FluxPyt- a Python-based free and open-source software for ^{13}C-metabolic flux analyses

Trunil S Desai [1, 2] , **Shireesh Srivastava** [Corresp. 1, 2]

[1] Systems Biology for Biofuels Group, International Centre for Genetic Engineering and Biotechnology, New Delhi, Delhi, India

[2] DBT-ICGEB Center for Advanced Bioenergy Research, International Centre for Genetic Engineering and Biotechnology, New Delhi, Delhi, India

Corresponding Author: Shireesh Srivastava
Email address: shireesh@icgeb.res.in

^{13}C-Metabolic Flux Analysis (MFA) is a powerful approach to estimate intracellular reaction rates which could be used in strain analysis and design. Processing and analysis of labeling data for calculation of fluxes and associated statistics is an essential part of MFA. However, various software currently available for data analysis employ proprietary platforms and thus limit accessibility. We developed FluxPyt, a python-based truly open source software package for conducting stationary ^{13}C-MFA data analysis. The software is based on the efficient Elementary Metabolite Unit (EMU) framework. The standard deviations in the calculated fluxes are estimated using the Monte-Carlo analysis. FluxPyt also automatically creates flux maps based on a template for visualization of the MFA results. The flux distributions for two separate models: a small tricarboxylic acid cycle model and a larger *Corynebacterium glutamicum* model, calculated by FluxPyt are in good agreement with those calculated by a previously-published software. FluxPyt was tested in Microsoft™ Windows 7 and 10, as well as in Linux Mint 18.2. The availability of a free and open ^{13}C-MFA software that works in various operating systems will enable more researchers to perform ^{13}C-MFA and to further modify and develop the package.

1 **FluxPyt- a Python-based free and open-source software for ^{13}C-metabolic flux analyses**

3 Trunil S. Desai[1,2] and Shireesh Srivastava[1,2]

5 [1]Systems Biology for Biofuels Group, International Centre for Genetic Engineering and
6 Biotechnology (ICGEB), New Delhi, India

7 [2]DBT-ICGEB Center for Advanced Bioenergy Research, New Delhi, India

9 Corresponding Author

10 Shireesh Srivastava

11 Ph: +91-26741358/61 Extn. 450

12 Email: shireesh@icgeb.res.in

Abstract

[13]C-Metabolic Flux Analysis (MFA) is a powerful approach to estimate intracellular reaction rates which could be used in strain analysis and design. Processing and analysis of labeling data for calculation of fluxes and associated statistics is an essential part of MFA. However, various software currently available for data analysis employ proprietary platforms and thus limit accessibility. We developed FluxPyt, a python-based truly open source software package for conducting stationary ^{13}C-MFA data analysis. The software is based on the efficient Elementary Metabolite Unit (EMU) framework. The standard deviations in the calculated fluxes are estimated using the Monte-Carlo analysis. FluxPyt also automatically creates flux maps based on a template for visualization of the MFA results. The flux distributions for two separate models: a small tricarboxylic acid cycle model and a larger *Corynebacterium glutamicum* model, calculated by FluxPyt are in good agreement with those calculated by a previously-published software. FluxPyt was tested in Microsoft[TM] Windows 7 and 10, as well as in Linux Mint 18.2. The availability of a free and open ^{13}C-MFA software that works in various operating systems will enable more researchers to perform ^{13}C-MFA and to further modify and develop the package.

Introduction

Metabolic flux analysis (MFA) is an important component of metabolic engineering (Stephanopoulos, 1999). MFA measures the distribution of flux in the different metabolic pathways of an organism as well as identifies the intracellular flux changes associated with altered cellular response such as overproduction of a desired metabolite for biotechnological applications or understanding the altered pathways in diseases. The flux values from MFA can

36   be used to constrain the solution space of the linear optimization problem in Flux Balance

37   Analysis (FBA) to give a physiologically meaningful flux distribution (Kim & Reed, 2012; Desai

38   & Srivastava, 2015).

39   MFA methodology has evolved over years. Earlier studies employed an approach termed

40   stoichiometric MFA which balanced the fluxes around the intracellular metabolites in a

41   stoichiometric network (vanGulik et al., 2001; Srivastava & Chan, 2008). While it provided

42   important insights into altered metabolism under different conditions, there are several

43   limitations of stoichiometric MFA such as the inability to determine fluxes in reversible or

44   parallel reactions (Wiechert et al., 2001). ^{13}C-MFA, which employs ^{13}C-labeled substrates can

45   resolve fluxes through such reactions. For a typical ^{13}C-MFA, the cells are fed with a labeled

46   substrate or mixture of labeled substrates, labeled at one more carbon atoms. The cells are

47   required to be at metabolically steady state, i.e. when the intracellular concentrations of

48   metabolites do not change over time (Zamboni et al., 2009). This ensures that the differences in

49   the labeling patterns are not due to temporal variations in the fluxes, but due to the differential

50   activity of the pathways. Once the labeled substrate attains steady state distribution throughout

51   the metabolic network (isotopic steady state), the relative abundances of the mass isotopomers of

52   intracellular metabolites and proteinogenic amino acids are measured using NMR (Szyperski et

53   al., 1999) or mass spectrometry to derive their respective mass isotopomer distributions (MIDs).

54   The gas chromatography/mass spectrometry (GCMS) technique for measurement of MIDs of

55   *tert*-butyldimethylsilyl (TBDMS) derivatives of amino acids is well established. The

56   fragmentation pattern produced by the derivatives is also well characterized and these ions are

57   used to calculate the MIDs which are eventually used to calculate the fluxes (Antoniewicz,

58   Kelleher & Stephanopoulos, 2007b; Leighty & Antoniewicz, 2013; Crown, Long &

59 Antoniewicz, 2015; Gonzalez, Long & Antoniewicz, 2017). Zamboni et al. provide a detailed

60 protocol for ^{13}C-MFA (Zamboni et al., 2009). A suitable metabolic model and the MID data are

61 then fed in software tools which estimate the flux distributions in reactions included in the MFA

62 model.

63 Along with the development of experimental techniques for stationary MFA, the methodology to

64 calculate flux distribution from the labeling data has also evolved over the years. An important

65 development in this direction was the introduction of the Elementary metabolite unit (EMU)

66 framework (Antoniewicz, Kelleher & Stephanopoulos, 2007a). The EMU decomposition

67 algorithm detects minimum number of isotopomer balances needed to simulate the observed

68 labeling pattern which greatly reduces the number of variables without any loss of information,

69 thus making it much more efficient than the isotopomer mapping matrices approach (Schmidt et

70 al., 1997). Based on the measured fractional enrichments of metabolites, the metabolic network

71 is decomposed into elementary reactions of the EMUs. These elementary reactions form the

72 basis of system of equations representing the relation between metabolic fluxes and the observed

73 isotopic labeling. Recent MFA software such as WUFlux (He et al., 2016), Metran

74 (Antoniewicz, Kelleher & Stephanopoulos, 2007a) and OpenFLUX (Quek et al., 2009) employ

75 the elementary metabolite units (EMU) framework. Most software for ^{13}C-MFA such as

76 OpenFLUX (Quek et al., 2009), WUFlux (He et al., 2016), INCA (Young, 2014) and Metran

77 (Antoniewicz, Kelleher & Stephanopoulos, 2007a) work either on the proprietary MATLAB

78 platform or are not free and open source. Many also use the MATLAB Statistics and

79 Optimization toolboxes which are proprietary. Thus, a truly free and open-source software is not

80 available yet. Understanding the requirement for open source tools, Birkel et al. have recently

81 developed jQMM, an open source Python library to perform ^{13}C-MFA which uses GAMS

82   (General Algebraic Modeling System) to solve the optimization problem (Birkel et al., 2017).

83   However, GAMS is a proprietary system for mathematical optimization. We believe that a truly

84   open source software in which all the components and packages are free and open source would

85   make MFA more accessible to researchers and also encourage further development of ^{13}C-MFA

86   and related tools.

87   **Material and Methods**

88   FluxPyt is written in Python (version 3.6.1) because Python is an open source platform and has

89   free, well-developed data handling packages with ability to perform the various tasks in ^{13}C-

90   MFA. In FluxPyt, the numerical and matrix operations are conducted using the packages NumPy

91   and SciPy (Walt, Colbert & Varoquaux, 2011), and SymPy (Meurer et al., 2017). The non-linear

92   optimization (see equation 3 below) is conducted using the package lmfit (Newville et al., 2014).

93   Python was installed as Anaconda3 (version 3.4.4.0) distribution. Anaconda3 also acts as a

94   package manager to handle the dependency and compatibility issues between the different

95   packages necessary for the functioning of FluxPyt. FluxPyt was developed and tested using the

96   Spyder integrated development environment (IDE) with the IPython console. FluxPyt was tested

97   in Windows 7, Windows 10 and Linux Mint (18.2) operating systems. Detailed information of

98   packages in the Anaconda distribution is provided in the Data S1. While FluxPyt is designed to

99   be used by people with limited programming skills, the open source nature allows further

100 modification by advanced users. The general framework of FluxPyt is shown in Figure 1. Table

101 1 describes different modules present in FluxPyt. The MFA model, isotope correction data and

102 substrate input data are written in a csv file which can be easily edited in a spreadsheet. The

103 reactions are divided into different types, namely, irreversible ("F"), reversible forward ("FR"),

104 reversible reverse ("R"), metabolite balancing reactions ("B") and isotopomer balancing

105 reactions ("S"). The metabolites which do not take part in isotopomer balancing are marked as

106 "X" in the atom transition equations. The basis flux, which typically is the substrate uptake flux

107 in a network, is marked with a "*". The upper bounds of other reactions are limited to 15 times

108 this basis flux. The reverse free fluxes cannot be determined as they have no natural upper limit

109 (Quek et al., 2009). The substrate information and the natural correction data are read from

110 separate csv files which must be named substrate_input.csv and corr_file.csv, respectively. The

111 correction file also specifies the number of mass isotopomer measurements for every amino acid

112 provided for fitting. This is used to extract the MID vectors of length equal to the length of MID

113 vectors provided in the measurement data. For detailed information on creating an MFA model

114 for FluxPyt, a user guide is provided in the supplemental material (Data S2).

115 The model is automatically parsed into metabolite and isotopomer models. The metabolite model

116 is a mass balance model that satisfies the metabolic steady state condition,

$$S \cdot v = 0 \tag{1}$$

117 where, S is the stoichiometric matrix and v is the flux distribution vector. For a given

118 stoichiometric matrix, the flux distribution vector can be written as a function of free fluxes (

119 v_{free}) as follows:

$$v = N \cdot v_{free} \tag{2}$$

120 where N is the null space of stoichiometric matrix.

121 The isotopomer balance model is generated based on the EMU decomposition algorithm

122 (Antoniewicz, Kelleher & Stephanopoulos, 2007a). It assumes an isotopic steady state and is

123 used to solve the non-linear optimization problem using the free fluxes as the optimization

124 parameters to minimize the sum of squared residues (SSR) as given in equation 3.

$$minimize\left[\frac{(MID_{calc} - MID_{measured})^2}{error^2}\right] \qquad (3)$$

125 Levenberg-Marquardt algorithm implemented in the lmfit package is used to optimize the free

126 flux parameters in the metabolite model. The free flux assignments are done based on the

127 reduced row echelon form (rref) of the stoichiometric matrix. Before calculating the rref, the

128 reactions are arranged in following order:

129
$$\begin{bmatrix} \overrightarrow{v_{rev}} \\ v_{irrev} \\ \overrightarrow{v_{free}} \\ v_{basis} \\ \overleftarrow{v_{rev}} \end{bmatrix}$$

130 $\overrightarrow{v_{rev}}$ and $\overleftarrow{v_{rev}}$ are the forward and reverse reactions of reversible reactions; v_{irrev} are irreversible

131 reactions; $\overrightarrow{v_{free}}$ are reactions that the user chooses to act as free fluxes; v_{basis} are reactions with

132 measured or predetermined values. The arrangement of reactions in a specified order ensures that

133 the measured fluxes are calculated as free fluxes. In addition, the user may provide a preference

134 for certain reactions to be calculated as free fluxes ($\overrightarrow{v_{free}}$). The default number of iterations is 10,

135 but this value can be changed by the user. The iterations are run with different random initial

136 parameters so as to increase the chances of finding global minimum.

137 Flux standard deviations are calculated using the Monte-Carlo method. Multiple datasets of the

138 measured MID values are generated using the mean and standard deviations of the respective

139 MIDs. The default value for the number of datasets is 500, but can be changed by the user. Each

140 dataset of the measured MID values is then used to solve the non-linear optimization problem

141   (He et al., 2016). This results in multiple optimal solutions (500 solutions by default) for each

142   reaction. From these values the 68% and 95% confidence intervals are determined from the

143   upper and lower 16% and 2.5% values, respectively, using the bootstrap method (He et al., 2016)

144   as follows. One thousand samples of 500 values each (if the default number of samples for

145   Monte Carlo analysis is used) are used to calculate the required percentiles (2.5, 16, 50, 84 and

146   97.5 percentiles). The calculated percentiles from each sample are stored as a separate set. The

147   median of each percentile value is reported as the upper or lower boundary of the respective

148   confidence interval. For example, the median 2.5 and 97.5 percentile values represent the lower

149   and upper boundaries of the 95% confidence interval and the median 16 and 84 percentile values

150   represent the lower and upper boundaries of the 68% confidence interval, respectively. The

151   median of the 50 percentile values represents the median of the estimated fluxes.

152   **Goodness of fit**

153   Assuming that the model is correct and the data are without gross errors, the sum of squares

154   residual (SSR) has a χ^2 distribution with number of degrees of freedom equal to number of

155   measurements to be fitted (n) minus the number of independent parameters (p) (Antoniewicz,

156   Kelleher & Stephanopoulos, 2006; Leighty & Antoniewicz, 2013). The acceptable range of SSR

157   is between $\chi^2_{\alpha/2}(n - p)$ and $\chi^2_{1 - \alpha/2}(n - p)$, where α is the chosen significance level, for example,

158   α=0.05 for 95% confidence interval.

159   **Output files and visualization of the results**

160   FluxPyt generates several result files during the MFA, a list of which is provided in Table S1.

161   The data results, containing the calculated MIDs, the optimal flux distribution values and

162   confidence intervals are stored as separate .csv and .pckl files. FluxPyt provides graphical

163   representation of results. The differences between the measured and the calculated MIDs of the

164   best fit solution are automatically plotted. Flux map is automatically generated using a template

165   flux diagram created using Inkscape. The template for flux map is a svg file that can be created

166   using Inkscape. Templates for tricarboxylic acid cycle (TCA) and *C. glutamicum* networks are

167   provided in the additional data (Data S1) or can be downloaded from the FluxPyt project site.

168   The user can easily modify the template based on their model of interest using the same drawing

169   tool to make flux maps for their organism of interest. The users need to write the ID of the

170   reactions at the appropriate position in the template where they want the flux value of that

171   particular reaction to be printed (Data S1). The bootstrap results are plotted as box plots. The

172   user can choose the specific reactions to be plotted. Additionally, the bootstrap output is stored

173   as a pandas dataframe (McKinney, 2010).

174   **Results**

175   We employed the models and the MID data given in OpenFLUX (Quek et al., 2009) for initial

176   evaluation of FluxPyt. Both the TCA cycle and the *Corynebacterium glutamicum* central

177   metabolic network (Data S1) were compared.

178   **Preliminary validation with the TCA cycle model**

179   The TCA cycle metabolic network (see Data S1 for a detailed description of the model as well as

180   a list of metabolites) consisted of 18 metabolite reactions and 3 isotopomer balancing reactions

181   (type "S" reactions). The network has two substrate input reactions, pyruvate (V01) and

182   glutamate (V02) and one reversible reaction (represented as two separate reactions, V07 and

183   V08). Three of the reactions (V16, V17 and V18) have measured values relative to pyruvate flux

184   fixed at 1. The stoichiometric matrix has 7 free variables. Rearranging the reactions as mentioned

185   in the implementation section above enables the measured flux values to be calculated as free

186   fluxes. The three measured free flux parameters were allowed to vary within three standard

187   deviations of their means while the remaining 3 free flux parameters were allowed to vary

188   between the maximum bounds as explained in the implementation section. The only remaining

189   parameter was the flux through pyruvate uptake reaction which was fixed to 1. 18 data points

190   were fitted to 6 flux parameters, therefore the maximum acceptable SSR was 21. The number of

191   iterations was set to 10. Flux analysis results from both software converged to a similar solution.

192   The MIDs calculated by FluxPyt were the same as those calculated by OpenFLUX (Table 2).

193   Very minor differences in the values of two fluxes were observed (Table 3). These could be due

194   to the different solvers and algorithms used to solve the least squares fitting problem and linear

195   system of equations (He et al., 2016). The flux map, which was automatically generated (see

196   methods for details) based on the template, is shown in Figure 2. The Monte Carlo analysis

197   suggested tight confidence intervals for the TCA model (Figure 3). The reversible reaction

198   spanned the entire flux range, as expected. The optimal values for all fluxes lay within 2.5 and

199   97.5 percentile range, i.e., the 95% CI (Table 4).

200   **MFA of *C. glutamicum* metabolic network**

201   The *C. glutamicum* model (Quek et al., 2009) consisted of 67 reactions, 42 metabolites and 25

202   free fluxes (see Data S1 for a detailed description of the model as well as a list of metabolites).

203   The flux measurements for 18 reactions were included in the model. The 24 variable parameters

204   were fit to the 29 data points of amino acid labeling pattern. Here too, the calculated fluxes were

205   similar to those calculated by OpenFLUX (Table S2).

206 The SSR values for the measured vs. calculated MIDs were similar for FluxPyt (690) and

207 OpenFLUX (701) (Table 5). However, the SSR values for both the software were higher than the

208 maximum acceptable SSR value (11.07), suggesting that either the model used was incorrect or

209 there were measurement errors. Use of a recent MID data (Shupletsov et al., 2014), reduced the

210 SSR to 10.33 (Table 6), which is within the acceptable range (1.15 to 11.07). This showed that

211 the MID data previously used had measurement errors. The Figure 4 shows the flux map for the

212 *C. glutamicum* model for this data (see Table S3) for the list of flux values). The CIs calculated

213 with Monte Carlo method are shown in Table S4.

214 Our results show that FluxPyt can be used with for performing MFA with real metabolic models.

215 It is capable of flux estimation when multiple labeled substrates are used (e.g., the TCA model

216 uses three labeled substrates, namely 1-^{13}C pyruvate, U-^{13}C pyruvate and 1-^{13}C Glutamate).

217 **Discussion**

218 MFA is a powerful method to measure intracellular fluxes. However, all of the currently used

219 software have some component that is proprietary and thus makes it impossible to conduct ^{13}C-

220 MFA without obtaining/ purchasing a license. Most licenses do not allow altering the code,

221 restricting further development. In designing this open source software, our effort was to keep

222 the interface simple and easy to use for scientists somewhat familiar with other MFA software.

223 Therefore, the files for labeling-data preparation have been kept similar to the popular software

224 OpenFLUX. The software can be run through different interfaces. The Anaconda command

225 prompt may be convenient for people familiar with Python, while the Spyder IDE would appear

226 more familiar to scientists familiar with MATLAB interface. The iPython console in Spyder

227 offers a more interactive environment than the command prompt. A print of FluxPyt for

228   performing MFA of TCA network running in the Spyder console is provided in supplemental file

229   (Data S3).

230   The fluxes calculated by MFA are dependent on the network chosen and the measurements used.

231   In case of the *C. glutamicum* metabolic network, it was found that the MID data used by (Quek

232   et al., 2009) had measurement errors giving unacceptably high SSR for the MIDs calculated. Use

233   of a more recent MID data (Shupletsov et al., 2014) significantly reduced the SSR to acceptable

234   values.

235   At this point the software is released for use in Microsoft Windows and Linux operating systems

236   (OS). Similarly, the confidence interval calculations currently use the Monte-Carlo method. The

237   open source nature makes it possible for advanced users to adapt it to other OS as well as add

238   other methods for calculating confidence intervals.

239

240   **Conclusion**

241   FluxPyt is free and open source software for conducting stationary ^{13}C-MFAanalyses with

242   performance comparable to other available software. It can be used for flux estimation with

243   experimental data of practical scale. Automatic generation of flux maps allows better

244   visualization of results. This tool will be useful to researchers interested in stationary ^{13}C-MFA

245   studies.

246

247   **Additional information**

248   The source code can be downloaded from the repository

249   (https://sourceforge.net/projects/fluxpyt/)

250   **Operating system:** Windows 10, Windows 7 and Linux.

251   **Programming language:** e.g. Python (~3.6).

252   **Other requirements:** e.g. NumPy, SciPy, SymPy, lmfit, csvkit, glpk, pandas.

253   **License:** BSD 3-clause

254   **Any restrictions to use by non-academics:** None

255

256

257

258   **References**

259   Antoniewicz MR., Kelleher JK., Stephanopoulos G. 2006. Determination of confidence intervals

260   of metabolic fluxes estimated from stable isotope measurements. *Metabolic Engineering*

261   8:324–337. DOI: 10.1016/j.ymben.2006.01.004.

262   Antoniewicz MR., Kelleher JK., Stephanopoulos G. 2007a. Elementary metabolite units (EMU):

263   a novel framework for modeling isotopic distributions. *Metabolic Engineering* 9:68–86.

264   DOI: 10.1016/j.ymben.2006.09.001.

265   Antoniewicz MR., Kelleher JK., Stephanopoulos G. 2007b. Accurate assessment of amino acid

266   mass isotopomer distributions for metabolic flux analysis. *Analytical Chemistry* 79:7554–

267   7559. DOI: 10.1021/ac0708893.

268   Birkel GW., Ghosh A., Kumar VS., Weaver D., Ando D., Backman TWH., Arkin AP., Keasling

269   JD., Martín HG. 2017. The JBEI quantitative metabolic modeling library (jQMM): a

270   python library for modeling microbial metabolism. *BMC bioinformatics* 18:205. DOI:

271   10.1186/s12859-017-1615-y.

272   Crown SB., Long CP., Antoniewicz MR. 2015. Integrated 13C-metabolic flux analysis of 14

273   parallel labeling experiments in Escherichia coli. *Metabolic Engineering* 28:151–158.

274   DOI: 10.1016/j.ymben.2015.01.001.

275   Desai T., Srivastava S. 2015. Constraints-Based Modeling to Identify Gene Targets for

276   Overproduction of Ethanol by Escherichia coli: The Effect of Glucose Phosphorylation

277   Reaction. *Metabolomics:Open Access* 5:1000145. DOI: 10.4172/2153-0769.1000145.

278   Gonzalez JE., Long CP., Antoniewicz MR. 2017. Comprehensive analysis of glucose and xylose

279   metabolism in Escherichia coli under aerobic and anaerobic conditions by (13)C

280   metabolic flux analysis. *Metabolic Engineering* 39:9–18. DOI:

281   10.1016/j.ymben.2016.11.003.

282   He L., Wu SG., Zhang M., Chen Y., Tang YJ. 2016. WUFlux: an open-source platform for (13)C

283   metabolic flux analysis of bacterial metabolism. *BMC bioinformatics* 17:444. DOI:

284   10.1186/s12859-016-1314-0.

285   Kim J., Reed JL. 2012. RELATCH: relative optimality in metabolic networks explains robust

286   metabolic and regulatory responses to perturbations. *Genome Biology* 13:R78. DOI:

287   10.1186/gb-2012-13-9-r78.

288   Leighty RW., Antoniewicz MR. 2013. COMPLETE-MFA: complementary parallel labeling

289   experiments technique for metabolic flux analysis. *Metabolic Engineering* 20:49–55.

290   DOI: 10.1016/j.ymben.2013.08.006.

291   McKinney W. 2010. Data Structures for Statistical Computing in Python. In: Walt S van der,

292   Millman J eds. *Proceedings of the 9th Python in Science Conference*. 51–56.

293   Meurer A., Smith CP., Paprocki M., Čertík O., Kirpichev SB., Rocklin M., Kumar Am., Ivanov

294   S., Moore JK., Singh S., Rathnayake T., Vig S., Granger BE., Muller RP., Bonazzi F.,

295   Gupta H., Vats S., Johansson F., Pedregosa F., Curry MJ., Terrel AR., Roučka Š., Saboo

296   A., Fernando I., Kulal S., Cimrman R., Scopatz A. 2017. SymPy: symbolic computing in

297   Python. *PeerJ Computer Science* 3:e103. DOI: 10.7717/peerj-cs.103.

298   Newville M., Stensitzki T., Allen DB., Ingargiola A. 2014. *LMFIT: Non-Linear Least-Square*

299   *Minimization and Curve-Fitting for Python¶*. Zenodo.

300   Quek L-E., Wittmann C., Nielsen LK., Krömer JO. 2009. OpenFLUX: efficient modelling

301   software for 13C-based metabolic flux analysis. *Microbial Cell Factories* 8:25. DOI:

302   10.1186/1475-2859-8-25.

303   Schmidt K., Carlsen M., Nielsen J., Villadsen J. 1997. Modeling isotopomer distributions in

304   biochemical networks using isotopomer mapping matrices. *Biotechnology and*

305   *Bioengineering* 55:831–840. DOI: 10.1002/(SICI)1097-0290(19970920)55:6<831::AID-

306   BIT2>3.0.CO;2-H.

307   Shupletsov MS., Golubeva LI., Rubina SS., Podvyaznikov DA., Iwatani S., Mashko SV. 2014.

308   OpenFLUX2: 13C-MFA modeling software package adjusted for the comprehensive

309   analysis of single and parallel labeling experiments. *Microbial Cell Factories* 13:152.

310   DOI: 10.1186/s12934-014-0152-x.

311   Srivastava S., Chan C. 2008. Application of metabolic flux analysis to identify the mechanisms

312   of free fatty acid toxicity to human hepatoma cell line. *Biotechnology and Bioengineering*

313   99:399–410. DOI: 10.1002/bit.21568.

314   Stephanopoulos G. 1999. Metabolic fluxes and metabolic engineering. *Metabolic Engineering*

315   1:1–11. DOI: 10.1006/mben.1998.0101.

316   Szyperski T., Glaser RW., Hochuli M., Fiaux J., Sauer U., Bailey JE., Wüthrich K. 1999.

317   Bioreaction network topology and metabolic flux ratio analysis by biosynthetic fractional

318   13C labeling and two-dimensional NMR spectroscopy. *Metabolic Engineering* 1:189–

319   197. DOI: 10.1006/mben.1999.0116.

320   vanGulik WM., Antoniewicz MR., deLaat WT., Vinke JL., Heijnen JJ. 2001. Energetics of

321   growth and penicillin production in a high-producing strain of Penicillium chrysogenum.

322   *Biotechnology and Bioengineering* 72:185–193.

323   Walt S van der., Colbert SC., Varoquaux G. 2011. The NumPy Array: A Structure for Efficient

324   Numerical Computation. *Computing in Science Engineering* 13:22–30. DOI:

325   10.1109/MCSE.2011.37.

326   Wiechert W., Möllney M., Petersen S., de Graaf AA. 2001. A universal framework for 13C

327   metabolic flux analysis. *Metabolic Engineering* 3:265–283. DOI:

328   10.1006/mben.2001.0188.

329   Young JD. 2014. INCA: a computational platform for isotopically non-stationary metabolic flux

330   analysis. *Bioinformatics (Oxford, England)* 30:1333–1335. DOI:

331   10.1093/bioinformatics/btu015.

332   Zamboni N., Fendt S-M., Rühl M., Sauer U. 2009. (13)C-based metabolic flux analysis. *Nature*

333   *Protocols* 4:878–892. DOI: 10.1038/nprot.2009.58.

334

Figure 1

The general framework of FluxPyt.

The user of the software provides the model definition and the mass isotopomer distribution (MID) data. The metabolite model is then calculated. The flux parameter optimization is then performed using the Levenberg-Marquardt algorithm to calculate the MIDs and the best-fit fluxes. The differences in the calculated vs. the measured MIDs are plotted. The flux maps of the calculated flux distribution is automatically generated based on the provided svg file.

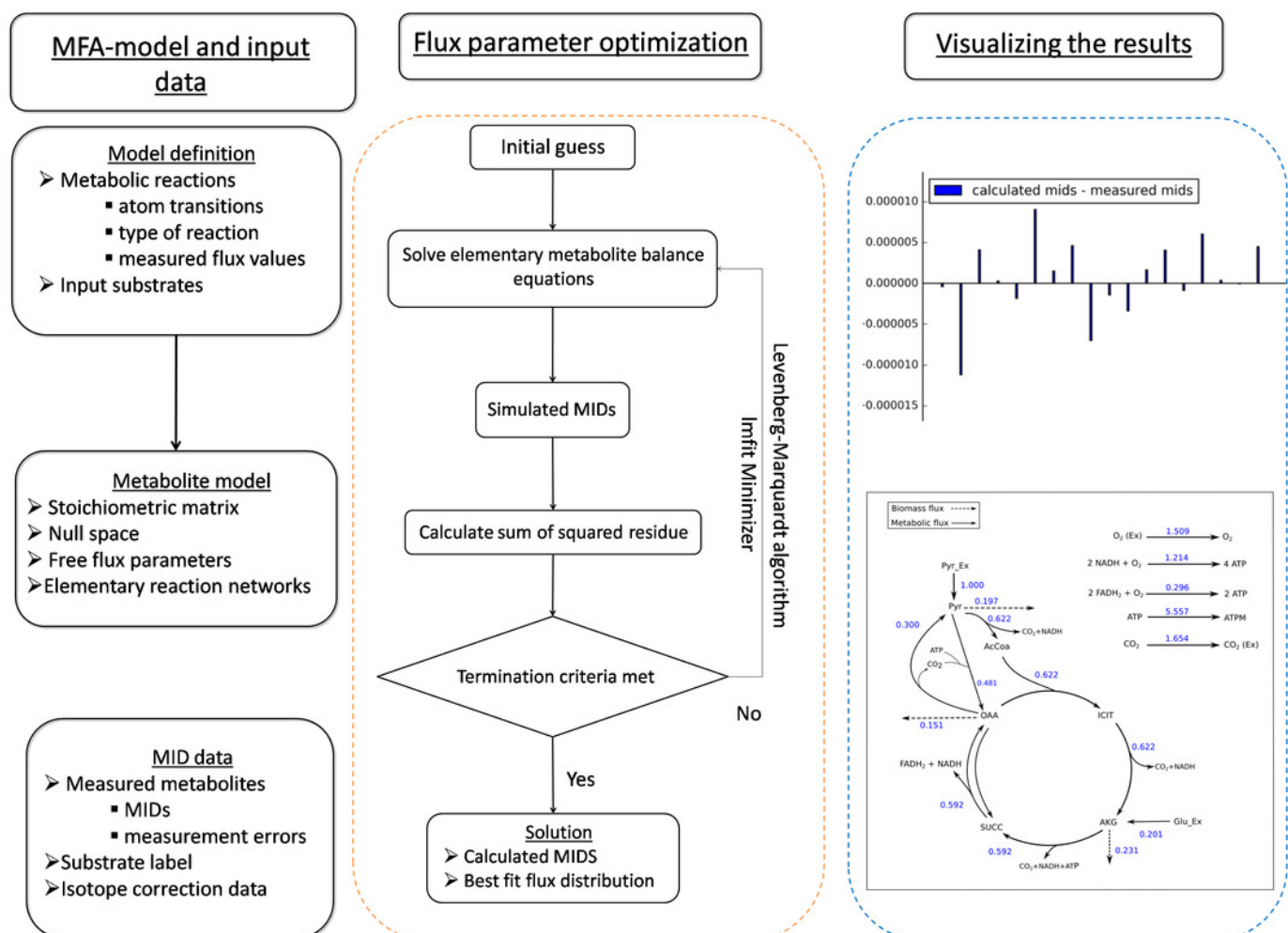



Figure 2

Flux map of the TCA generated by FluxPyt.

The fluxes for the reactions of the TCA cycle model are shown as generated by the FluxPyt.
Net flux values are shown for reversible reactions.

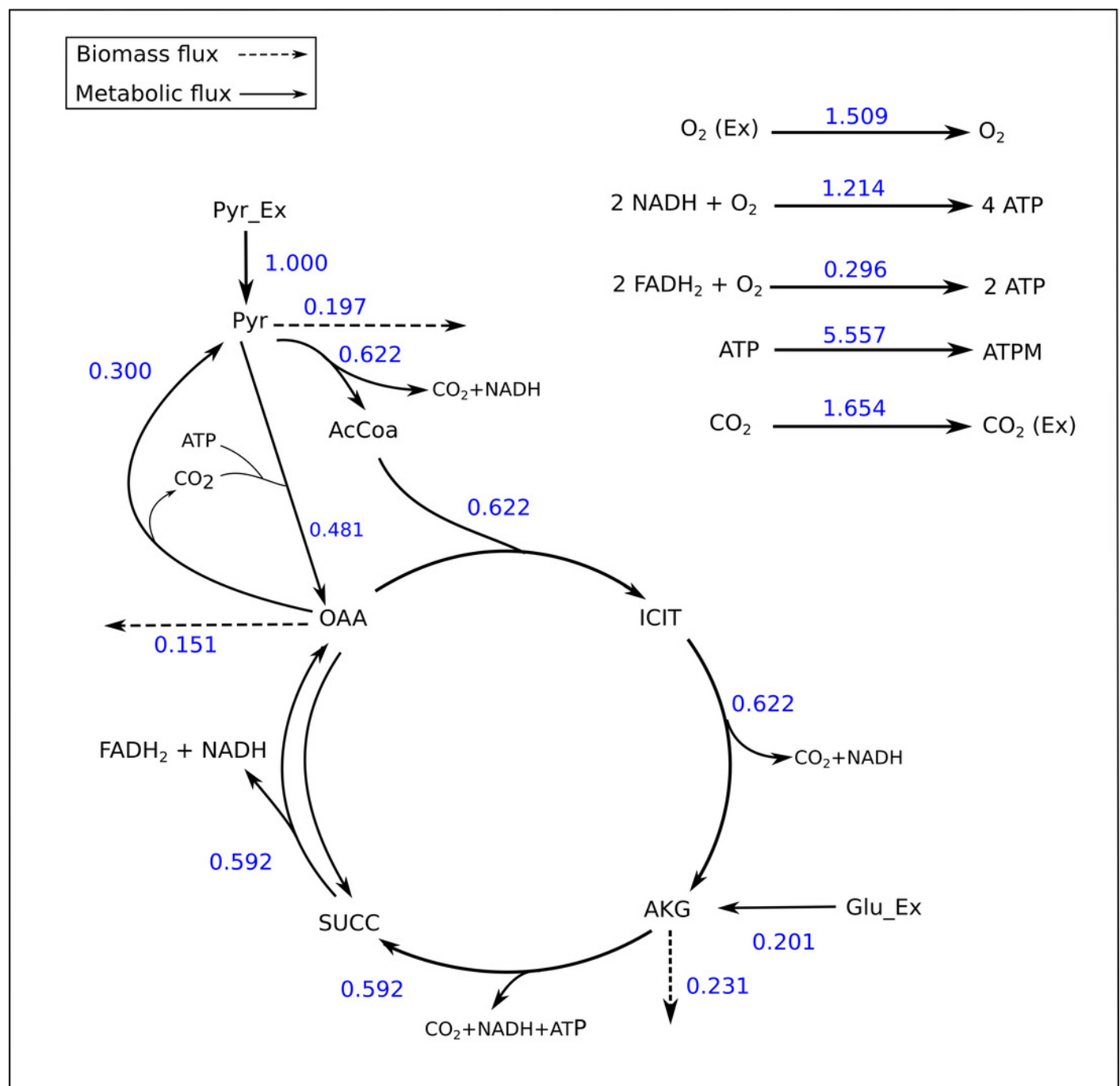



Figure 3

Flux confidence intervals (CIs) calculated by Monte-Carlo method for selected reactions.

The boxes span 16 and 84 percentiles and whiskers span 2.5 and 97.5 percentiles. See the TCA model in Data S1 for full form of the reactions. (A) The flux confidence intervals for the reactions V02, V06 and V09, (B) The flux confidence intervals for the reactions V08, V11, V13 and V14.

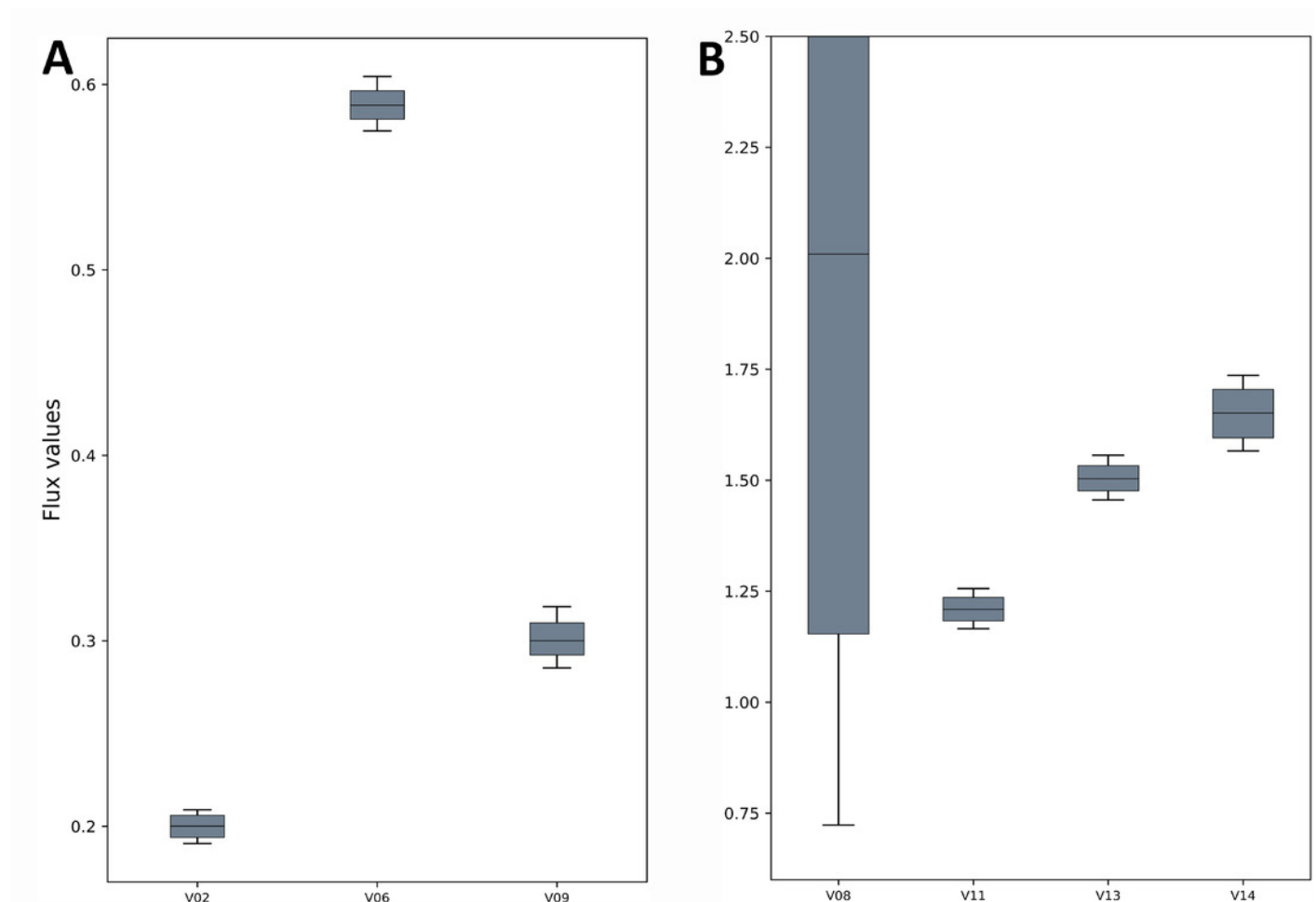



Figure 4

A map of the *C. glutamicum* fluxes generated by FluxPyt.

The fluxes of the metabolic network of the C. glutamicum calculated using the data from (Shupletsov et al., 2014) are shown on the metabolic network. The figure is generated automatically by the software.

CO_2 315.258   → CO_2 (Ex)

Glc_Ex

| Biomass flux | ------> |
| Metabolic flux | ——> |

TREM ← 1.520 — G6P —— CO_2 48.252

106.000

54.704

F6P   14.643   S7P   14.643   → P5P -------->

79.511

F16BP   E4P   G3P

79.511

DHAP ← → G3P   P5P   F6P

79.511   12.438

170.869

3PG

159.913

LAC ← 0.086 — Pyr --------->

29.570   89.648

CO_2+NADH

AcCoa

CO_2 34.527   65.457

OAA   CIT

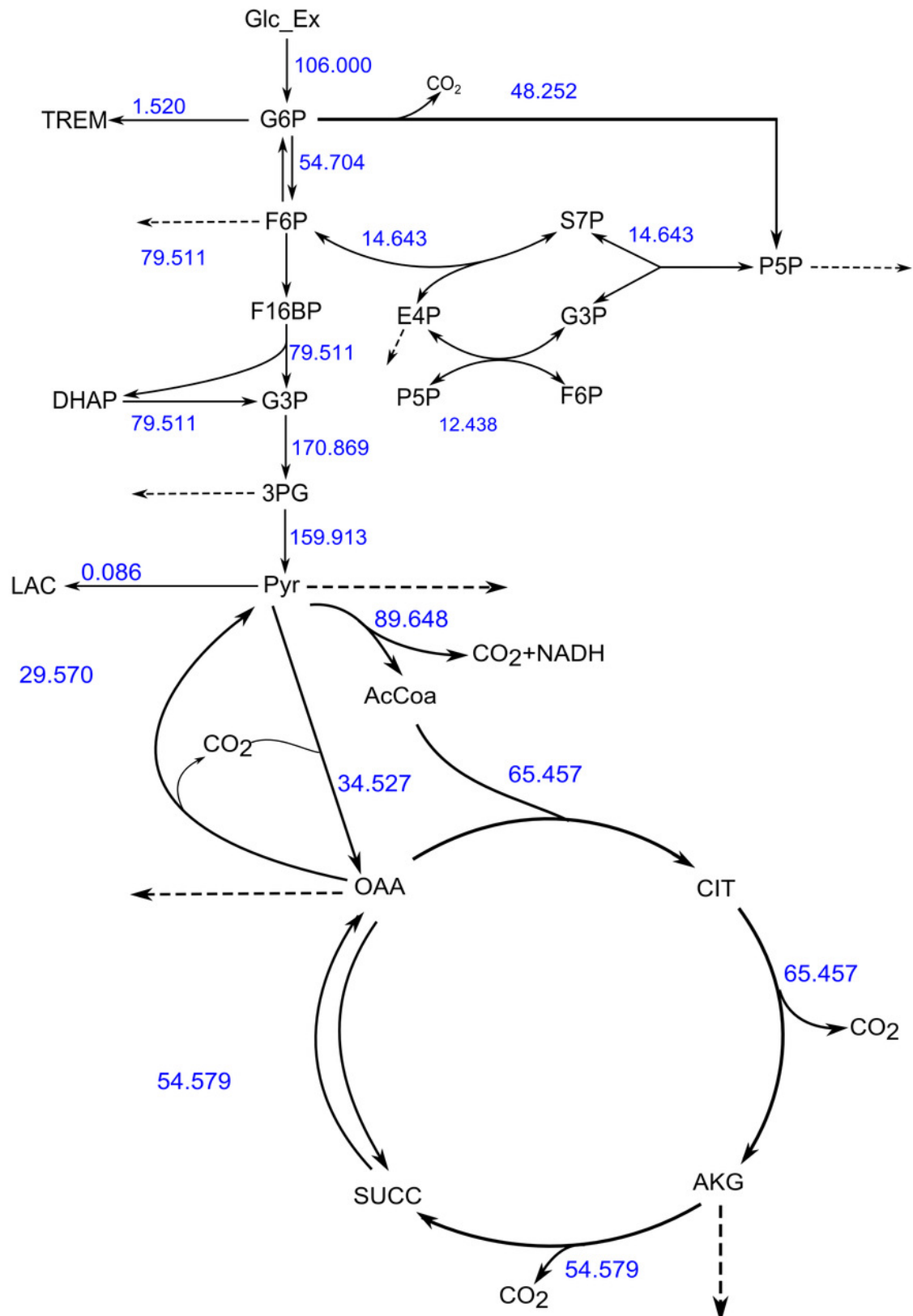
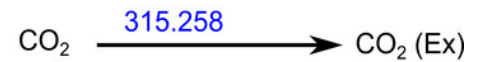
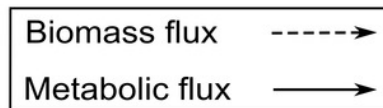
65.457

CO_2

54.579

SUCC   AKG

54.579

CO_2

Description of modules in FluxPyt.

Module	Description
main.py	Main driver module. Asks user inputs.
build_model.py	Parses csv model file. Generates metabolite and isotopomer models, stoichiometric matrix. Reads measured MIDs.
create_atm_transition_equations.py	Creates list of elementary reactions from atom transition equations.
input_substrate_emu.py	Reads substrate input file and calculates substrate MIDs required in the emu network.
mid_vec_gen.py	Generates correction vector for a given natural isotope distribution vector and number of atomic elements.
priliminary_fba.py	Generates initial feasible flux distribution using glpk as linear programming solver and sets feasible reaction bounds.
glpk_solve.py	Solves linear programming problems using glpk
make_emu_networks.py	Makes EMU networks from elementary reactions and measured MIDs
free_flux_null.py	Calculates null matrix and free fluxes.
solve_mfa_lm.py	Generates and solves non-linear optimization problem using lmfit.
solve_mid_networks.py	Solves EMU network to find calculated MIDS with respect to parameters of the non-linear problem.
mid_corr.py	Corrects calculated MIDs
draw_flux_map.py	Draws flux map from optimization result and flux map template.
monte_carlo.py	Performs Monte Carlo analysis.
bootstrap.py	For bootstrapping results from Monte Carlo analysis.

Mass isotopomer distributions (MIDs) for the TCA cycle model as calculated by FluxPyt and OpenFLUX.

		FluxPyt	**OpenFLUX**
VALX:11111	M+0	0.0298	0.0298
	M+1	0.2438	0.2438
	M+2	0.0589	0.0589
	M+3	0.4399	0.4399
	M+4	0.0292	0.0292
	M+5	0.1984	0.1984
LYSX:111111	M+0	0.0344	0.0344
	M+1	0.1854	0.1854
	M+2	0.1677	0.1677
	M+3	0.2499	0.2499
	M+4	0.2074	0.2074
	M+5	0.0764	0.0764
	M+6	0.0788	0.0788
ASPX:1111	M+1	0.0661	0.0661
	M+2	0.3532	0.3532
	M+3	0.2497	0.2497
	M+4	0.1590	0.1590
	M+5	0.1720	0.1720
Sum of squared residual (SSR)		0.0002	0.0002

Comparison of the optimal flux values for the TCA cycle model as calculated by FluxPyt and OpenFLUX.

Reaction ID	Reaction Formula	FluxPyt	OpenFLUX
V01	PYR_EX -> PYR	1.00	1.00
V02	GLU_EX -> AKG	0.20	0.20
V03	PYR -> ACCOA + CO2 + NADH	0.62	0.62
V04	ACCOA + OAA -> ICI	0.62	0.62
V05	ICI -> AKG + CO2 + NADH	0.62	0.62
V06	AKG -> 0.5 SUC + 0.5 SUC + CO2 + NADH + ATP	0.59	0.59
V07	SUC -> OAA + FADH2 + NADH	2.62	2.61
V08	OAA + FADH2 + NADH -> 0.5 SUC + 0.5 SUC	2.02	2.02
V09	OAA -> PYR + CO2	0.30	0.30
V10	PYR + CO2 + ATP -> OAA	0.48	0.48
V11	2 NADH + O2 -> 4 ATP	1.21	1.21
V12	2 FADH2 + O2 -> 2 ATP	0.30	0.30
V13	O2_EX -> O2	1.51	1.51
V14	CO2 -> CO2_EX	1.65	1.65
V15	ATP -> ATPM	5.56	5.54
V16	PYR -> PYR_B	0.20	0.20
V17	AKG -> AKG_B	0.23	0.23
V18	OAA -> OAA_B	0.15	0.15

The 95% and 68% confidence intervals for the TCA network calculated by the Monte Carlo analysis in FluxPyt.

Reaction ID	-95% CI	-68% CI	median	+68% CI	+95% CI
V01	1.00	1.00	1.00	1.00	1.00
V02	0.19	0.19	0.20	0.21	0.21
V03	0.59	0.60	0.62	0.64	0.66
V04	0.59	0.60	0.62	0.64	0.66
V05	0.59	0.60	0.62	0.64	0.66
V06	0.57	0.58	0.59	0.60	0.60
V07	1.30	1.74	2.59	4.17	9.41
V08	0.72	1.15	2.01	3.58	8.81
V09	0.29	0.29	0.30	0.31	0.32
V10	0.46	0.47	0.48	0.49	0.50
V11	1.17	1.18	1.21	1.24	1.26
V12	0.29	0.29	0.29	0.30	0.30
V13	1.46	1.48	1.50	1.53	1.56
V14	1.57	1.60	1.65	1.70	1.74
V15	5.33	5.41	5.53	5.66	5.76
V16	0.17	0.18	0.20	0.22	0.23
V17	0.20	0.20	0.23	0.26	0.26
V18	0.12	0.12	0.15	0.18	0.18

The MIDs for the *C. glutamicum* metabolic network calculated by FluxPyt and OpenFLUX.

		Measured MIDs (Quek et al., 2009)	FluxPyt	OpenFLUX
Ala 260	M+0	0.5085	0.5096	0.5095
	M+1	0.3529	0.3537	0.3538
	M+2	0.1058	0.1062	0.1063
Val 288	M+0	0.3455	0.3472	0.3477
	M+1	0.3983	0.3979	0.3978
	M+2	0.1845	0.1840	0.1838
Thr 404	M+0	0.3330	0.3342	0.3341
	M+1	0.3764	0.3750	0.3758
	M+2	0.1957	0.1958	0.1957
Asp 418	M+0	0.3343	0.3336	0.3334
	M+1	0.3732	0.3743	0.3750
	M+2	0.1955	0.1962	0.1961
Glu 432	M+0	0.2469	0.2499	0.2493
	M+1	0.3648	0.3652	0.3657
	M+2	0.2412	0.2392	0.2396
Ser 390	M+0	0.4497	0.4483	0.4486
	M+1	0.3576	0.3582	0.3581
	M+2	0.1428	0.1437	0.1435
Phe 336	M+0	0.2712	0.2736	0.2736
	M+1	0.3816	0.3812	0.3814
	M+2	0.2282	0.2282	0.2283
Gly 246	M+0	0.7407	0.7416	0.7416
	M+1	0.1845	0.1852	0.1852
Tyr 466	M+0	0.2344	0.2356	0.2356
	M+1	0.3530	0.3560	0.3561
	M+2	0.2423	0.2448	0.2448
Tre 361	M+0	0.0613	0.0618	0.0618
	M+1	0.6040	0.6059	0.6060
	M+2	0.2070	0.2071	0.2070
SSR			690	701

The MIDs for the *C. glutamicum* measured by Shupletsov et al., 2014, vs. those calculated by FluxPyt.

		Measured*	Simulated
Ala 260	m	0.504	0.506
	m+1	0.352	0.354
	m+2	0.107	0.109
Val 288	m	0.346	0.346
	m+1	0.397	0.397
	m+2	0.185	0.185
	m+3	0.056	0.056
Thr 404	m	0.332	0.333
	m+1	0.373	0.375
	m+2	0.195	0.196
	m+3	0.073	0.074
Asp 418	m	0.331	0.333
	m+1	0.373	0.375
	m+2	0.195	0.197
	m+3	0.074	0.074
Glu 432	m	0.248	0.248
	m+1	0.364	0.366
	m+2	0.240	0.241
	m+3	0.103	0.103
Ser 390	m	0.443	0.448
	m+1	0.353	0.358
	m+2	0.142	0.145
	m+3	0.049	0.050
Phe 336	m	0.273	0.273
	m+1	0.381	0.379
	m+2	0.229	0.228
	m+3	0.087	0.087
Gly 246	m	0.731	0.737
	m+1	0.183	0.189
	m+2	0.072	0.074
Tyr 466	m	0.235	0.235
	m+1	0.355	0.354
	m+2	0.245	0.245
	m+3	0.112	0.112
Trem 361	m	0.062	0.062
	m+1	0.605	0.605
	m+2	0.208	0.208
	m+3	0.097	0.097
SSR			**10.33**