How to use the MATLAB code for MFA

The code requires three input files in XLS format: the metabolic model, atom mapping, and the experimental data. Please use the example files in the folder as a template for creating your own input files.

The metabolic model file (model.xlsx) contains two tabs: “Reactions” and “Metabolites”.

In the “Reactions” tab:

RxnID is the abbreviated name of the reaction. E.g. PGI

Rxn name is the full name of the reaction. E.g. Phosphoglucose isomerase for PGI

Rxn Formula contains the stoichiometric reaction formula for the overall reaction expressed using abbreviated metabolite names. Stoichiometric coefficient is specified in parenthesis before the metabolite E.g. (1) G6P.

Reversible reactions are indicated using a double-headed arrow (⇔) and irreversible reactions are indicated using a single headed arrow (→). A default lower bound of -10^8 is set for reversible reactions. Irreversible reactions have a lower bound of 0.

The default upper bound for all reactions is set to 10^8. Alternatively, if the direction of flux through a reversible reaction is known to be in the reverse direction only, the upper bound for such reactions can be set to 0.

Macro/bm is not used for FBA. It is used to exclude reactions with non-integer stoichiometric coefficients (such as biomass reaction, protein synthesis) when checking for correctness of the atom mapping model.

Subsystem describes the pathway(s) to which the reaction belongs. It is included for convenience of data analysis once fluxes have been elucidated.

A distinguishing feature of MFA models is that reactions are not allowed to have an empty reactants or products side. Unlike COBRA models, the exchange reaction representing glucose uptake cannot be “(1) glc-D[e] ⇔” with an empty products side. It must instead be represented as

“(1) glc-D[supplied] → (1) glc-D[e]”. These reactions must always be irreversible, i.e., substrates and products of a process are pre-specified.

This means that certain metabolites will always be unbalanced.

The metabolites tab contains details on the list of metabolites in the model.

“metid” contains the list of abbreviated metabolite names. The corresponding metabolite names are contained under “metnames”.

Only “metid” needs to be specified for constructing the metabolic model and performing analyses using stoichiometric methods (any type of FBA except thermodynamic-FBA, FBA, OptKnock, etc…). The supplied code checks for consistence between the list of metabolites specified in the “Metabolites” tab and the metabolite abbreviations used in the “Reactions” tab and returns an error if there are missing/extra metabolites. The error log also reports the list of missing/extra metabolites.

“metnames” is an optional input in this excel file. While it is not used in the code, this information allows the user to compare models from different sources as different sources use different metabolite IDs (e.g. BiGG, modelSEED, KEGG, etc).

“metformula” is also an optional input. It allows the user to check for elemental balancing in their constructed models but is not used in this code. Additional troubleshooting features will be introduced in subsequent versions of this code.

The second input file required for performing 13C-MFA is the atom mapping information. This excel file contains two tabs: “Atom Transition” and “Symmetry”.

The “Atom Mapping” tab details the tracing of carbons from reactants to products. in this Tab:

“rxnAbbreviation” is the RxnID from the “Reactions” tab in the metabolic model input file.

“metabAbbreviation” is the metabolite ID of the metabolites participating in the reaction.

“ReactantProductFlag” indicates whether the metabolite is a reactant or a product in the reaction represented by specified RxnID.

“startNodeSymbol” represents the symbol of the element being mapped. E.g. C for carbon, H for hydrogen, O for oxygen, and N for nitrogen.

Every entry in “maps” contains three pieces of information:

(a) How many carbons are contained within the metabolite being mapped.

(b) The identifier of the carbons.

(c) The position of the identifier.

For example, G6P contains six carbons. The identifiers for these carbons is “1”, “2”, “3”, “4”, “5”, and “6” and correspond to carbons C1, C2, C3, C4, C5, and C6, respectively. This ordering of carbons is implicitly encoded and must be consistent for all instances of metabolite G6P in the metabolic network. i.e., the identifiers are always in the order: C1,C2,C3,C4,C5,C6 for G6P.

Through this implicit encoding based on sequence of identifiers, atom tracing between reactants and products is achieved. For example, in PGI, the mapping of C1 of G6P to C1 of F6P is encoded by using a common identifier “1” for both atoms. Identifiers are only unique to a reaction. Every atom in the reactants side of a particular reaction is associated with a unique identifier and maps to exactly one atom in the products side. Mapping of identifiers is only valid within the scope of one reaction (e.g. “1” from G6P to “1” from F6P within PGI) and is not valid across reactions (e.g. “1” from G6P in PGI cannot be mapped to “1” from PEP in PYK).

With advances in databases for managing atom mapping information, the format of the input file and subsequently the code for reading the input file will be updated to be compatible with database formats.

The “Symmetry” tab contains information on the existence of a plane or point of symmetry within metabolites. This is an alternate permutation of atoms. For example, the symmetry of succinate is captured by the fact that the atom configurations C1-C2-C3-C4 and C4-C3-C2-C1 are equivalent.

The third and final input file contains the experimental data to be used for flux elucidation. Three pieces of information must be provided here: (a) list of MS measurements, (b) list of extracellular flux measurements, and (c) the tracer scheme used in the isotope labeling experiment.

In the MSData tab:

i) EMU refers to the fragment of a particular metabolite being measured.

ii) “fragment formula” refers to the complete chemical formula of the fragment whose labeling distribution is measured by MS. This information is used to correct for labeling contribution from natural abundance of untraced atoms.

iii) The data is the MDV and the “error” is the standard deviation for each measurement.

In the FluxData tab:

i) “flux” refers to the reaction whose flux is fitted.

ii) “value” is the experimentally measured flux.

iii) “error” is the standard deviation of the measurement.

In the TracerData tab:

i) “metabolite” is the list of metabolites that are labeled with 13C in the tracer experiment.

ii) “fraction” denotes what fraction of the metabolite is labeled (e.g. 100%, 50%, etc…). If the fraction for any metabolite is less than 1, the remaining is assumed to be labeled based on natural abundance (1.12%)

iii) “nCarbons” denotes the number of carbons in the metabolite used as a tracer.

iv) “position” denotes the carbon position(s) that is/are labeled.

v) “purity” denotes the extent of enrichment (99%, 99.5%, etc) as specified by the supplier.

If the model contains other carbon sources that are not listed in this tab, they are assumed to be labeled based on natural abundance of 13C (1.12%)

Instationary MFA requires time-course labeling data and therefore contains an additional column in the “MSData” tab called “time” and is specified in seconds.

An additional tab called “PoolData” can also be included if any pool-size measurements are to be fitted. This, however, has not been tested using available datasets.

The MFA code is executed in three steps:

The first step is to compile a model structure using the input data by calling the code “createmumodel.m” with the three excel files as inputs. It is important to note that if any labeling data is added or removed, this model must be created again as the EMU networks will change depending on included fragments.

The next step is to estimate fluxes using the function “flxestimate”. This returns an output file containing the fluxes and fit statistics.

Finally, confidence intervals are computed using “confintestimate”. By default, this code updates the output file generated in “flxestimate” to include lower and upper bounds for all fluxes. It also returns a second output file, which contains the lowest SSR solution. generally, this will be the same as the original output file. However, if a better solution is identified, that solution will be reported here (in “impres”) instead.

The output file:

The output of the MFA code is a MATLAB structure called “res” which contains all the information pertaining to the fit statistics, best solution, and the confidence intervals.

“DOF” corresponds to the degrees of freedom calculated as the difference between the number of data points and number of adjustable parameters (free fluxes and pool sizes).

For a prespecified confidence level α, “frange” is the expected range within which the minimum SSR (“fmin”) must lie for statistical acceptance. If min SSR is greater than the upper bound of frange, the model is likely missing some important parameter required to explain the experimental data. If SSR is less than the lower bound of frange, the experimental data is likely under-weighted and more precise measurements are needed.

“fluxes” contains the optimal flux distribution. The flux corresponding to the reaction in “id” is reported under “val” with the lower and upper bounds reported in “vLB” and “vUB” respectively.

“residuals” contains the lack-of-fit information for fitted fluxes (“flxfit”) and fitted MDVs (“mdvfit”). In each of these fields, “data” is the experimental data, “val” is the predicted flux/MDV, “WRES” is the weighted residual, and “SSRES” is the (sum of) squared residuals. SSRES is computed as $\mathbf{WRES}^T*\mathbf{WRES}$.