Plant Metabolic Modelling

Toh Qin Wayne

Yale-NUS College, Singapore

Background and Rationale

Since computational modelling has been effective in metabolic engineering for microorganisms, it has been recently extended to the realm of plant metabolism engineering. However, the higher complexity of the metabolic pathways in plants have complicated the task of creating metabolic models of plants which are accurate in its simulation of metabolism.

Yet, the rewards from successfully creating accurate plant metabolic models are huge: we will be able use these models to identify the key metabolic pathways in plants that produce desired metabolites/biomasses for industry usage, and even analyze how some plant species are able to survive in certain climates more successfully than others. In this project, we modelled the plant species *Neuontobotrys tarapacana (*NTARA) in order to understand how the metabolism of such desert plants allows them to effectively grow in dry conditions.

Methods

1) Creation of basic metabolic model from source file containing Gene-Reaction Pathways.

Source File Formatting

Model File

- Filter out all unique reactions from the source file.

- Create an initial model of the plant using Gene information from the Metacyc Database, which might contain incorrect Genes/ GPR information.

- Fill up the Genes and GPR column with the correct information from the source file. Done via siphoning out all the genes related to each unique reaction in the source file.

- Identify metabolites that have missing information (related to its Charge, Neutral Formula etc) and source as much of these information from other completed models.

- Then, fill up these missing entries for the metabolites which we have information of, while leaving the rest blank.

2) Compartmentalize the reactions in the model.

- Using a reference Python dictionary which contains the name of the compartments and its constituent reactions, remove all of the reactions that are not in the model from this reference dictionary.

- Then, use this updated dictionary to compartmentalize the model. Compartmentalization entails the addition of proper suffixes at the end of the reaction (Data Tagging) to allow the user and program to identify which compartment the reaction belongs to.

How the Compartmentalization code works

Step 1: Add all reactions in model that are not in 'default' compartment of input dictionary to a new list, reac_not_default.

Step 2: Add all reactions in model that are not in both reac_not_default and default compartment of input dictionary to the default compartment.
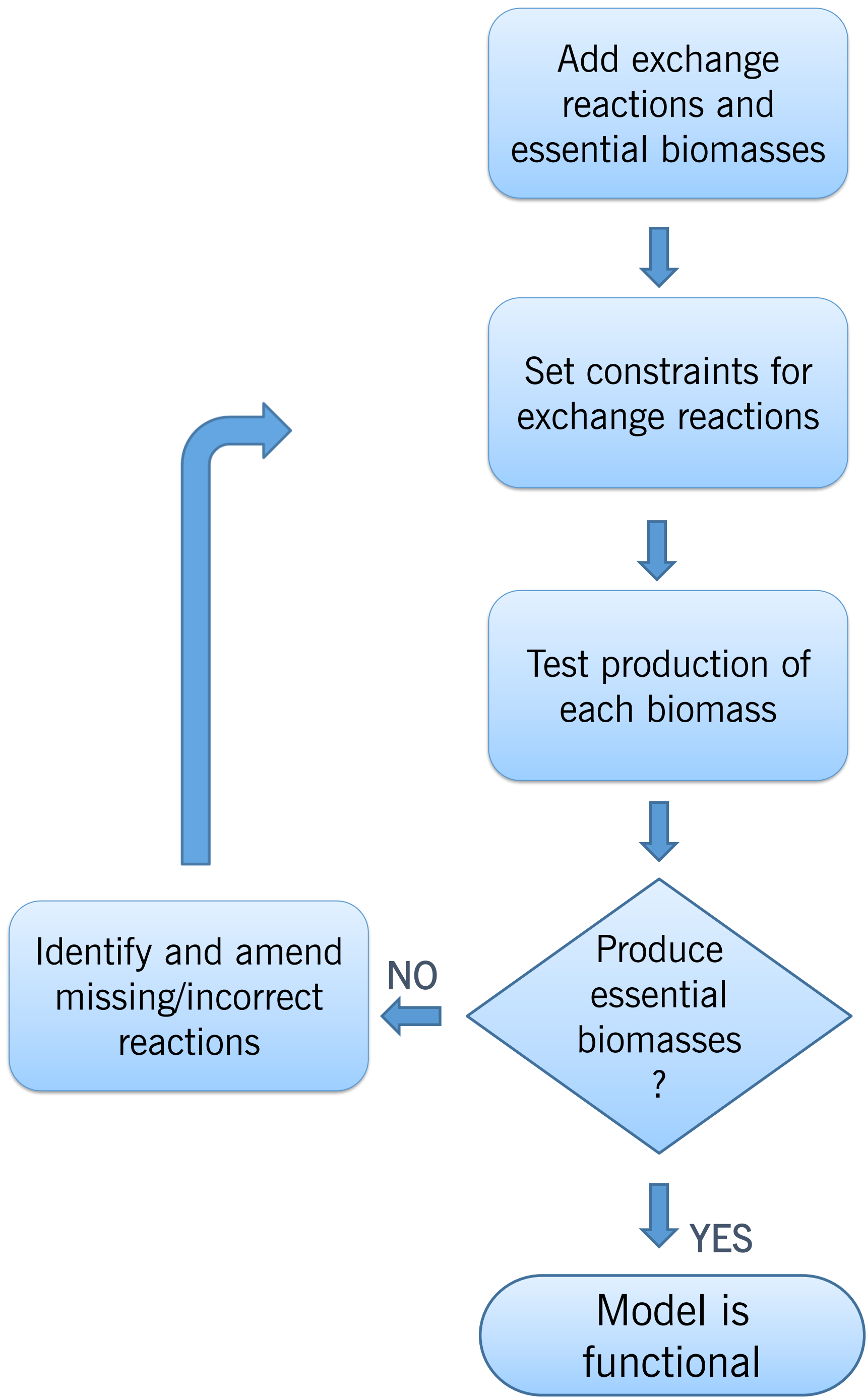
Step 3: Create an empty model, compartmented_model.

Step 4: Add all reactions in the model to the compartmented_model with their suffixes and IDs.

Step 5: Add all metabolites of the aforementioned reactions into the model too with their respective suffixes.

Step 6: Return this model, compartmented_model, to the user.

3) Test the model's ability to produce essential biomasses and check for unbalanced or missing reactions..

```
Add exchange reactions and essential biomasses
        ↓
Set constraints for exchange reactions
        ↓
Test production of each biomass
        ↓
Produce essential biomasses?
   NO → Identify and amend missing/incorrect reactions ↑
   YES ↓
Model is functional
```

Results and Future Directions

- The final draft model contains a sizeable amount of information, namely: 3073 Reactions, 3209 Metabolites, 9020 Genes.

- Upon testing the model, essential reactions containing Pumped-Proton have either incorrectly named metabolites or were missing from the model. Some essential transporters were missing from the model as well.

- After amending the above issues, the number of biomasses produced for the NTARA model rose to 11. The biomasses are: GLC, Glycerol, NO_3^-, Pi, SO_4^{2-}, Xylan, sETOH, sFUM, sGLC, sMAL and sSucrose.

- Some biomasses, however, remain unproduced, and work is needed to identify the remaining missing/incorrect reactions. Once we have a functional model, we can use FBA to simulate the metabolic processes of the plant.

Acknowledgements