

Creation and simulation of personalized microbiota models through metagenomic data integration

Author: Federico Baldini, Molecular Systems Physiology Group, University of Luxembourg.

INTRODUCTION

This tutorial shows the steps that MgPipe automatically performs to create and simulate personalized microbiota models through metagenomic data integration.

The pipeline is divided in 3 parts:

1. **[PART 1]** Analysis of individuals' specific microbes abundances is computed. Individuals' metabolic diversity in relation to microbiota size and disease presence as well as Classical multidimensional scaling (PCoA) on individuals' reaction repertoire are examples.
2. **[PART 2]:** 1 Constructing a global metabolic model (setup) containing all the microbes listed in the study. 2 Building individuals' specific models integrating abundance data retrieved from metagenomics. For each organism, reactions are coupled to their objective function.
3. **[PART 3]** A specific range of growth is imposed for each microbiota model and Simulations under specific diet regimes are carried. Set of standard analysis to apply to the personalized models. PCA of computed MNPCs of individuals as for example.

USAGE

Normally, once provided all the input variables in the driver (StartMgPipe), the only action required is to run the driver itself. However, for the purposes of this tutorial, we will disable the autorun functionality and compute each section manually.

DRIVER

This file has to be modified from the user in order to launch the pipeline and to define inputs and outputs files and locations.

We first set the paths to input and output files

```
% path to microbiota models
modPath='YOUR_PATH_TO_AGORA\';
% path to where the Microbiome Modeling Toolbox is located
toolboxPath='YOUR_PATH_TO_Microbiome_Modeling_Toolbox\'
```

```
toolboxPath = YOUR_PATH_TO_Microbiome_Modeling_Toolbox\
```

```
% % path where to save results
resPath='YOUR_PATH_TO_RESULT_FOLDER' ;
% path to and name of the file with dietary information. Here,
```

```
% we will use an "Average European" diet that is located in the
% DietImplementation folder.
```

```
modPath = /Users/almut.heinken/Documents/Unconstrained/mat/
toolboxPath = /Users/almut.heinken/Documents/GitHub/Microbiome_Toolbox/
resPath = /Users/almut.heinken/Documents/test/
```

```
dietFilePath=strcat(toolboxPath, 'DietImplementation/AverageEuropeanDiet');
```

Then we set the path and the name of the file from which to load the abundances. For this tutorial, in order to reduce the time of computations, we will use a reduced version of the example file (normCoverageReduced.csv) provided in the folder Resources: only 4 individuals and 30 strains will be considered. Please note that abundances are normalized to a total sum of one.

```
abunFilePath=strcat(toolboxPath, 'Resources/normCoverageReduced.csv');
```

Next inputs will define:

1. name of objective function of organisms
2. name of diet to use
3. format to use to save images
4. solver (and interface) to use
5. number of cores to use for the pipeline execution
6. if to enable automatic detection and correction of possible bugs
7. if to enable compatibility mode
8. if health status for individuals is provided
9. if to simulate also a rich diet
10. if to use an external solver and save models with diet
11. the type of FVA function to use to solve

The following setting should work for almost any system, but please check carefully to be sure these options are valid for you. A more detailed description of these variables is available in the documentation.

The same inputs need to be set in the driver file StartMgPipe when running MgPipe outside of this tutorial.

```
% name of objective function of organisms
objre={'EX_biomass(e)'};
% the output is vectorized picture, change to '-dpng' for .png
figForm = '-depvc';
% which solver (and interface) to use
```

Where is the solver variable used? I can't find that anywhere

and the pipeline crashed because the solver was not set!-Almut %%%%

```
solver = 'tomlab_cplex'
```

```
solver = tomlab_cplex
```

```
% number of cores dedicated for parallelization
```

```
numWorkers = 3;
% autofix for names mismatch
autoFix = 1
```

```
autoFix = 1
```

```
% if outputs in open formats should be produced for each section (1=T)
compMod = 0;
% if documentations on patient health status is provided (0 not 1 yes)
patStat = 0;
% to enable also rich diet simulations
rDiet = 0;
% Add description-Almut
extSolve = 0;
% Add description-Almut
fvaType = 1;
% Additionally we will turn off the autorun to be able to manually execute each part of the p
autorun=0;
if autorun==1
    if numWorkers >= 2
        poolobj=gcp('nocreate');
        if isempty(poolobj)
            parpool(numWorkers)
        end
    end
    disp('Well done! Pipeline successfully activated and running!')
    MgPipe
else
    if numWorkers >= 2
        poolobj=gcp('nocreate');
        if isempty(poolobj)
            parpool(numWorkers)
        end
    end
    warning('autorun function was disabled. You are now running in manual / debug mode. If thi
    edit('MgPipe.m')
end
```

Warning: autorun function was disabled. You are now running in manual / debug mode. If this is not what you wanted, change back to ?autorun?=1. Please note that the usage of manual mode is strongly discouraged and should be used only for debugging purposes.

```
if compMod == 1
    warning('compatibility mode activated. Output will also be saved in .csv / .sbml format. T
else
    warning('pipeline output will be saved in .mat format. Please enable compomod option if yo
end
```

Warning: pipeline output will be saved in .mat format. Please enable compomod option if you wish to activate compatibility mode.

```
if numWorkers<2
    warning('apparently you disabled parallel mode to enable sequential one. Computations might
end
if patStat==0
    disp('Individuals health status not declared. Analysis will ignore that.')
end
```

Individuals health status not declared. Analysis will ignore that.

PIPELINE: [PART 1]

The number of organisms, their names, the number of samples and their identifiers are automatically detected from the input file.

```
[patNumb, sampName, strains]=getIndividualSizeName(abunFilePath)
```

```
patNumb = 4
sampName =
    'SRS011061'
    'SRS012902'
    'SRS011134'
    'SRS054956'

strains =
    'Bacteroides_cellulosilyticus_DSM_14838'
    'Parabacteroides_merdae_ATCC_43184'
    'Bacteroides_vulgatus_ATCC_8482'
    'Alistipes_putredinis_DSM_17216'
    'Bacteroides_sp_4_3_47FAA'
    'Bacteroides_caccae_ATCC_43185'
    'Burkholderiales_bacterium_1_1_47'
    'Bacteroides_sp_20_3'
    'Bacteroides_ovatus_ATCC_8483'
    'Bacteroides_sp_2_1_33B'
    'Bacteroides_intestinalis_341_DSM_17393'
    'Bacteroides_sp_3_1_19'
    'Parabacteroides_sp_D13'
    'Parabacteroides_distasonis_ATCC_8503'
    'Bacteroides_sp_2_1_7'
    'Faecalibacterium_prausnitzii_M21_2'
    'Bacteroides_sp_9_1_42FAA'
    'Bacteroides_sp_2_2_4'
    'Bacteroides_plebeius_M12_DSM_17135'
    'Bacteroides_sp_3_1_23'
    'Bacteroides_ovatus_SD_CMC_3f'
    'Faecalibacterium_prausnitzii_L2_6'
    'Parabacteroides_johnsonii_DSM_18315'
    'Bacteroides_coprocola_M16_DSM_17136'
    'Roseburia_intestinalis_L1_82'
    'Bacteroides_sp_1_1_6'
    'Bacteroides_eggerthii_1_2_48FAA'
    'Bacteroides_dorei_DSM_17855'
    'Bacteroides_sp_D2'
    'Bacteroides_finegoldii_DSM_17565'
```

Now we detect from the content of the results folder if PART1 was already computed: if the associated file is already present in the results folder its execution is skipped else its execution starts

```
[mapP]=detectOutput(resPath, 'mapInfo.mat');

if ~isempty(mapP)
    s= 'mapping file found: loading from resPath and skipping [PART1] analysis';
    disp(s)
    load(strcat(resPath, 'mapInfo.mat'))
end
```

mapping file found: loading from resPath and skipping [PART1] analysis

In case PART 1 was not computed we will compute it now. We will first load the models and create a cell array containing them. This cell array will be used as input by many functions in the pipeline. Any possible constraint from each model reactions will be removed. Moreover we will run and subsequently plot the results of some analysis that are computed. The main outputs are:

1. **Metabolic diversity** The number of mapped organisms for each individual compared to the total number of unique reactions (extrapolated by the number of reactions of each organism)
2. **Classical multidimensional scaling of each individual reactions repertoire**

```
if isempty(mapP)
% Loading models
models=loadUncModels(modPath,straains,objre);
% Computing genetic information
[reac,micRea,binOrg,patOrg, reacPat, reacNumb, reacSet, reacTab, reacAbun, reacNumber]=getMappingInfo(modPath,models)
writetable(cell2table(reacAbun),strcat(resPath,'reactions.csv'))

% Plotting genetic information
[PCoA]=plotMappingInfo(resPath,patOrg, reacPat, reacTab, reacNumber, patStat, figForm)

if compMod==1
    mkdir(strcat(resPath,'compfile'))
    csvwrite(strcat(resPath,'compfile/reacTab.csv'),reacTab)
    writetable(cell2table(reacSet),strcat(resPath,'compfile/reacset.csv'))
    csvwrite(strcat(resPath,'compfile/reacNumb.csv'),reacNumb)
    csvwrite(strcat(resPath,'compfile/ReacPat.csv'),reacPat)
    csvwrite(strcat(resPath,'compfile/PCoA_tab.csv'),Y)
end

%Save all the created variables
save(strcat(resPath,'mapInfo.mat'))
end
```

```
reac =
'12DGR180ti'
'12PPDt'
'1P4H2CBXLAH'
'23DHMP0'
'23PDE2'
'23PDE4'
'23PDE7'
'23PDE9'
'26DAPLLAT'
'26DAPt2r'
'2AHBUTI'
'2DHPL'
'2HB0'
'2HYOXPLACt2r'
'2INSD'
'2MBCOATA'
'2MMALD'
'2MMALD2'
'20BUTt2r'
'20MMBOX'
'20MPHH'
'34DHOXPEG0X'
'36DAHXI'
'3ABUTCOAL'
```

'3CARLPDH'
'3HACPR1'
'3HACPR2'
'3HAD100'
'3HAD10M11'
'3HAD10M12'
'3HAD11M12'
'3HAD120'
'3HAD12M13'
'3HAD12M14'
'3HAD13M14'
'3HAD140'
'3HAD14M15'
'3HAD14M16'
'3HAD15M16'
'3HAD160'
'3HAD180'
'3HAD40'
'3HAD4M5'
'3HAD4M6'
'3HAD5M6'
'3HAD60'
'3HAD6M7'
'3HAD6M8'
'3HAD7M8'
'3HAD80'
'3HAD8M10'
'3HAD8M9'
'3HAD9M10'
'3HBCOAE'
'3MOBS'
'3MOPt2r'
'30AACPR1'
'30AACPR2'
'30AR100'
'30AR10M11'
'30AR10M12'
'30AR11M12'
'30AR120'
'30AR12M13'
'30AR12M14'
'30AR13M14'
'30AR140'
'30AR14M15'
'30AR14M16'
'30AR15M16'
'30AR160'
'30AR180'
'30AR40'
'30AR4M5'
'30AR4M6'
'30AR5M6'
'30AR60'
'30AR6M7'
'30AR6M8'
'30AR7M8'
'30AR80'
'30AR8M10'
'30AR8M9'
'30AR9M10'
'30AS100'
'30AS10M11'
'30AS10M12'
'30AS11M12'
'30AS120'
'30AS121'
'30AS12M13'
'30AS12M14'

'30AS13M14'
'30AS140'
'30AS141'
'30AS160'
'30AS161'
'30AS180'
'30AS181'
'30AS2M3'
'30AS3M4'
'30AS4M306'
'30AS4M5'
'30AS4M6'
'30AS5M6'
'30AS60'
'30AS6M7'
'30AS6M8'
'30AS7M8'
'30AS80'
'30AS8M10'
'30AS8M9'
'30AS9M10'
'4HBZt2'
'4HTHRS'
'4NTPPH2'
'5AIZCDC'
'5DGLCNR'
'5DOAN'
'5GLUP5T1'
'5GLUP5T2'
'6PHBG'
'6PHBG2'
'7AHSDH'
'7AHSDHe'
'A5PISO'
'AAC0AT'
'AACPS6'
'AACPS7'
'AADSACYCL'
'AB6PGH'
'AB6PGH2'
'ABTOXy'
'ABUTD'
'ACACT1r'
'ACACT2r'
'ACACT3r'
'ACACT4r'
'ACACT5r'
'ACACT6r'
'ACACT7r'
'ACALD'
'ACALDt'
'ACBIPGT'
'ACCOAC'
'ACCOACL'
'ACCOALYS2'
'ACGALK3'
'ACGALt2r'
'ACGAM2E'
'ACGAMK'
'ACGAMPM'
'ACGAMT'
'ACGAMtr2'
'ACGApts'
'ACGK'
'ACGS'
'ACHBPL'
'ACKr'
'ACLS_a'

'ACLS_b'
'ACMANAtr'
'ACNAMS'
'ACNAMabc'
'ACNAMt2r'
'ACNML'
'ACOAAGPAT140'
'ACOAAGPAT160'
'ACOAAGPAT180'
'ACOAAGPATAI15'
'ACOAAGPATAI17'
'ACOAAGPATI14'
'ACOAAGPATI15'
'ACOAAGPATI16'
'ACOAAGPATI17'
'ACOAD1'
'ACOAD1i'
'ACOAD20'
'ACOADH2'
'ACOATA'
'ACOCT'
'ACODAr'
'ACONT'
'ACONTa'
'ACONTb'
'ACOTA'
'ACPS1'
'ACS'
'ACTNDH'
'ACYP'
'ACYP_2'
'ACtr'
'ADA'
'ADAD'
'ADCL'
'ADCPS1'
'ADCPS2'
'ADCS'
'ADCYRS'
'ADD'
'ADet2r'
'ADK1'
'ADK10'
'ADK11'
'ADK2'
'ADK3'
'ADK5'
'ADK7'
'ADK8'
'ADK9'
'ADMDCr'
'ADNCNT3tc'
'ADNCYC'
'ADNK1'
'ADNUC'
'ADOCBIK'
'ADOCBIK2'
'ADOCBLS'
'ADOCBLabc'
'ADPRDP'
'ADPT'
'ADSELK'
'ADSK'
'ADSL1r'
'ADSL2r'
'ADSS'
'AEPPT'
'AGDC'

'AGDC2'
'AGMD'
'AGMHE'
'AGMT'
'AGPAT120'
'AGPAT140'
'AGPAT141'
'AGPAT160'
'AGPAT161'
'AGPAT180'
'AGPAT181'
'AGPR'
'AGTi'
'AHC'
'AHCYSNS'
'AHCYSNS_r'
'AHEXASE3'
'AHMMPS'
'AHSERL2'
'AHSERL3'
'AICART'
'AIHDUDPMG'
'AIHDUPDG'
'AIHPDCOATA'
'AIPDCOATA'
'AIPDUDPG'
'AIPDUDPMG'
'AIRC4'
'AKGS'
'AKP1'
'ALAALA'
'ALAASP1c'
'ALAASPabc'
'ALACI'
'ALAD_R'
'ALAGLN1c'
'ALAGLNabc'
'ALAGLU1c'
'ALAGLUabc'
'ALAGLY1c'
'ALAGLYabc'
'ALAHIS1c'
'ALAHISabc'
'ALALEU1c'
'ALALEUabc'
'ALAPGPL'
'ALAR'
'ALATA_L'
'ALATHR1c'
'ALATHRabc'
'ALAt2r'
'ALAt4r'
'ALCD1'
'ALCD19'
'ALCD2x'
'ALCD4'
'ALDD1'
'ALKP'
'ALLTNRS'
'ALLTNS'
'ALPHNH'
'ALTRH'
'AMAA'
'AMALT1'
'AMALT2'
'AMALT3'
'AMALT4'
'AMANAPEr'

'AMANK'
'AMA0Tr'
'AMID'
'AMID2'
'AMID4'
'AMID5'
'AMMQT7rev'
'AMMQT8r'
'AMOPBHL'
'AMPMS2'
'AMPN'
'AMPS030X'
'AMPTASECG'
'AMPTASEGC'
'AMPt2r'
'AMY1e'
'AMY2e'
'AMYLOPECTe'
'AMYLE'
'ANPRT'
'ANS'
'ANS2'
'A0BUTDs'
'A0XHEXCYCL'
'A0XSr'
'A0XSr2'
'AP4AH'
'APAT'
'APPLDH'
'APRAUR'
'ARABASE3e'
'ARABINANASEe'
'ARABINOGALASEe'
'ARABINOXYL_DEGe'
'ARAI'
'ARBTpts'
'ARBabc'
'ARBt2r'
'ARGDA'
'ARGDC'
'ARGN'
'ARGSL'
'ARGSSr'
'ARGabc'
'ARGt2'
'ARGt2r'
'AROH'
'ARSENBabc'
'AS3TASE_HS1'
'AS3TASE_HS2'
'AS6TASE_HS1'
'AS6TASE_HS2'
'ASAD'
'ASNS1'
'ASNS2'
'ASNt2r'
'ASP1DC'
'ASPCT'
'ASPKi'
'ASP01'
'ASP05'
'ASP06'
'ASP07'
'ASPT'
'ASPTA'
'ASPabc'
'ASPt2'
'ASPt2_3'

'ASPt2r'
'ATPPRT'
'ATPS4'
'BACCLi'
'BC0AACT'
'BDH'
'BGLA'
'BGLA1'
'BGLAr'
'BGLC_DEGe'
'BIACt1'
'BIACt4'
'BPNT'
'BTC0AACCOAT'
'BTC0ADH'
'BTNCL'
'BTNCLi'
'BTNabc'
'BTS4'
'BUP2'
'BUTKr'
'BUTS03abc'
'BUTt2'
'BUTt2r'
'BZAMAH'
'CA2abc'
'CAt4i'
'CBIAT'
'CBL1abc'
'CBL2abc'
'CBLAT'
'CBMK'
'CBMTHL2'
'CBPS'
'CD2abc1'
'CD2t6r'
'CDCA7aHSDHe'
'CDOSKT'
'CDPEAS'
'CDPGHL'
'CDPGLYCGPT'
'CDPMEK'
'CELBpts'
'CELLBabc'
'CGLYabc'
'CGLYt3_2_'
'CHLt2r'
'CHOLD'
'CHOLSH'
'CHORM'
'CHORS'
'CHORt'
'CHRPL'
'CHTBSabc'
'CITL'
'CITRAMALS'
'CLPNS140'
'CLPNS160'
'CLPNS180'
'CLPNSAI15'
'CLPNSAI17'
'CLPNSI14'
'CLPNSI15'
'CLPNSI16'
'CLPNSI17'
'CMLDC'
'CMPSAS'
'C02t'

'COBALt5'
'COBCOM0X'
'CPC2MT'
'CPC3MT'
'CPC4MT'
'CPC5MT'
'CPC5MT1'
'CPC5MT2'
'CPC6MT'
'CPC6R'
'CPC8MM'
'CPPPG0'
'CPPPG02'
'CR04t3'
'CS4TASE'
'CS4TASE2'
'CS6TASE'
'CSABCASE_A_e'
'CSABCASE_B_e'
'CSABCASE_C_e'
'CSND'
'CSNt2'
'CSNt6'
'CTBTabc'
'CTPS1'
'CTPS2'
'CYF0'
'CYRDAAT'
'CYRDAR'
'CYRDAS'
'CYSDS'
'CYSS'
'CYSS3r'
'CYSTGL'
'CYSTL'
'CYSTS'
'CYSabc'
'CYSt2r'
'CYTBD'
'CYTBD2'
'CYTD'
'CYTDK1'
'CYTDK2'
'CYTDK3'
'CYTDK4'
'CYTDt2r'
'CYTK1'
'CYTK2'
'Coabc'
'Cuabc'
'Cut1'
'DADA'
'DADNK'
'DADNt2r'
'DAGK120'
'DAGK140'
'DAGK141'
'DAGK160'
'DAGK161'
'DAGK180'
'DAGK181'
'DAPDA'
'DAPDC'
'DAPE'
'DAPMDH'
'DAPabc'
'DASYN120'
'DASYN140'

'DASYN141'
'DASYN160'
'DASYN161'
'DASYN180'
'DASYN181'
'DASYNAI15'
'DASYNAI17'
'DASYNI14'
'DASYNI15'
'DASYNI16'
'DASYNI17'
'DB4PS'
'DBTS'
'DCMPDA'
'DCTPAH'
'DCTPD'
'DCTPD2'
'DCYTD'
'DCYTt2r'
'DDCat'
'DDGLK'
'DDPA'
'DDPGA'
'DEXTRAN40e'
'DEXTRINASE'
'DEXTRINabc'
'DGK1'
'DGSNt2'
'DHAD1'
'DHAD2'
'DHAK'
'DHDPRy'
'DHDPS'
'DHF0R2'
'DHFR'
'DHFS'
'DHNAOPT'
'DHNAOT4'
'DHNAOT7'
'DHNPA'
'DHNPA2'
'DHORD2'
'DHORD5'
'DHORDfum'
'DHORDi'
'DHORTS'
'DHPPDA'
'DHPPDA2'
'DHPR'
'DHPS'
'DHPS2'
'DHQS'
'DHQTi'
'DINSt2'
'DMATT'
'DMLARGAH'
'DMPPS'
'DMPPS2r'
'DMQMT'
'DMQMT2'
'DMSOR1'
'DMSOR2'
'DMSORdmq'
'DMSORmq'
'DM_2HYMEPH'
'DM_4HBA'
'DM_5DRIB'
'DM_5MTR'

'DM_AMOB'
'DM_adocbl(c)'
'DM_atp_c_'
'DM_btn'
'DM_clpn140(c)'
'DM_clpn160(c)'
'DM_clpn180(c)'
'DM_clpnai17(c)'
'DM_clpni16(c)'
'DM_clpni17(c)'
'DM_dad_5'
'DM_dhptd(c)'
'DM_hcys_L[c]'
'DM_kdo2lipid4L(c)'
'DM_pheme(c)'
'DM_teich_45_BS(c)'
'DPCOAK'
'DPGM'
'DPRr'
'DRBK'
'DRIBabc'
'DRPAr'
'DTMPK'
'DURAD'
'DURAD2'
'DURIK1'
'DURIPP'
'DURIt2'
'DUTPDP'
'DXPRIr'
'DXPS'
'D_LACD'
'D_LACt2'
'E4PD'
'EACPR1'
'EACPR2'
'EAR100x'
'EAR10M11x'
'EAR10M12'
'EAR10M12x'
'EAR11M12x'
'EAR120x'
'EAR121x'
'EAR12M13x'
'EAR12M14x'
'EAR13M14x'
'EAR140x'
'EAR14M15x'
'EAR14M16'
'EAR14M16x'
'EAR15M16'
'EAR15M16x'
'EAR160x'
'EAR161x'
'EAR180x'
'EAR181x'
'EAR40x'
'EAR4M5x'
'EAR4M6'
'EAR4M6x'
'EAR5M6'
'EAR5M6x'
'EAR60x'
'EAR6M7x'
'EAR6M8'
'EAR6M8x'
'EAR7M8'
'EAR7M8x'

'EAR80x'
'EAR8M10x'
'EAR8M9x'
'EAR9M10'
'EAR9M10x'
'ECOAH1'
'ECOAH12'
'ECOAH1R'
'ECOAH2'
'ECOAH3'
'ECOAH4'
'ECOAH5'
'ECOAH6'
'ECOAH7'
'ECOAH9'
'EDA_R'
'EDTXS1'
'EDTXS2'
'EHGLAT'
'ENO'
'ESHCYS2_FPe'
'ESHCYS_FPe'
'ESHGLU2_FPe'
'ESHGLU_FPe'
'ETHAAL'
'ETHAP'
'ETHS03abc'
'ET0Ht'
'EX_12dgr180(e)'
'EX_12ppd_S(e)'
'EX_26dap_M(e)'
'EX_2hyoxplac(e)'
'EX_2obut(e)'
'EX_2omfuc(e)'
'EX_2omxyl(e)'
'EX_3ddlhept(e)'
'EX_3mop(e)'
'EX_4hbz(e)'
'EX_7dhcdchol(e)'
'EX_7ocholate(e)'
'EX_C02528(e)'
'EX_Lcyst(e)'
'EX_Lcystin(e)'
'EX_Ser_Thr(e)'
'EX_T_antigen(e)'
'EX_Tn_antigen(e)'
'EX_Tyr_ggn(e)'
'EX_ac(e)'
'EX_acald(e)'
'EX_acerA(e)'
'EX_acgal(e)'
'EX_acgalglcur(e)'
'EX_acgalidour(e)'
'EX_acgalidour2s(e)'
'EX_acgam(e)'
'EX_acmana(e)'
'EX_acnam(e)'
'EX_ade(e)'
'EX_adn(e)'
'EX_adpcb1(e)'
'EX_ala_L(e)'
'EX_alaasp(e)'
'EX_alagln(e)'
'EX_alaglu(e)'
'EX_alagly(e)'
'EX_alahis(e)'
'EX_alaleu(e)'
'EX_alathr(e)'

'EX_amp(e)'
 'EX_amylopect900(e)'
 'EX_amylose300(e)'

micRea =

0	1	0	1	1	1	1	1	1	0	1	0	0
1	1	0	1	0	0	0	0	1	1	1	0	0
0	1	0	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	0	1	0	0
0	1	0	1	1	1	1	1	1	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	1	1	1	1	1	1	0	1	0	0
0	1	0	1	1	1	1	1	1	0	1	0	0
0	0	0	1	1	1	1	1	1	0	1	0	0

binOrg =

SRS011061 SRS012902 SRS011134 SRS054956

-----	-----	-----	-----
1	0	1	0
1	1	1	1
1	1	1	0
1	1	1	1
1	1	1	1
0	0	1	1
1	1	1	0
1	1	1	1
1	1	1	1
1	0	1	1
1	1	1	1
1	1	0	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	0	1
1	1	1	1
0	1	1	1
1	1	1	1
1	1	1	1
1	1	1	0
1	1	1	0
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	0	1	1

patOrg =

28 25 28 24

reacPat =

1301	0	1301	0
1302	1302	1302	1302
1314	1314	1314	0
858	858	858	858
1169	1169	1169	1169
0	0	1225	1225
949	949	949	0
1196	1196	1196	1196
1292	1292	1292	1292
1228	0	1228	1228

reacNumb =

1301
1302
1314
858
1169
1225
949
1196
1292
1228

reacSet =

'12DGR180ti'	'12DGR180ti'	'12DGR180ti'	'12DGR180ti'
'12PPDt'	'12PPDt'	'12PPDt'	'12PPDt'
'1P4H2CBXLAH'	'1P4H2CBXLAH'	'1P4H2CBXLAH'	'1P4H2CBXLAH'
'23DHMP0'	'23DHMP0'	'23DHMP0'	'23DHMP0'
'23PDE2'	'23PDE2'	'23PDE2'	'23PDE2'
'23PDE4'	'23PDE4'	'23PDE4'	'23PDE4'
'23PDE7'	'23PDE7'	'23PDE7'	'23PDE7'
'23PDE9'	'23PDE9'	'23PDE9'	'23PDE9'
'26DAPLLAT'	'26DAPLLAT'	'26DAPLLAT'	'26DAPLLAT'
'26DAPt2r'	'26DAPt2r'	'26DAPt2r'	'26DAPt2r'
'2AHBUTI'	'2AHBUTI'	'2AHBUTI'	'2AHBUTI'
'2DHPL'	'2DHPL'	'2DHPL'	'2DHPL'
'2HB0'	'2HB0'	'2HB0'	'2HB0'
'2HYOXPLACt2r'	'2HYOXPLACt2r'	'2HYOXPLACt2r'	'2HYOXPLACt2r'
'2INSD'	'2INSD'	'2INSD'	'2INSD'
'2MBCOATA'	'2MBCOATA'	'2MBCOATA'	'2MBCOATA'
'2MMALD'	'2MMALD'	'2MMALD'	'2MMALD'
'2MMALD2'	'2MMALD2'	'2MMALD2'	'2MMALD2'
'20BUTt2r'	'20BUTt2r'	'20BUTt2r'	'20BUTt2r'
'20MMBOX'	'20MMBOX'	'20MMBOX'	'20MMBOX'
'20MPHH'	'20MPHH'	'20MPHH'	'20MPHH'
'34DH0XPEG0X'	'34DH0XPEG0X'	'34DH0XPEG0X'	'34DH0XPEG0X'
'36DAHXI'	'36DAHXI'	'36DAHXI'	'36DAHXI'
'3ABUTCOAL'	'3ABUTCOAL'	'3ABUTCOAL'	'3ABUTCOAL'
'3CARLPDH'	'3CARLPDH'	'3CARLPDH'	'3HACPR1'
'3HACPR1'	'3HACPR1'	'3HACPR1'	'3HACPR2'
'3HACPR2'	'3HACPR2'	'3HACPR2'	'3HAD100'
'3HAD100'	'3HAD100'	'3HAD100'	'3HAD10M11'
'3HAD10M11'	'3HAD10M11'	'3HAD10M11'	'3HAD10M12'
'3HAD10M12'	'3HAD10M12'	'3HAD10M12'	'3HAD11M12'
'3HAD11M12'	'3HAD11M12'	'3HAD11M12'	'3HAD120'
'3HAD120'	'3HAD120'	'3HAD120'	'3HAD12M13'
'3HAD12M13'	'3HAD12M13'	'3HAD12M13'	'3HAD12M14'
'3HAD12M14'	'3HAD12M14'	'3HAD12M14'	'3HAD13M14'
'3HAD13M14'	'3HAD13M14'	'3HAD13M14'	'3HAD140'
'3HAD140'	'3HAD140'	'3HAD140'	'3HAD14M15'
'3HAD14M15'	'3HAD14M15'	'3HAD14M15'	'3HAD14M16'
'3HAD14M16'	'3HAD14M16'	'3HAD14M16'	'3HAD15M16'
'3HAD15M16'	'3HAD15M16'	'3HAD15M16'	'3HAD160'
'3HAD160'	'3HAD160'	'3HAD160'	'3HAD180'
'3HAD180'	'3HAD180'	'3HAD180'	'3HAD40'
'3HAD40'	'3HAD40'	'3HAD40'	'3HAD4M5'
'3HAD4M5'	'3HAD4M5'	'3HAD4M5'	'3HAD4M6'
'3HAD4M6'	'3HAD4M6'	'3HAD4M6'	'3HAD5M6'
'3HAD5M6'	'3HAD5M6'	'3HAD5M6'	'3HAD60'
'3HAD60'	'3HAD60'	'3HAD60'	'3HAD6M7'
'3HAD6M7'	'3HAD6M7'	'3HAD6M7'	'3HAD6M8'
'3HAD6M8'	'3HAD6M8'	'3HAD6M8'	'3HAD7M8'
'3HAD7M8'	'3HAD7M8'	'3HAD7M8'	'3HAD80'
'3HAD80'	'3HAD80'	'3HAD80'	'3HAD8M10'
'3HAD8M10'	'3HAD8M10'	'3HAD8M10'	'3HAD8M9'
'3HAD8M9'	'3HAD8M9'	'3HAD8M9'	'3HAD9M10'

'3HAD9M10'	'3HAD9M10'	'3HAD9M10'	'3HBCOAE'
'3HBCOAE'	'3HBCOAE'	'3HBCOAE'	'3MOBS'
'3MOBS'	'3MOBS'	'3MOBS'	'3MOPt2r'
'3MOPt2r'	'3MOPt2r'	'3MOPt2r'	'30AACPR1'
'30AACPR1'	'30AACPR1'	'30AACPR1'	'30AACPR2'
'30AACPR2'	'30AACPR2'	'30AACPR2'	'30AR100'
'30AR100'	'30AR100'	'30AR100'	'30AR10M11'
'30AR10M11'	'30AR10M11'	'30AR10M11'	'30AR10M12'
'30AR10M12'	'30AR10M12'	'30AR10M12'	'30AR11M12'
'30AR11M12'	'30AR11M12'	'30AR11M12'	'30AR120'
'30AR120'	'30AR120'	'30AR120'	'30AR12M13'
'30AR12M13'	'30AR12M13'	'30AR12M13'	'30AR12M14'
'30AR12M14'	'30AR12M14'	'30AR12M14'	'30AR13M14'
'30AR13M14'	'30AR13M14'	'30AR13M14'	'30AR140'
'30AR140'	'30AR140'	'30AR140'	'30AR14M15'
'30AR14M15'	'30AR14M15'	'30AR14M15'	'30AR14M16'
'30AR14M16'	'30AR14M16'	'30AR14M16'	'30AR15M16'
'30AR15M16'	'30AR15M16'	'30AR15M16'	'30AR160'
'30AR160'	'30AR160'	'30AR160'	'30AR180'
'30AR180'	'30AR180'	'30AR180'	'30AR40'
'30AR40'	'30AR40'	'30AR40'	'30AR4M5'
'30AR4M5'	'30AR4M5'	'30AR4M5'	'30AR4M6'
'30AR4M6'	'30AR4M6'	'30AR4M6'	'30AR5M6'
'30AR5M6'	'30AR5M6'	'30AR5M6'	'30AR60'
'30AR60'	'30AR60'	'30AR60'	'30AR6M7'
'30AR6M7'	'30AR6M7'	'30AR6M7'	'30AR6M8'
'30AR6M8'	'30AR6M8'	'30AR6M8'	'30AR7M8'
'30AR7M8'	'30AR7M8'	'30AR7M8'	'30AR80'
'30AR80'	'30AR80'	'30AR80'	'30AR8M10'
'30AR8M10'	'30AR8M10'	'30AR8M10'	'30AR8M9'
'30AR8M9'	'30AR8M9'	'30AR8M9'	'30AR9M10'
'30AR9M10'	'30AR9M10'	'30AR9M10'	'30AS100'
'30AS100'	'30AS100'	'30AS100'	'30AS10M11'
'30AS10M11'	'30AS10M11'	'30AS10M11'	'30AS10M12'
'30AS10M12'	'30AS10M12'	'30AS10M12'	'30AS11M12'
'30AS11M12'	'30AS11M12'	'30AS11M12'	'30AS120'
'30AS120'	'30AS120'	'30AS120'	'30AS121'
'30AS121'	'30AS121'	'30AS121'	'30AS12M13'
'30AS12M13'	'30AS12M13'	'30AS12M13'	'30AS12M14'
'30AS12M14'	'30AS12M14'	'30AS12M14'	'30AS13M14'
'30AS13M14'	'30AS13M14'	'30AS13M14'	'30AS140'
'30AS140'	'30AS140'	'30AS140'	'30AS141'
'30AS141'	'30AS141'	'30AS141'	'30AS160'
'30AS160'	'30AS160'	'30AS160'	'30AS161'
'30AS161'	'30AS161'	'30AS161'	'30AS180'
'30AS180'	'30AS180'	'30AS180'	'30AS181'
'30AS181'	'30AS181'	'30AS181'	'30AS2M3'

reacTab =

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

reacAbun =

'12DGR180ti'	[0.2243]	[0.2797]	[0.2249]	[0.2662]
'12PPDt'	[0.6338]	[0.5771]	[0.6576]	[0.4524]
'1P4H2CBXLAH'	[0.0159]	[0.0187]	[0.0164]	[0.0178]
'23DHMP0'	[0.9342]	[0.9760]	[0.9493]	[0.9475]
'23PDE2'	[0.6864]	[0.6492]	[0.7066]	[0.6991]

'23PDE4'	[0.6864]	[0.6492]	[0.7066]	[0.6991]
'23PDE7'	[0.6864]	[0.6492]	[0.6999]	[0.6991]
'23PDE9'	[0.6864]	[0.6492]	[0.6999]	[0.6991]
'26DAPLLAT'	[0.8949]	[0.9101]	[0.9083]	[0.9475]
'26DAPt2r'	[0.3113]	[0.5122]	[0.3107]	[0.2968]
'2AHBUTI'	[0.9342]	[0.9760]	[0.9427]	[0.9475]
'2DHPL'	[0.0159]	[0.0187]	[0.0164]	[0.0178]
'2HB0'	[0.0159]	[0.0187]	[0.0164]	[0.0178]
'2HYOXPLAct2r'	[0.1333]	[0.1445]	[0.1070]	[0.1696]
'2INSD'	[0.3417]	[0.4055]	[0.3154]	[0.4180]
'2MBCOATA'	[1.0000]	[1.0405]	[1.0085]	[1.1467]
'2MMALD'	[0.6719]	[0.2418]	[0.7767]	[0.4261]
'2MMALD2'	[0.3359]	[0.1209]	[0.3884]	[0.2131]
'20BUTt2r'	[0.6473]	[0.6299]	[0.6991]	[0.5098]
'20MMBOX'	[0.0658]	[0.0677]	[0.0658]	[0.1992]
'20MPHH'	[0.0658]	[0.0677]	[0.0658]	[0.1992]
'34DH0XPEG0X'	[0.0159]	[0.0187]	[0.0164]	[0.0178]
'36DAHXI'	[0.0658]	[0.0677]	[0.0658]	[0.1992]
'3ABUTCOAL'	[0.0159]	[0.0187]	[0.0159]	[0.0178]
'3CARLPDH'	[0.0393]	[0.0658]	[0.0343]	[0]
'3HACPR1'	[0.8278]	[0.8033]	[0.8439]	[0.8385]
'3HACPR2'	[0.8278]	[0.8033]	[0.8439]	[0.8385]
'3HAD100'	[1.0000]	[1.0405]	[1.0069]	[1.1467]
'3HAD10M11'	[0.9841]	[1.0218]	[0.9865]	[1.1289]
'3HAD10M12'	[1.0000]	[1.0405]	[1.0024]	[1.1467]
'3HAD11M12'	[1.0000]	[1.0405]	[1.0024]	[1.1467]
'3HAD120'	[1.0000]	[1.0405]	[1.0024]	[1.1467]
'3HAD12M13'	[0.9841]	[1.0218]	[0.9865]	[1.1289]
'3HAD12M14'	[1.0000]	[1.0405]	[1.0024]	[1.1467]
'3HAD13M14'	[1.0000]	[1.0405]	[1.0024]	[1.1467]
'3HAD140'	[1.0000]	[1.0405]	[1.0024]	[1.1467]
'3HAD14M15'	[0.9841]	[1.0218]	[0.9865]	[1.1280]
'3HAD14M16'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD15M16'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD160'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD180'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD40'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD4M5'	[0.9841]	[1.0218]	[0.9865]	[1.1280]
'3HAD4M6'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD5M6'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD60'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD6M7'	[0.9841]	[1.0218]	[0.9865]	[1.1280]
'3HAD6M8'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD7M8'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD80'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD8M10'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HAD8M9'	[0.9841]	[1.0218]	[0.9865]	[1.1280]
'3HAD9M10'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'3HBCOAE'	[0.0328]	[0.0410]	[0.0328]	[0.0432]
'3MOBS'	[0.9342]	[0.9728]	[0.9365]	[0.9466]
'3MOPt2r'	[0.0159]	[0.0187]	[0.0159]	[0.0178]
'30AACPR1'	[0.8278]	[0.8033]	[0.8384]	[0.8376]
'30AACPR2'	[0.8278]	[0.8033]	[0.8384]	[0.8376]
'30AR100'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR10M11'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR10M12'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR11M12'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR120'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR12M13'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR12M14'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR13M14'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR140'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR14M15'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR14M16'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR15M16'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR160'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR180'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR40'	[1.0000]	[1.0405]	[1.0024]	[1.1458]

'30AR4M5'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR4M6'	[1.0000]	[1.0405]	[1.0024]	[1.1458]
'30AR5M6'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR60'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR6M7'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR6M8'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR7M8'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR80'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR8M10'	[1.0000]	[1.0400]	[1.0024]	[1.1458]
'30AR8M9'	[1.0000]	[1.0400]	[1.0000]	[1.1458]
'30AR9M10'	[1.0000]	[1.0391]	[1.0000]	[1.1458]
'30AS100'	[1.0000]	[1.0391]	[1.0000]	[1.1458]
'30AS10M11'	[1.0000]	[1.0391]	[1.0000]	[1.1458]
'30AS10M12'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS11M12'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS120'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS121'	[0.0159]	[0.0178]	[0.0159]	[0.0178]
'30AS12M13'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS12M14'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS13M14'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS140'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS141'	[0.0159]	[0.0178]	[0.0159]	[0.0178]
'30AS160'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS161'	[0.0159]	[0.0178]	[0.0159]	[0.0178]
'30AS180'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS181'	[0.0159]	[0.0178]	[0.0159]	[0.0178]
'30AS2M3'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS3M4'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS4M306'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS4M5'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS4M6'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS5M6'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS60'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS6M7'	[1.0000]	[1.0357]	[1.0000]	[1.1458]
'30AS6M8'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'30AS7M8'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'30AS80'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'30AS8M10'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'30AS8M9'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'30AS9M10'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'4HBZt2'	[0.5662]	[0.6845]	[0.5745]	[1.0620]
'4HTHRS'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'4NTPPH2'	[0.9533]	[0.9537]	[0.9583]	[1.1332]
'5AIZCDC'	[1.0000]	[1.0291]	[1.0000]	[1.1458]
'5DGLCNR'	[0.0939]	[0.1321]	[0.1041]	[0.1597]
'5DOAN'	[0.2636]	[0.3578]	[0.2697]	[0.2793]
'5GLUP5T1'	[0.0393]	[0.0658]	[0.0343]	[0]
'5GLUP5T2'	[0.0393]	[0.0658]	[0.0343]	[0]
'6PHBG'	[0.5602]	[0.3730]	[0.6081]	[0.4604]
'6PHBG2'	[0.0253]	[0.0306]	[0.0253]	[0.0306]
'7AHS DH'	[0.0357]	[0.0316]	[0.0357]	[0.0539]
'7AHS DHe'	[0.0179]	[0.0158]	[0.0179]	[0.0269]
'A5PISO'	[0.9831]	[1.0044]	[0.9831]	[1.1204]
'AACOAT'	[0.1667]	[0.2847]	[0.1667]	[0.2170]
'AACPS6'	[0.3113]	[0.5085]	[0.3080]	[0.2968]
'AACPS7'	[0.9342]	[0.9590]	[0.9342]	[0.9466]
'AADSACYCL'	[0.8333]	[0.7420]	[0.8333]	[0.9288]
'AB6PGH'	[0.0507]	[0.0611]	[0.0507]	[0.0611]
'AB6PGH2'	[0.0253]	[0.0306]	[0.0253]	[0.0306]
'ABTOXy'	[0.2565]	[0.3220]	[0.2532]	[0.3239]
'ABUTD'	[0.0159]	[0.0178]	[0.0159]	[0.0178]
'ACACT1r'	[0.0328]	[0.0400]	[0.0328]	[0.0432]

reacNumber =

2189

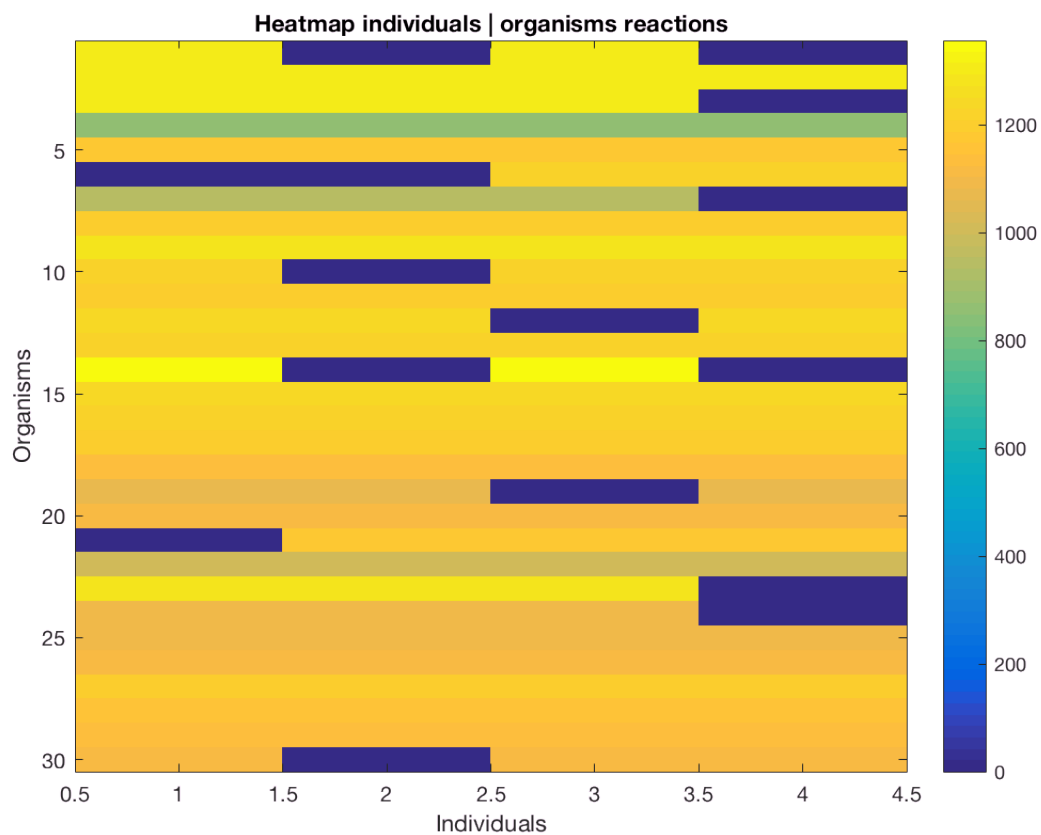
2175

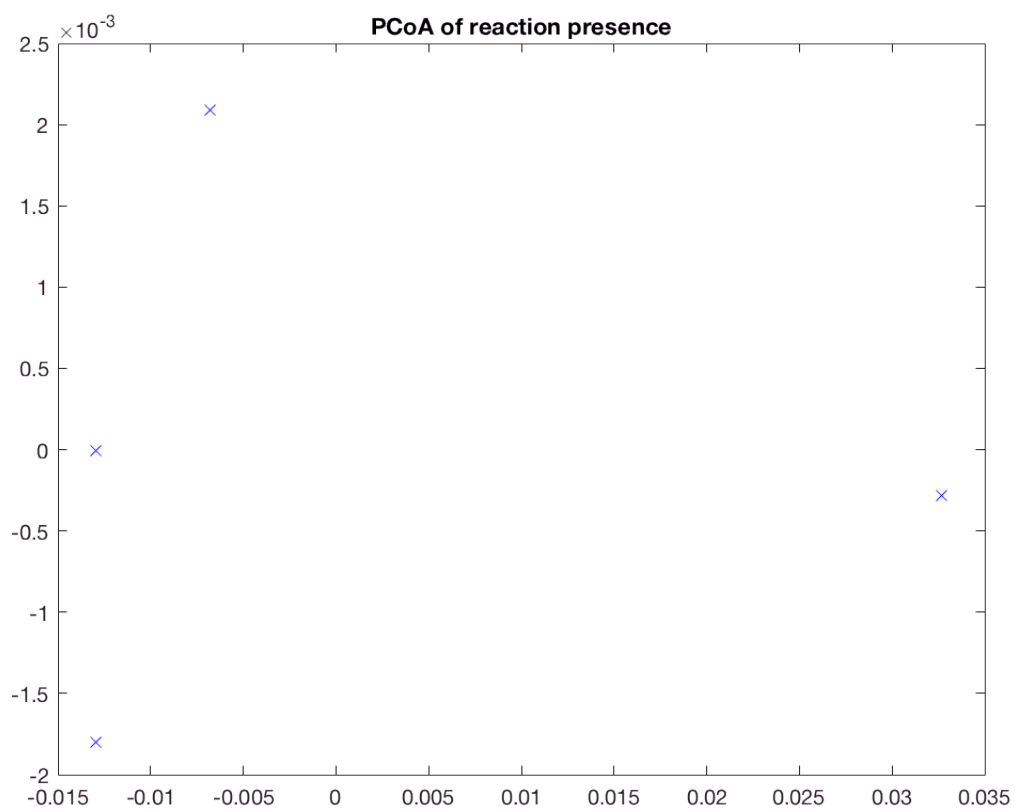
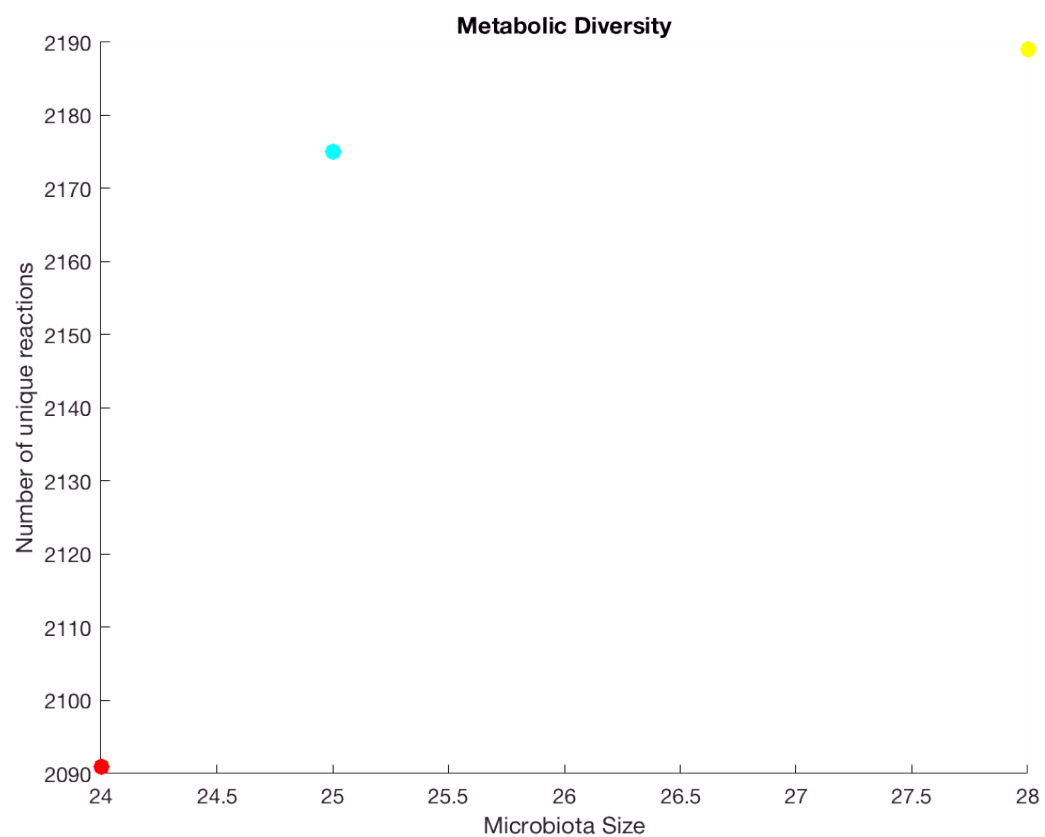
2189

2091

PCoA =

```
-0.0130    -0.0000
-0.0068     0.0021
-0.0129    -0.0018
 0.0327    -0.0003
```





%end of trigger for Autoload

PIPELINE: [PART 2.1]

Checking consistence of inputs: if autofix == 0 halts execution with error msg if inconsistencies are detected, otherwise it really tries hard to fix the problem and continues execution when possible.

```
[autoStat,fixVec,straains]=checkNomenConsist(straains,autoFix);
```

[illegible]

Now we detect from the content of the results folder if PART2 was already computed: if the associated file is already present in the results folder its execution is skipped else its execution starts

```
[mapP]=detectOutput(resPath,'Setup_allbacs.mat');

if isempty(mapP)
    modbuild = 1;
else
    modbuild = 0;
    s = 'global setup file found: loading from resPath and skipping [PART2.1] analysis';
    disp(s)
end
```

```
global setup file found: loading from resPath and skipping [PART2.1] analysis
```

```
%end of trigger for Autoload
```

A ?global? model joining all the reconstructions contained in the model will be created in this section. This model will be later used, integrating abundances coming from the metagenomic sequencing,

to derive the different microbiota models. The result of this section will be automatically saved in the results folder.

```
if modbuild == 1
    setup=fastSetupCreator(models, strains, {})
    setup.name='Global reconstruction with lumen / fecal compartments no host'
    setup.recon=0
    save(strcat(resPath, 'Setup_allbacs.mat'), 'setup')
end

if modbuild==0
    load(strcat(resPath, 'Setup_allbacs.mat'))
end
```

PIPELINE: [PART 2.2]

Now we will create the different microbiota models integrating the given abundances. Coupling constraints and personalized "cumulative biomass" objective functions are also added. Models that are already existent will not be recreated, and new microbiota models will be saved in the results folder.

```
[createdModels]=createPersonalizedModel(abunFilePath,resPath,setup,sampName,straains,patNumb)
```

```
mgmodel =
    c: [36426x1 double]
    citations: {36426x1 cell}
    comments: {36426x1 cell}
    confidenceScores: {36426x1 cell}
    ecNumbers: {36426x1 cell}
    grRules: {36426x1 cell}
    lb: [36426x1 double]
    rev: [36426x1 double]
    rules: {36426x1 cell}
    rxnKeggID: {36426x1 cell}
    rxnNames: {36426x1 cell}
    rxns: {36426x1 cell}
    subSystems: {36426x1 cell}
    ub: [36426x1 double]
    rxnConfidenceScores: [36426x1 double]
    rxnECNumbers: {36426x1 cell}
    rxnKEGGID: {36426x1 cell}
    S: [32054x36426 double]
    b: [32054x1 double]
    metCHEBIID: {32054x1 cell}
    metCharge: [32054x1 double]
    metFormulas: {32054x1 cell}
    metHMDB: {32054x1 cell}
    metInchiString: {32054x1 cell}
    metKeggID: {32054x1 cell}
    metNames: {32054x1 cell}
    metPubChemID: {32054x1 cell}
    metSmile: {32054x1 cell}
    mets: {32054x1 cell}
    csense: [32054x1 char]
    metChEBIID: {32054x1 cell}
    metCharges: [32054x1 double]
    methMDBID: {32054x1 cell}
    metInChIString: {32054x1 cell}
    metKEGGID: {32054x1 cell}
    metSmiles: {32054x1 cell}
    genes: {}
    name: 'Global reconstruction with lumen / fecal compartments no host'
    recon: 0
```

```
id =
  'SRS011061'

createdModels =
  []
  'SRS011061'

mapP = 8
microbiota model file found: skipping model creation for this sample
```

```
mgmodel =
      c: [36426x1 double]
      citations: {36426x1 cell}
      comments: {36426x1 cell}
      confidenceScores: {36426x1 cell}
      ecNumbers: {36426x1 cell}
      grRules: {36426x1 cell}
      lb: [36426x1 double]
      rev: [36426x1 double]
      rules: {36426x1 cell}
      rxnKeggID: {36426x1 cell}
      rxnNames: {36426x1 cell}
      rxns: {36426x1 cell}
      subSystems: {36426x1 cell}
      ub: [36426x1 double]
      rxnConfidenceScores: [36426x1 double]
      rxnECNumbers: {36426x1 cell}
      rxnKEGGID: {36426x1 cell}
      S: [32054x36426 double]
      b: [32054x1 double]
      metCHEBIID: {32054x1 cell}
      metCharge: [32054x1 double]
      metFormulas: {32054x1 cell}
      methMDB: {32054x1 cell}
      metInchiString: {32054x1 cell}
      metKeggID: {32054x1 cell}
      metNames: {32054x1 cell}
      metPubChemID: {32054x1 cell}
      metSmile: {32054x1 cell}
      mets: {32054x1 cell}
      csense: [32054x1 char]
      metChEBIID: {32054x1 cell}
      metCharges: [32054x1 double]
      methMDBID: {32054x1 cell}
      metInChIString: {32054x1 cell}
      metKEGGID: {32054x1 cell}
      metSmiles: {32054x1 cell}
      genes: {}
      name: 'Global reconstruction with lumen / fecal compartments no host'
      recon: 0
```

```
id =
  'SRS012902'

createdModels =
  []
  'SRS011061'
  'SRS012902'

mapP = 10
microbiota model file found: skipping model creation for this sample
```

```
mgmodel =
      c: [36426x1 double]
      citations: {36426x1 cell}
      comments: {36426x1 cell}
```

```
confidenceScores: {36426x1 cell}
ecNumbers: {36426x1 cell}
grRules: {36426x1 cell}
lb: [36426x1 double]
rev: [36426x1 double]
rules: {36426x1 cell}
rxnKeggID: {36426x1 cell}
rxnNames: {36426x1 cell}
rxns: {36426x1 cell}
subSystems: {36426x1 cell}
ub: [36426x1 double]
rxnConfidenceScores: [36426x1 double]
rxnECNumbers: {36426x1 cell}
rxnKEGGID: {36426x1 cell}
S: [32054x36426 double]
b: [32054x1 double]
metCHEBIID: {32054x1 cell}
metCharge: [32054x1 double]
metFormulas: {32054x1 cell}
metHMDB: {32054x1 cell}
metInchiString: {32054x1 cell}
metKeggID: {32054x1 cell}
metNames: {32054x1 cell}
metPubChemID: {32054x1 cell}
metSmile: {32054x1 cell}
mets: {32054x1 cell}
csense: [32054x1 char]
metChEBIID: {32054x1 cell}
metCharges: [32054x1 double]
metHMDBID: {32054x1 cell}
metInChIString: {32054x1 cell}
metKEGGID: {32054x1 cell}
metSmiles: {32054x1 cell}
genes: {}
name: 'Global reconstruction with lumen / fecal compartments no host'
recon: 0
```

```
id =
    'SRS011134'
```

```
createdModels =
    []
    'SRS011061'
    'SRS012902'
    'SRS011134'
```

```
mapP = 9
```

```
microbiota model file found: skipping model creation for this sample
```

```
mgmodel =
    c: [36426x1 double]
    citations: {36426x1 cell}
    comments: {36426x1 cell}
    confidenceScores: {36426x1 cell}
    ecNumbers: {36426x1 cell}
    grRules: {36426x1 cell}
    lb: [36426x1 double]
    rev: [36426x1 double]
    rules: {36426x1 cell}
    rxnKeggID: {36426x1 cell}
    rxnNames: {36426x1 cell}
    rxns: {36426x1 cell}
    subSystems: {36426x1 cell}
    ub: [36426x1 double]
    rxnConfidenceScores: [36426x1 double]
    rxnECNumbers: {36426x1 cell}
    rxnKEGGID: {36426x1 cell}
```

```

        S: [32054x36426 double]
        b: [32054x1 double]
        metCHEBIID: {32054x1 cell}
        metCharge: [32054x1 double]
        metFormulas: {32054x1 cell}
        metHMDB: {32054x1 cell}
        metInchiString: {32054x1 cell}
        metKeggID: {32054x1 cell}
        metNames: {32054x1 cell}
        metPubChemID: {32054x1 cell}
        metSmile: {32054x1 cell}
        mets: {32054x1 cell}
        csense: [32054x1 char]
        metChEBIID: {32054x1 cell}
        metCharges: [32054x1 double]
        metHMDBID: {32054x1 cell}
        metInChIString: {32054x1 cell}
        metKEGGID: {32054x1 cell}
        metSmiles: {32054x1 cell}
        genes: {}
        name: 'Global reconstruction with lumen / fecal compartments no host'
        recon: 0

```

```

id =
    'SRS054956'

```

```

createdModels =
    []
    'SRS011061'
    'SRS012902'
    'SRS011134'
    'SRS054956'

```

```

mapP = 11

```

```

microbiota model file found: skipping model creation for this sample

```

```

createdModels =
    []
    'SRS011061'
    'SRS012902'
    'SRS011134'
    'SRS054956'

```

PIPELINE: [PART 3]

Add some description of what the computed outputs are and what is located in which file-Almut

In this phase, for each microbiota model, a diet, in the form of set constraints to the exchanges reactions of the diet compartment, is integrated. Flux Variability analysis for all the exchange reactions of the diet and fecal compartment is also computed and saved.

```

[ID,fvaCt,nsCt,presol,inFesMat]=microbiotaModelSimulator(resPath,setup,sampName,dietFilePath,r

```

```

mapP = 397

```

```

simulations already done, file found: loading from resPath

```

```

ID =
    'EX_12dgr180[fe]'
    'EX_12ppd_S[fe]'
    'EX_26dap_M[fe]'
    'EX_2hyoxplac[fe]'
    'EX_2obut[fe]'
    'EX_2omfuc[fe]'

```

'EX_2omxyl[fe]'
'EX_3ddlhept[fe]'
'EX_3mop[fe]'
'EX_4hbz[fe]'
'EX_7dhcdchol[fe]'
'EX_7ocholate[fe]'
'EX_C02528[fe]'
'EX_Lcyst[fe]'
'EX_Lcystin[fe]'
'EX_Ser_Thr[fe]'
'EX_T_antigen[fe]'
'EX_Tn_antigen[fe]'
'EX_Tyr_ggn[fe]'
'EX_ac[fe]'
'EX_acald[fe]'
'EX_acerA[fe]'
'EX_acgal[fe]'
'EX_acgalglcur[fe]'
'EX_acgalidour2s[fe]'
'EX_acgalidour[fe]'
'EX_acgam[fe]'
'EX_acmana[fe]'
'EX_acnam[fe]'
'EX_ade[fe]'
'EX_adn[fe]'
'EX_adocbl[fe]'
'EX_ala_L[fe]'
'EX_alaasp[fe]'
'EX_alagln[fe]'
'EX_alaglu[fe]'
'EX_alagly[fe]'
'EX_alahis[fe]'
'EX_alaleu[fe]'
'EX_alathr[fe]'
'EX_amp[fe]'
'EX_amylopect900[fe]'
'EX_amylose300[fe]'
'EX_apio_D[fe]'
'EX_arab_L[fe]'
'EX_arabinan101[fe]'
'EX_arabinogal[fe]'
'EX_arabinoxyl[fe]'
'EX_arabttr[fe]'
'EX_arbt[fe]'
'EX_arg_L[fe]'
'EX_arsenb[fe]'
'EX_asn_L[fe]'
'EX_asp_L[fe]'
'EX_bglc[fe]'
'EX_btn[fe]'
'EX_but[fe]'
'EX_butso3[fe]'
'EX_ca2[fe]'
'EX_cbl1[fe]'
'EX_cbl2[fe]'
'EX_cd2[fe]'
'EX_cellb[fe]'
'EX_cgly[fe]'
'EX_chol[fe]'
'EX_cholate[fe]'
'EX_chor[fe]'
'EX_chtbs[fe]'
'EX_cit[fe]'
'EX_cl[fe]'
'EX_co2[fe]'
'EX_cobalt2[fe]'
'EX_core2[fe]'
'EX_core3[fe]'

'EX_core4[fe]'
'EX_core5[fe]'
'EX_core6[fe]'
'EX_core7[fe]'
'EX_core8[fe]'
'EX_cro4[fe]'
'EX_csn[fe]'
'EX_cspg_a[fe]'
'EX_cspg_a_degr[fe]'
'EX_cspg_ab_rest[fe]'
'EX_cspg_b[fe]'
'EX_cspg_b_degr[fe]'
'EX_cspg_c[fe]'
'EX_cspg_c_degr[fe]'
'EX_cspg_c_rest[fe]'
'EX_ctbt[fe]'
'EX_cu2[fe]'
'EX_cys_L[fe]'
'EX_cytd[fe]'
'EX_dad_2[fe]'
'EX_dcyt[fe]'
'EX_ddca[fe]'
'EX_dextran40[fe]'
'EX_dextrin[fe]'
'EX_dgchol[fe]'
'EX_dgsn[fe]'
'EX_din[fe]'
'EX_dms[fe]'
'EX_dms0[fe]'
'EX_drib[fe]'
'EX_dsT_antigen[fe]'
'EX_duri[fe]'
'EX_ethso3[fe]'
'EX_etoh[fe]'
'EX_f1a[fe]'
'EX_fe2[fe]'
'EX_fe3[fe]'
'EX_fecrm[fe]'
'EX_fol[fe]'
'EX_for[fe]'
'EX_fru[fe]'
'EX_fuc_L[fe]'
'EX_fum[fe]'
'EX_gal[fe]'
'EX_galmannan[fe]'
'EX_galt[fe]'
'EX_galur[fe]'
'EX_gam26s[fe]'
'EX_gam[fe]'
'EX_gbbtn[fe]'
'EX_gcald[fe]'
'EX_gchola[fe]'
'EX_glc_D[fe]'
'EX_glcmannan[fe]'
'EX_glcn[fe]'
'EX_glcur[fe]'
'EX_gln_L[fe]'
'EX_glu_L[fe]'
'EX_gly[fe]'
'EX_glyasn[fe]'
'EX_glyasp[fe]'
'EX_glyb[fe]'
'EX_glyc3p[fe]'
'EX_glyc[fe]'
'EX_glyclt[fe]'
'EX_glycys[fe]'
'EX_glygln[fe]'
'EX_glyglu[fe]'

'EX_glygn2[fe]'
'EX_glygn4[fe]'
'EX_glygn5[fe]'
'EX_glyleu[fe]'
'EX_glymet[fe]'
'EX_glyphe[fe]'
'EX_glypro[fe]'
'EX_glytyr[fe]'
'EX_gncore1[fe]'
'EX_gncore2[fe]'
'EX_gsn[fe]'
'EX_gthox[fe]'
'EX_gthrd[fe]'
'EX_gua[fe]'
'EX_h2[fe]'
'EX_h2o[fe]'
'EX_h2s[fe]'
'EX_h[fe]'
'EX_ha[fe]'
'EX_ha_deg1[fe]'
'EX_ha_pre1[fe]'
'EX_hdca[fe]'
'EX_hexs[fe]'
'EX_hg2[fe]'
'EX_his_L[fe]'
'EX_homogal[fe]'
'EX_hspg[fe]'
'EX_hspg_degr_10[fe]'
'EX_hspg_degr_11[fe]'
'EX_hspg_degr_12[fe]'
'EX_hspg_degr_13[fe]'
'EX_hspg_degr_14[fe]'
'EX_hspg_degr_15[fe]'
'EX_hspg_degr_1[fe]'
'EX_hspg_degr_2[fe]'
'EX_hspg_degr_3[fe]'
'EX_hspg_degr_4[fe]'
'EX_hspg_degr_5[fe]'
'EX_hspg_degr_6[fe]'
'EX_hspg_degr_7[fe]'
'EX_hspg_degr_8[fe]'
'EX_hspg_degr_9[fe]'
'EX_hspg_rest[fe]'
'EX_hxan[fe]'
'EX_idour[fe]'
'EX_ile_L[fe]'
'EX_ind3ac[fe]'
'EX_indole[fe]'
'EX_ins[fe]'
'EX_inulin[fe]'
'EX_isetac[fe]'
'EX_isobut[fe]'
'EX_isocapr[fe]'
'EX_isoal[fe]'
'EX_k[fe]'
'EX_kdo[fe]'
'EX_kesto[fe]'
'EX_kestopt[fe]'
'EX_kestottr[fe]'
'EX_lac_D[fe]'
'EX_lac_L[fe]'
'EX_lanost[fe]'
'EX_lcts[fe]'
'EX_leu_L[fe]'
'EX_levan1000[fe]'
'EX_levanb[fe]'
'EX_levanttr[fe]'
'EX_levantttr[fe]'

'EX_lichn[fe]'
'EX_lmn2[fe]'
'EX_lmn30[fe]'
'EX_lys_L[fe]'
'EX_mal_L[fe]'
'EX_malt[fe]'
'EX_malthx[fe]'
'EX_malttr[fe]'
'EX_man[fe]'
'EX_mannan[fe]'
'EX_mantr[fe]'
'EX_melib[fe]'
'EX_meoh[fe]'
'EX_met_D[fe]'
'EX_met_L[fe]'
'EX_metala[fe]'
'EX_metsox_R_L[fe]'
'EX_metsox_S_L[fe]'
'EX_mg2[fe]'
'EX_mn2[fe]'
'EX_mnl[fe]'
'EX_mops[fe]'
'EX_mqn7[fe]'
'EX_mqn8[fe]'
'EX_mso3[fe]'
'EX_nal[fe]'
'EX_nac[fe]'
'EX_ncam[fe]'
'EX_nh4[fe]'
'EX_nmn[fe]'
'EX_no2[fe]'
'EX_no3[fe]'
'EX_o2[fe]'
'EX_ocdca[fe]'
'EX_ocdcea[fe]'
'EX_orn[fe]'
'EX_pac[fe]'
'EX_pb[fe]'
'EX_pect[fe]'
'EX_pecticgal[fe]'
'EX_phe_L[fe]'
'EX_pheme[fe]'
'EX_pi[fe]'
'EX_pnto_R[fe]'
'EX_ppa[fe]'
'EX_ppi[fe]'
'EX_pro_L[fe]'
'EX_ptrc[fe]'
'EX_pullulan1200[fe]'
'EX_pydam[fe]'
'EX_pydx[fe]'
'EX_pydxn[fe]'
'EX_q8[fe]'
'EX_raffin[fe]'
'EX_rbflvrd[fe]'
'EX_rhamnogalurII[fe]'
'EX_rhamnogalurI[fe]'
'EX_rib_D[fe]'
'EX_ribflv[fe]'
'EX_rmn[fe]'
'EX_sT_antigen[fe]'
'EX_sTn_antigen[fe]'
'EX_salcn[fe]'
'EX_sbt_D[fe]'
'EX_ser_L[fe]'
'EX_sheme[fe]'
'EX_so3[fe]'
'EX_so4[fe]'

```

'EX_spm�[fe]'
'EX_strch1[fe]'
'EX_strch2[fe]'
'EX_stys[fe]'
'EX_succ[fe]'
'EX_sucr[fe]'
'EX_sulfac[fe]'
'EX_taur[fe]'
'EX_tchola[fe]'
'EX_tdchola[fe]'
'EX_thm[fe]'
'EX_thr_L[fe]'
'EX_thymd[fe]'
'EX_tma[fe]'
'EX_tmao[fe]'
'EX_tre[fe]'
'EX_trp_L[fe]'
'EX_tsuł[fe]'
'EX_ttdca[fe]'
'EX_tyr_L[fe]'
'EX_ura[fe]'
'EX_urea[fe]'
'EX_uri[fe]'
'EX_val_L[fe]'
'EX_xan[fe]'
'EX_xyl_D[fe]'
'EX_xylan[fe]'
'EX_xyluglc[fe]'
'EX_zn2[fe]'

```

```

fvaCt =
      []      []      []      []
{307×3 cell} {307×3 cell} {307×3 cell} {307×3 cell}
      []      []      []      []

```

```

nsCt =
      []      []      []      []
{307×3 cell} {307×3 cell} {307×3 cell} {307×3 cell}
      []      []      []      []

```

```

presol =
      []      []
[1]      [0.3601]
[1]      [0.2902]
[1]      [0.3725]
[1]      [0.3844]

```

```

inFesMat =

0×0 empty cell array

```

Finally, NMPCs (net maximal production capability) are computed in a metabolite resolved manner and saved in a comma delimited file in the results folder. NMPCs indicate the maximal production of each metabolite and are computing summing the maximal secretion flux with the maximal uptake flux. Similarity of metabolic profiles (using the different NMPCs as features) between individuals are also evaluated with classical multidimensional scaling.

```
[Fsp,Y]= mgSimResCollect(resPath,ID,rDiet,0,patNumb,fvaCt,figForm)
```

```

init = 2
fl = 2
names =

```


	'rich'	'standard'	'personalized'
noPcoa = 0			
noPcoa = 0			
noPcoa = 0			
noPcoa = 0			
P =			

	1.0e+04 *
2.9570	0.0001
1.9523	0.0001
0.7309	0.0000
-0.0000	-0.0000

Fsp =

	0	0	0	0
0	0	0	0	0
44.8687	35.5302	11.6037	0.9619	
0	0	0	0	
93.3584	77.7395	48.7473	78.4956	
0	0	0	0	
0	0	0	0	
0	0	0	0	
2.2933	0.0144	28.2133	6.7012	
0.0001	0.0000	0.0003	0.0000	

Y =

-98.6158	-70.4971	-34.8909
-63.6848	48.9105	59.8644
117.3366	-66.4325	20.6509
44.9640	88.0191	-45.6243

