Institute for theoretical biology

Humboldt University Berlin

Matlab Automization Corescript

Author:
Raik Otto

Supervisor:
Dr. Ralf Steuer

January 23, 2012

Contents

1 Tutorial

What is needed:

- Python $\geq$ 2.6
- Octave or Matlab

In order to create a metabolic network with MAC we first create a .txt file:

1.1 Creating TXT files

All reactions of this network are conainted in this txt file along with their modifiers e.g. Keq values. Every line contains one reaction. First an optional ([]) label, then the subtrates, products and optional modifiers.

Example:

```
input: G1_x -> S1 ; +n8ADP C=S1_0.7 C=G1_x_3 KEQ=1.5 MM
activation: S1 + 2' ATP = 2' ADP + 2 S2 ; C=ATP_0.2 C=ADP_0.8 +n8S1 -n1S3
S2 + 2' ADP -> S3 + 2' ATP ; KEQ=2
export: S3 = G2_x ; KEQ=1.5 -n1S1
ATPase: ATP -> ADP
```

The pattern:

$$[label :]substrate_1 + .. + substrate_n = product_1 + .. + product_n \quad [; modifiers]$$

It is important to separate the label with a ":", the subtrates and products with "=" and the modifiers from the products with ";". A " ' " after the stochiometry of a metabolite signifies that a metabolite's stochiometric value is different from its kinetic value. Example: 2 ATP means stochiometry of 2 and a kinetic value of two, 2' ATP means stochiometry of 1 and a kinetic value of 2. A " ' " always sets the stochimetric value to 1.

modifiers:

- +ni[metabolite]: Reaction activated by the metabolite with hill coefficient i. Example: +n3ATP
- -ni[metabolite]: Reaction inhibited by the metabolite with hill coefficient i. Example: -n1S1
- C=[metabolite]_c : Default concentration for that metabolite is c. Example: C=ATP_0.7
- KEQ=k: Default KEQ value for that reaction is k. Example: KEQ=0.5
- MM,MMIR, MA, MAIR: Reaction default type is either mass action, mass action irreversible, michaelis-menten or michaelis-menten irreversible. IF there is no specification, the default reaction type is mass action. Example: MMIR Hint: One can write "=>" instead of 'MMIR' or 'MAIR'

1.2 Creating Matblab files

- Create the txt and put it into the MAC directory MAC/
- Call

```
:python MAC.py ExampleNetwork.txt
```

- This creates a directory with name 'ExampleNetwork' which contains 5 files.
- ExampleNetwork.m: Contains the calculations for the reactions.
- ExampleNetwork_Jacobi.m: Contains the calculations for the Jacobian matrix.

- ExampleNetwork.cps: The network a Copasi-readable file (alpha version)
- ExampleNetwork_Sim.m: Contains the simulation.
- ExampleNetwork_parameter.m: Contains the parameter-information.

1.3 Simulation

- Go to the ExampleNetwork directory ExampleNetwork/
- Start octave or matlab
- `source('ExampleNetwork_Sim.m')`
- Modify the default values in the sim file according to your wishes.

1.4 SBML files

...

2 Structure of MAC

2.1 Files

The Matlab Automization Corescript (MAC) consists of parser, scripts and function definitions.

To do: Table of contained files

3 Workflow

...

3.1 TXT to Python

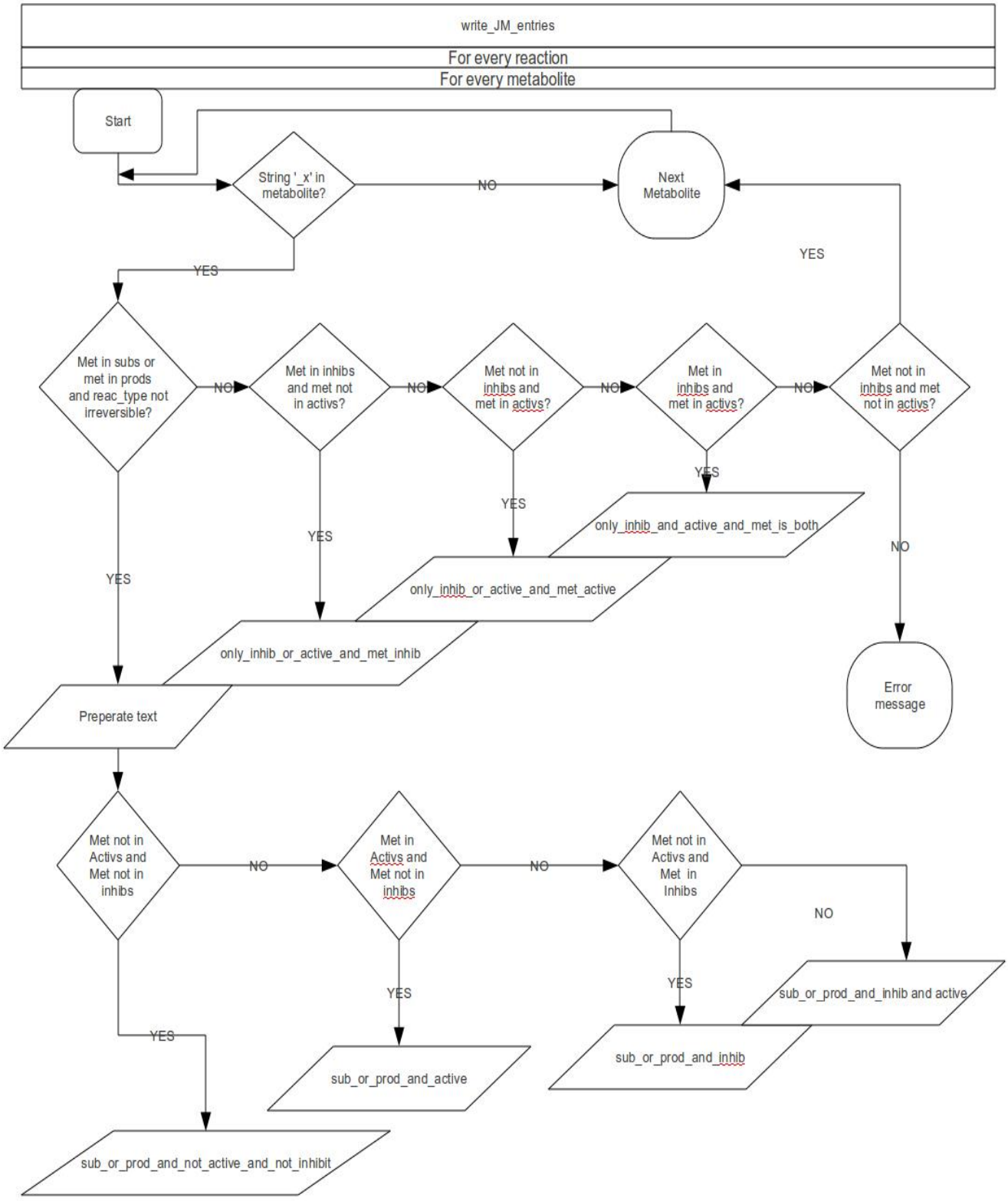
...

3.2 Python to Matlab

...

3.2.1 Jacobi Matrix

This pythonscript creates a matlab files which implements a Jacobi Matrix (JM). Workflow:



Every metabolite of every reaction is attributed to a position in the two dimensional JM if:

- It is not an external metabolite

- It is not a product in an irreversible reaction
- Exception: If a metabolite (met) is product in an irreversible reaction it still attributed in case it is an activator and or inhibitor

Formula for every met one derives:

$$v(i) = VM(i) \cdot reaction_rate(S, NS, P, NP, Keq(j)) \cdot REGA \cdot REGI$$

$$v'(i) = REGA \cdot REGI \cdot drate/dS + REGA \cdot dREGI/dS \cdot RATE + dREGA/dS \cdot REGI \cdot RATE$$

$$dvdS = reaction\_derive \cdot REGI \cdot REGA, dvdA = REGA\_derive, dvdI = REGI\_derive$$

If a met is only substrate, dvdI and dvdA are 0 and *not* written. The terms REGI and REGA are 1 and *not* written either. After the case decision as shown in the flow chart a function is called which creates the correct text for a metabolite, depending on the variables that were passed on. These functions, in return, access the *function.py* and *function.m* scripts which contain the direct function text.

3.3 Python to SBML

...

A Abreviations

act activator

inh inhibitor

JM Jacobi Matrix

MAC Matlab Automization Corescript

met metabolite

mets metabolites