_ppa(e)	31
EX_dgchol(e)	27
EX_akg(e)	24
EX_4mtolbutamide(e)	23
EX_apnnox(e)	23
EX_carveol(e)	23
EX_perillyl(e)	23
EX_5mta(e)	12
EX_5oxpro(e)	10
EX_5homeprazole(e)	9
EX_CE4881(e)	8
EX_htaxol(e)	7
EX_c12dc	6
EX_prostgh2(e)	6
EX_prostgi2(e)	6
EX_4abut(e)	5
EX_C02470(e)	5
EX_CE2011(e)	5
EX_4abutn(e)	4
EX_for(e)	4
EX_estrones(e)	3
EX_gd1c_hs(e)	3
EX_gq1b_hs(e)	3
EX_prgstrn(e)	3
EX_ump(e)	3
EX_HC01444(e)	2
EX_c4crn	2
EX_glygn5(e)	2

EX_2mcit(e)	1
EX_34dhphe(e)	1
EX_Rtotal2(e)	1
EX_aldstrn(e)	1
EX_thyox_L(e)	1
EX_tststerones(e)	1
EX_whddca(e)	1
EX_whhdca(e)	1
EX_whttdca(e)	1

5.4 Target environment: Which genes block secreting?

Using singleGeneKO, we will analysis which gene knockout affects the metabolite secret, that is: using the secret reations from FVA as the objective function. We can use a matlab interface of single gene deletion programm:

```
## matlab wrap program for multithreading multi-objective function single
gene knock out analysis: FastMM_singleGeneKO_multi.m
## Usage:  out = FastMM_singleGeneKO_multi(model,rxnsmatrix,numCPU,'-f
objectivfile -c constraint')
## Example in this case :

Out = FastMM_singleGeneKO_multi(consmodel,rxnsmatrix,8,'-f secret_rxns.txt')
```

Then, we can obtain the essential genes for each secret reaction (Table 2). The result indicates that 97 genes can change the metabolite secreting. Knockout of some genes, such as SLC35D1 and UGCG, do not directly affect the cancer cell growth, but may dramatically alter the cancer cell micro-environment.

Table 2: Essential genes for secret reactions in lung cancer samples

Exchange Rxns	**# of patients**	**Essential genes for secreting**
EX_lac_D(e)	118	
EX_fuc13galacglcgal14acglcgalgluside_hs(e)	89	B3GNT2,B3GNT3,B3GNT5,COL4A3BP,DHCR24,FUT9,SLC35C1,ENTPD4,SLC35A2,SLC35D2,UGCG,,SLC35D1
EX_fucacngal14acglcgalgluside_hs(e)	89	B3GNT3,B3GNT5,COL4A3BP,SLC35A1,CMPK1,DHCR24,FUT9,SLC35C1,ENTPD4,ST3GAL6,SLC35A2,SLC35D2,UGCG,,SLC35D1
EX_fucfuc12gal14acglcgalgluside_hs(e)	89	B3GNT3,B3GNT5,COL4A3BP,DHCR24,FUT1,FUT9,SLC35C1,ENTPD4,SLC35A2,SLC35D2,UGCG,,SLC35D1

EX_fucfuc132galacglcgal14acglcgalgluside_hs(e)	89	B3GNT2,B3GNT3,B3GNT5,COL4A3BP,DHCR24,FUT9,SLC35C1,ENTPD4,SLC35A2,SLC35D2,UGCG,,SLC35D1
EX_fucfucfucgalacglcgal14acglcgalgluside_hs(e)	89	B3GNT2,B3GNT3,B3GNT5,COL4A3BP,DHCR24,FUT1,FUT9,SLC35C1,ENTPD4,SLC35A2,SLC35D2,UGCG,,SLC35D1
EX_fucgal14acglcgalgluside_hs(e)	89	B3GNT3,B3GNT5,COL4A3BP,DHCR24,FUT9,SLC35C1,ENTPD4,SLC35A2,SLC35D2,UGCG,,SLC35D1
EX_4hdebrisoquine(e)	88	
EX_24nph(e)	69	CYP2E1
EX_gp1calpha_hs(e)	66	B3GNT3,B3GALT4,COL4A3BP,SLC35A1,CMPK1,DHCR24,B4GALNT1,ENTPD4,ST3GAL5,ST6GALNAC2,ST8SIA1,ST8SIA5,SLC35A2,SLC35D1,UGCG,
EX_13_cis_retnglc(e)	60	BCMO1,,UGDH,SLC35D1,UGT1A1,SLC7A9,RDH5
EX_retn(e)	50	SLC7A9,BCMO1,SLC26A4,PLA2G7,RDH5,SLC16A1
EX_bildglcur(e)	46	FASN,,SLC16A1,UGDH,SLC35D1,UGT1A7
EX_estradiolglc(e)	46	,UGDH,SLC35D1,ABCC4
EX_retnglc(e)	46	BCMO1,,UGDH,SLC35D1,SLC7A9
EX_CLPND(e)	39	ACACA,ACACB,ACSL1,ELOVL4
EX_retfa(e)	39	BCMO1,SLC27A4
EX_ala_B(e)	31	
EX_ppa(e)	31	SLC7A9,SLC26A4,,UGDH,SLC35D1,SLC16A1,ACSM1,UGT1A1
EX_dgchol(e)	27	SLC25A20,CRAT,SLC27A2,SCP2,CYP27A1,NSDHL,HSD17B4,MSMO1,STARD3,DHCR7,MVD,EBP,HMGCR,LSS,SC5D,MVK,PMVK,SQLE,FASN,AKR1D1,HSD3B7,CYP7A1
EX_akg(e)	24	SLC7A9,SLC26A4,UGDH,UGT1A1,,SLC35D1
EX_4mtolbutamide(e)	23	CYP2C9
EX_apnnox(e)	23	
EX_carveol(e)	23	
EX_perillyl(e)	23	
EX_5mta(e)	12	AMD1
EX_5oxpro(e)	10	GGCT,ANPEP
EX_5homeprazole(e)	9	CYP2C19
EX_CE4881(e)	8	LPO
EX_htaxol(e)	7	

EX_c12dc	6	ADH5,ACAA1,ACOX1,ALDH3A2,EHHADH,HSD17B4,CAT,SLC22A5,CYP4F3,CYP4F2
EX_prostgh2(e)	6	
EX_prostgi2(e)	6	PTGIS
EX_4abut(e)	5	ALDH9A1,SLC36A1
EX_C02470(e)	5	KMO,TDO2,CCBL1
EX_CE2011(e)	5	LPO
EX_4abutn(e)	4	ODC1,RPIA
EX_for(e)	4	SLC26A4
EX_estrones(e)	3	BPNT1
EX_gd1c_hs(e)	3	B3GNT3,B3GALT4,COL4A3BP,SLC35A1,CMPK1,DHCR24,B4GALNT1,ENTPD4,ST3GAL2,ST8SIA5, SLC35A2,UGCG,,SLC35D1
EX_gq1b_hs(e)	3	B3GNT3,B3GALT4,COL4A3BP,SLC35A1,CMPK1,DHCR24,B4GALNT1,ENTPD4,ST3GAL2,ST3GAL5, ST8SIA1,ST8SIA5,SLC35A2,UGCG,,SLC35D1
EX_prgstrn(e)	3	HSD3B1,CYP11A1
EX_ump(e)	3	
EX_HC01444(e)	2	LIPC
EX_c4crn	2	SLC22A5
EX_glygn5(e)	2	SI
EX_2mcit(e)	1	
EX_34dhphe(e)	1	SLC16A10
EX_Rtotal2(e)	1	
EX_aldstrn(e)	1	HSD3B2,CYP11A1,CYP11B2,CYP21A2
EX_thyox_L(e)	1	SLC5A8
EX_tststerones(e)	1	SULT2A1,BPNT1
EX_whddca(e)	1	CYP4A22
EX_whhdca(e)	1	CYP4A22
EX_whttdca(e)	1	CYP4A22

5.5 Target metabolites: which metabolite affects cell growth?

Analysis lethal and synergistic lethal metabolites using double metabolite knockout.

```
## matlab interface for multithreading doubleMetKO:
FastMM_doubleMetKO_multi.m
## Usage:  out = FastMM_doubleMetKO_multi(model,rxnsmatrix,numCPU)
## In this case:

metout = FastMM_doubleMetKO_multi(model,rxnsmatrix,numCPU)
```

The result suggests that the pair of glycylproline and L-Prolinylglycine is the cancer specific synergistic lethal metabolites.

Reference

1. Schellenberger J, Que R, Fleming RM, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S *et al*: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0**. *Nat Protoc* 2011, **6**(9):1290-1307.
2. Thiele I, Swainston N, Fleming RM, Hoppe A, Sahoo S, Aurich MK, Haraldsdottir H, Mo ML, Rolfsson O, Stobbe MD *et al*: **A community-driven global reconstruction of human metabolism**. *Nat Biotechnol* 2013, **31**(5):419-425.
3. Vlassis N, Pacheco MP, Sauter T: **Fast reconstruction of compact context-specific metabolic network models**. *PLoS Comput Biol* 2014, **10**(1):e1003424.
4. Guo Y, Sheng Q, Li J, Ye F, Samuels DC, Shyr Y: **Large scale comparison of gene expression levels by microarrays and RNAseq using TCGA data**. *PLoS One* 2013, **8**(8):e71462.
5. Ramskold D, Wang ET, Burge CB, Sandberg R: **An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data**. *PLoS Comput Biol* 2009, **5**(12):e1000598.