

FIA Software User Manual

2010-06-25

Contents

1	Introduction	2
1.1	Background	2
1.1.1	Main Features	2
1.1.2	Acknowledgements	3
2	Getting Started	4
2.1	Prerequisites	4
2.2	Installation	4
2.2.1	Installing using Ubuntu	4
2.2.2	Installing from scratch	5
2.3	FIA command line interface	5
2.4	FIA graphical user interface	6
2.5	Running a simple example	7
3	The FIA input file	10
3.1	Introduction	10
3.2	The PROJECT section	11
3.3	The NETWORK section	11
3.4	The EQUALITIES section	12
3.5	The FLUX_MEASUREMENTS section	12
3.6	The LABEL_INPUT section	12
3.7	The LABEL_MEASUREMENT section	13
3.8	The MASS_SPECTROMETRY section	13

Chapter 1

Introduction

1.1 Background

Stoichiometric balance equations are a set of linear constraints representing knowledge of a particular metabolic network structure. Unfortunately, this set of equations is ill-posed, yielding an infinite number of possible values for the metabolic fluxes. In order to fully quantify intracellular metabolic fluxes, various experimental procedures have been developed, exposing hidden features of the metabolic network and thus adding additional equations to the stoichiometric set. One of these methods is ^{13}C metabolic flux analysis (MFA), in which the metabolic pathway is supplied with an isotopically labeled carbon substrate whose influence on the labeling of various metabolite pools within the network is subsequently measured. Mathematically, these experiments are traditionally analyzed using separate variables representing the fluxes and labeling states of the metabolic system. Analysis of tracer experiments using this set of variables results in a difficult non-convex optimization problem, suffering from both implementation complexity and various convergence issues. In addition, the error model is relaxed in order to ease the optimization process. This relaxation detracts from the statistical quality of the results and, in addition, often necessitates estimation of measurement normalization factors. These considerations render the experimental process and its analysis a very complex (and in some cases impossible) task.

The FIA software addresses the mathematical and computational problems of ^{13}C MFA experiments using a new set of variables referred to as *fluxomers*. Combining both fluxes and isotopomers, fluxomer variables result in a simply-posed problem with a corrected error model that enables the analysis of incomplete measurement data without the need to estimate additional normalization factors. A simple yet powerful fluxomer iterative algorithm (FIA) is constructed and run in order to solve the MFA optimization problem.

Complete documentation of the algorithm is available at *[CITE THE ARTICLE](#)*.

1.1.1 Main Features

The main features of the FIA software are:

- Simple text declaration of the metabolic network and the experimental data (including support for the 13CFLUX FTBL file format).
- NO initial values nor variable types (e.g. free fluxes) are required.
- Supports metabolic fluxes equalities and directionality constraints.
- Supports text based metabolic fluxes formula constraints.
- Supports direct metabolic fluxes value measurements.
- Supports both complete and incomplete measurements of metabolic labels.
- Supports direct use of raw MS measurements data.

1.1.2 Acknowledgements

This work was done as part of the Masters thesis work at the Technion, Israel Institute of Technology.

Chapter 2

Getting Started

2.1 Prerequisites

FIA is written in python, and uses the scipy package.

- Python (preferably 2.6), <http://Introductionwww.python.org>.
- NumPy, <http://www.numpy.org> or <http://www.scipy.org> .
- SciPy, <http://www.scipy.org> .
- UMFPACK, <http://www.cise.ufl.edu/research/sparse/umfpack/> .

2.2 Installation

If all prerequisites are met, installing FIA is no more than downloading the source file from the author's website.

Installing the prerequisites (when necessary) can be sometimes a bit harder. The following section covers the possible methods for installing the required prerequisites. The easiest approach would be using Ubuntu or other well supported linux distribution.

Note: When maximization of the CPU capabilities is required (for very large simulations or in order to support multiple processors) source compilation of the UMFPACK and ATLAS libraries is highly recommended.

2.2.1 Installing using Ubuntu

Installing the prerequisites under Ubuntu(<http://www.ubuntu.com>) (tested using Ubuntu Desktop version 10.04, but should work with other versions as well) can be done simply by invoking the command:

```
sudo apt-get install python-scipy python-tk
```

2.2.2 Installing from scratch

1. Install Python
 - Download from <http://www.python.org/> .
 - Unpack and install Python.
2. Install UMFPACK
 - Best thing would be installing the binary package for your machine. If unavailable, compilation from sources is also possible.
 - Download sources from <http://www.cise.ufl.edu/research/sparse/umfpack/> .
 - Follow the author's compilation instructions for the building procedures.
3. Install Scientific Tools for Python
 - Download NumPy from <http://www.numpy.org/> or <http://www.scipy.org>.
 - Unpack and Install the NumPy package.
 - Download Scipy from <http://www.scipy.org>.
 - Unpack and Install the SciPy package. Note the specifications for UMFPACK support.

2.3 FIA command line interface

The heart of the FIA software is its optimizer, which is a command line based script. For ease of use, a Graphical User Interface (GUI) for that script was developed (see the section below), eliminating the need of using the command line. Unexperienced users are encouraged to use the GUI interface.

SYNOPSIS:

```
fia_analysis {-i[=--fia] PROJFILE | -t[=--ftbl] PROJFILE}  
             [-o[=--output_file] OUTPUTFILE]
```

USAGE:

fia_analysis supports two types of input file formats: FIA and 13CFlux FTBL. The two are very much alike (see below), except for the fact that the FIA format supports only declaration of uni-directional fluxes, and does not contain the FLUXES and INEQUALITIES sections. For bi-directional fluxes, FIA simply defines two seperated fluxes. When analyzing 13CFlux FTBL files, fia_analysis looks for C=0 constraints in the FLUXES XCH section in order to determine directionality of fluxes.

OPTIONS:

```
-i --fia  
    Specifies that the metabolic pathway input  
    file <PROJFILE> is a FIA formatted file.  
    All fluxes are assumed unidirectional  
    (for bi-directional specify 2 seperated fluxes).
```

```
-t --ftbl  
    Specifies that the metabolic pathway input
```

file <PROJFILE> is a 13CFlux FTBL formatted file.
Flux is assumed unidirectional unless C=0
constraint is applied to it in the FLUXES XCH section.
The FIA formatted file will be saved under <PROJFILE>.fia .

-o --output_file
optional.
Specifies the name of the result log output file.
The default file name is PROJFILE.out

at least one of the above must be supplied.

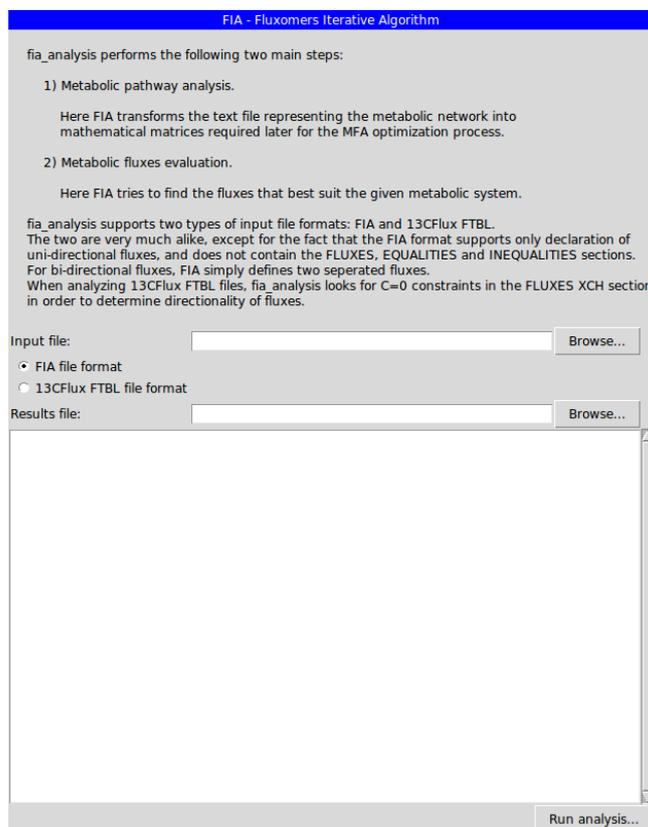
USAGE EXAMPLE:

fia_analysis -t temp.ftbl
Runs analysis & evaluation of the network defined in the
FTBL formatted input file temp.ftbl .
same as:
 fia_analysis --ftbl temp.ftbl

fia_analysis --fia temp.fia
Runs analysis & evaluation on the given FIA formatted input file.
Same as:
 fia_analysis -i temp.fia

2.4 FIA graphical user interface (GUI)

The graphical user interface of FIA is fairly simple, and is basically a front-end for the command line script “fia_analysis.py”.



The “Input file name” specifies the metabolic network and experiment data input file to be analyzed.

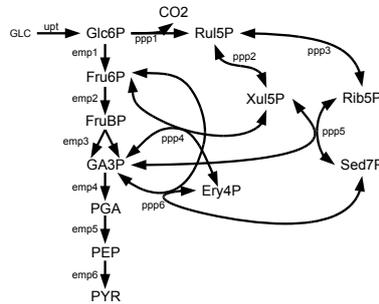
The combo box options “FIA” and “13CFLUX” allow the user to choose whether its input file format is FIA (where all fluxes are assumed unidirectional) or 13CFLUX format (where fluxes can be bidirectional, and thus additional FLUXES section is required).

The “Results file” specifies the name of the output and log file to be created. This file contains the final estimated fluxes values, and details about their agreement to the experimental data.

The bottom text area shows the progress of the simulation, along with its final results.

2.5 Running a simple example

Let us start by running a simple example, analyzing the *Embden-Meyerhof* and *Pentose Phosphate* metabolic pathways of *Escherichia coli*.

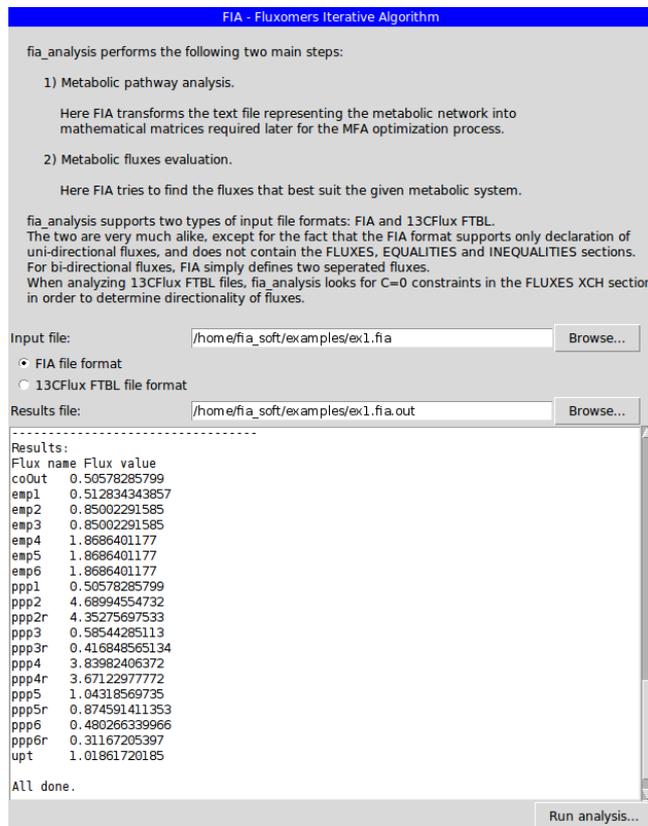


Setting up and running the simulation:

1. Input file selection - choose the “ex1.fia” file from the supplied examples directory. This input file declares the metabolic network and provide some experimental data to be analyzed.
2. Choose the file type - in this case “FIA” is already selected since our file extension indicated the software of its type. Files with “.fia” extension are assumed to be in FIA format. Files with “.ftbl” extension are assumed to be in 13CFLUX FTBL format.
3. Choose the output file name - the default output file name is an addition of “.out” extension to the input file name.
4. Running the simulation - simply click the “Run analysis...” button and wait for the software to finish!

Exploring the simulation results:

Once completed running, the simulation result file can be examined. For ease of use, the text block of the FIA GUI automatically shows its output:



If everything went well, an “All done.” message should appear at the bottom of the text area. Right above it, a table of fluxes and their associated estimated values can be found. Further up, various parameters of the simulation data and process can be found (running time, accuracy etc’).

Chapter 3

The FIA input file

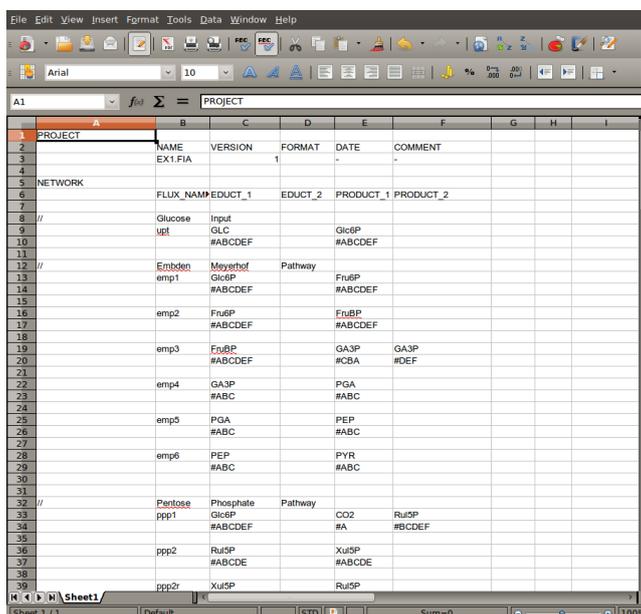
3.1 Introduction

The FIA input file completely defines both the network and the experimental data for the ^{13}C tracer experiment analysis. The file itself is a tab delimited text file, based upon the FTBL file format of the commonly used MFA estimation software 13CFLUX [?] (<http://www.13cflux.net/> and <http://www.uni-siegen.de/fb11/simteec/software/13cflux/>).

Tab-delimited files are simply tables in text format. Moving to the cell on the right (next column) is done by inserting the separator (hitting TAB), and moving to the next row is done by moving down a line (hitting ENTER). The easiest method for editing tab delimited files would be using any spreadsheet software (MS Excel, OpenOffice.org Calc, etc'), while importing and exporting in "text (tab-delimited)" or CSV (and selecting TAB as separator) format. Simple text editors (notepad, emacs etc') can of course also be used.

Adding comments to the file is possible by using the "//" marker. Everything in the row following this mark will be ignored.

The FIA file is constructed out of various sections. Every section begins with the section name as the first column's data and no other fields are filled within this row. After the section name row, its data is inserted according to the section's format.



In the screen-shot above one can see the two sections (PROJECT and NETWORK), and the comments lines (those starting with “//”).

3.2 The PROJECT section

The PROJECT section is a free text section, commonly used for documentation purposes. The suggested table includes field for the name of the project, its version, date and free text comments.

3.3 The NETWORK section

The NETWORK section is used to define the entire stoichiometric network.

Unlike the FTBL format, FIA defines only unidirectional fluxes. Bi-directional fluxes can be defined by adding a reverse fluxes as stand-alone fluxes where needed. The section is constructed as a table with 5 rows: FLUX_NAME, EDUCT_1, EDUCT_2, PRODUCT_1 and PRODUCT_2. Each flux is defined using 2 rows. The first indicates the flux name and its metabolic transactions. The second defines its inner isotopomers transactions. Let us demonstrate by defining ppp1 of the E. Coli network above:

FLUX_NAME	EDUCT_1	EDUCT_2	PRODUCT_1	PRODUCT_2
ppp1	G6P		Ru5P	CO2
	#ABCDE		#BCDEF	#A

Here we define a flux named ppp1 going from G6P to Ru5P+CO2, where the first atom of the G6P metabolite goes to the CO2, and the rest go to the Ru5P. Note that since there is only one input metabolite, the column of EDUCT_2 is left empty.

Using the same concept, defining ppp5 above is done by:

FLUX_NAME	EDUCT_1	EDUCT_2	PRODUCT_1	PRODUCT_2
ppp5	Xul5P	Rib5P	Sed7P	GA3P
	#ABCDEF	#abcde	#ABabcde	#CDE
ppp5r	Sed7P	GA3P	Xul5P	Rib5P
	#ABabcde	#CDE	#ABCDEF	#abcde

Since ppp5 is a bidirectional flux, we use two fluxes for its definition. The forward flux, called ppp5, transforms Xul5P+Rib5P into Sed7P+GA3P. The reversed flux, ppp5r, transforms Sed7P+GA3P into Xul5P+Rib5P, with reversed carbon atom transitions.

If the FTBL file format is supplied, the software automatically generates reversed fluxes for the bidirectional ones after analyzing the FLUXES sections of the FTBL file.

3.4 The EQUALITIES section

The EQUALITIES section is used to add equality constraints for the metabolic network and fluxes. In order to support the legacy of the FTBL file format, it has a subsection called NET (which is its only subsection). Since the FIA format contains only unidirectional fluxes, for our use it makes no difference. When 13CFLUX files are imported however, only this section is analyzed. The formulas are automatically translated into their bi-directional equivalents.

The EQUALITIES section contains a table with only two columns: VALUE and FORMULA. Yet, as can be seen, it is very powerful:

VALUE	FORMULA
0	4*ppp1-ppp5
0.387162	2*ppp1+2*ppp2+ppp3+ppp4

The first row demands the value of ppp5 to be equal to 4 times the value of ppp1 ($ppp5=4*ppp1$). The second row enforces the formula $2*ppp1+2*ppp2+ppp3+ppp4 = 0.387162$ to apply.

3.5 The FLUX_MEASUREMENTS section

The FLUX_MEASUREMENTS section is used to supply FIA with fluxes measurements values. The user supplies the software the value of the measurement along with its standard deviation. For example:

FLUX_NAME	VALUE	DEVIATION
upt	2.3	0.2

Here we supply measurement of the flux “upt” with the value of 2.3 (units here are some factor of reactions/sec, and are usually normalized by the input flux value) and standard deviation of 0.2.

3.6 The LABEL_INPUT section

The LABEL_INPUT section is used to supply FIA with known enrichments of its input fluxes.

The data is supplied as follows:

META_NAME	ISOTOPOMER	VALUE
GLC	#000000	0.44500000
	#100000	0.50000000
	#000001	0.01100000
	#000010	0.01100000
	#000100	0.01100000
	#001000	0.01100000
	#010000	0.01100000

Here we provide FIA with the structure of the input fluxes. We specify that 50% of the flux is constructed from the isotopomer 100000 (left carbon labeled, other unlabeled), 44.5% completely unlabeled, and the rest uni-labeled isotopomers appear each in 0.011% of the quantity.

3.7 The LABEL_MEASUREMENT section

The LABEL_MEASUREMENT section is used to supply FIA with the measured enrichments of fluxes throughout the system.

The data is supplied as follows:

META_NAME	CUM_GROUP	VALUE	DEVIATION	CUM_CONSTRAINTS
Rul5P	1.00000000	0.19800977	0.00200000	#1xxxx
	2.00000000	0.01859482	0.00200000	#x1xxx
	3.00000000	0.02580861	0.00200000	#xx1xx
	4.00000000	0.01245740	0.00200000	#xxx1x
	5.00000000	0.09924171	0.00200000	#xxxx1
Ery4P	1.00000000	0.05536176	0.00200000	#1xxx
	2.00000000	0.02301332	0.00200000	#x1xx
	3.00000000	0.01543625	0.00200000	#xx1x
	4.00000000	0.07323575	0.00200000	#xxx1

The META_NAME column indicates which metabolite is measured. The CUM_GROUP is simply index for the measurement. VALUE specifies the probability of the isotopomer supplied in CUM_CONSTRAINTS in the system. DEVIATION specifies the standard deviation of the measurement.

For example, here we see that the measured probability of attaining the isotopomer Rul5P_1xxxx is 0.198 (x indicates “everything”, hence either 1 or 0), with std. of 0.002.

3.8 The MASS_SPECTROMETRY section

The MASS_SPECTROMETRY section is used to supply FIA with the data of the MS measurements. MS measurements specify relative strength of isotopomers, separated by the number of labeled carbon atoms in them (isotopomers with no labeled atoms, only 1 labeled atom, 2 etc’).

The data is supplied by:

META_NAME	FRAGMENT	WEIGHT	VALUE	DEVIATION
Asp	1,2,3,4	0	0.513	0.020
		1	0.214	0.020
		2	0.192	0.020
		3	0.047	0.020
		4	0.012	0.020
Asp	1,2,3,4	0	0.508	0.020
		1	0.203	0.020
		2	0.192	0.020
		3	0.056	0.020
		4	0.019	0.020
Asp	2,3,4	0	0.556	0.020
		1	0.250	0.020
		2	0.144	0.020
		3	0.034	0.020

Here we can see that some metabolite named Asp was measured 3 times. Twice all of its atoms were inspected (atoms 1,2,3 and 4), and once only atoms numbered 2,3 and 4. As an example, the first measurement indicates that the weight of finding isotopomers with 2 labeled atoms is 0.192, with std. of 0.02, while the weight of 3 labeled atoms is 0.047 (with the same std.). Hence the probability of finding 2 labeled atoms isotopomers in this case is $0.192/0.047$ times higher than the probability of finding 3 labeled atoms.

Note: FIA can handle multiple MS measurements of the same quantities by simply adding all of the constraints to the optimization problem, and finding the solution that minimizes the error for the entire set of equations.