metaboLITE: a web-based application for learning principles of metabolic modeling

Michal Stolarczyk, Maureen A. Carey, Jason A. Papin

Biological modeling is currently a niche technique, the way that flow cytometry, microfluidics, and even microscopy once were. However, like flow cytometry, microfluidics, and microscopy, computational modeling is a powerful technique and well-suited to address many biological questions. Modeling approaches, such as genome-scale metabolic modeling, highlight knowledge gaps, generate high-confidence data-driven hypotheses, and contextualize sparse data. Genome-scale metabolic reconstructions are built from genomic data and by inferring function to complete or connect metabolic pathways; these reconstructions are supplemented with data from functional genetic and biochemical studies. Genome-scale metabolic reconstructions represent our best understanding of an organism's biochemistry and cell biology and can be used to make predictions about the system. These predictions (*e.g.* essential components or pathways usage after changing environmental conditions) are high-confidence hypotheses, but even failed predictions illuminate knowledge gaps or unappreciated experimental complexities. A model can be viewed as a framework for storing thousands of hypotheses, such as hypotheses about gene-protein-reaction associations or enzyme function, as well as rigorously evaluating and documenting these hypotheses.

However, at least basic programming is necessary to use most computational tools, including metabolic models. Thus, learning to program is a major obstacle to obtaining the technical knowledge required to evaluate a model or to understand if metabolic modeling is the right approach to address your biological question of interest. Accordingly, usage of metabolic modeling is limited to researchers who feel comfortable with programming. This limits the diversity of thought in the metabolic modeling field and generates a barrier to entry for biologists with only experimental and subject matter expertise.

Thus, we developed a tool, named metaboLITE, to decouple learning to program and learning about genome-scale metabolic modeling. metaboLITE is a web-based application that visualizes and implements two small reconstructions (iSIM in **Figure 1**) and an analytic framework used to analyze metabolic reconstructions (**Box 1**). Importantly, metaboLITE is presented from a biologist's point of view. With metaboLITE, the user can perform three *in silico* experiments: (1) changing growth media, (2) knocking out genes, and (3) under/overexpressing enzymes. This simple shift in language retains all precision and accuracy of the technical details but is approachable to biologists.

Using metaboLITE

We recommend readers first visit **iSIM paper ref** as this manuscript builds on the ideas presented there.
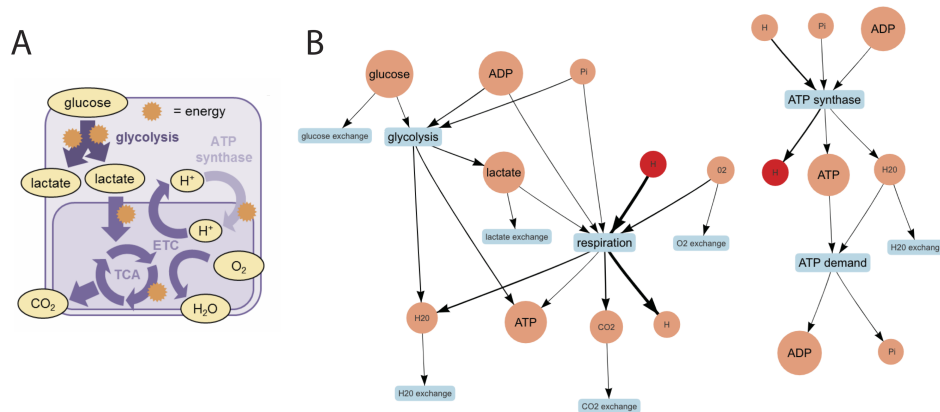

Figure 1: iSIM represents mitochondrial metabolism (A). This can also be represented as a network (B) with metabolites in circles and reactions as arrows connect the metabolites. Reactions are named in boxes. This model includes 10 metabolites, 9 reactions, and 9 genes. Panel A adapted from iSIM REFRENCE. ETC = electron transport chain, TCA = tricarboxylic acid cycle.

> **Box 1: What is FBA?**
>
> **Flux balance analysis**, or FBA, is just one analytic approach used with genome-scale metabolic models; however, it is the most common approach. FBA uses linear optimization and calculates steady-state fluxes for a metabolic system. 'Flux' can be thought of as a flow rate or the number of times a reaction is used per unit time, and is often convered to millimoles per gram dry weight per hour (mmol/gDW/h).

The app is interactive and dynamic presentation of small metabolite networks. Users first pick an experimental setup in metaboLITE by selecting a model (*i.e.* organism) of choice. Both a model of human mitochondrial metabolism, iSIM, and a model of core *E. coli* metabolism are preloaded for users. The app home page is shown in **Figure 1**; users pick an experimental setup on this screen. The visual representation of the model changes as simulation results are updated. The user can also moves metabolites and reactions in the field of view (by clicking and dragging) and can zoom in to interesting network features (see **Supplemental Video**). Help is offered throughout the experience in three places: first, there is a help page on the app's information website (michalstolarczyk.github.io/metaboLITE), second, there is an extra help page in the app, and, third, there are hover windows, indicated by question marks, throughout the app.

Here, we will describe the three kinds of experiments, or simulations, offered on metaboLITE. To emphasize, this app opens in a web browser (*i.e.* Chrome or Safari) and all functionality can be accessed without programming in a point-and-click environment like a website. We have provided code in the *Supplemental Information* for users who are interested in seeing the behind-the-sceens analysis for each experiment.

Simulation 1: Change media

Microbial growth behavior is dependent on the environment. Thus, a common phenotypic profiling technique involves growing an organism on many media types (*i.e.* growing on Biolog plates). This can also be done *in silico*. After changing the environmental conditions, we can measure how an organism's growth is affected.

To understand how this is implemented, it is important to understand that models contain spatially information (*e.g.* **Box 2**, reaction 2). A reconstruction typically has an extracellular and an intracellular compartment; models representing a eukaryotic cell's metabolism may also have additional compartments representing additional organelles or subcellular spaces like the mitochondria. The extracellular compartment represents the most proximal environment (*e.g.* media for free living organisms *in vitro* or host cell for intracellular organisms) and this compartment can be manipulated to represent changes in that environment.

So, in order to change the growth medium or environment, we change the availability of metabolites in that compartment. This is performed by adding constraints on exchange reactions. Exchange reactions represent dynamic features of the extracellular environment, like gas exchange into the media (**Figure 2A**) or a fresh bolus of media (**Figure 2B**). Thus, exchange reactions can reflect a metabolite's availability in media and we can constrain these reactions to represent different sets of nutrient availability. Constraints (**Box 2**) can change the flow through a reaction.
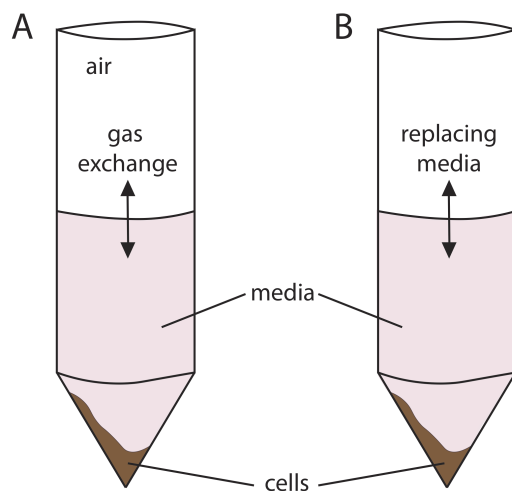
Figure 2: Biological motivation for exchange reactions.

Box 2: Metabolic reactions and constraints

Metabolic reactions are formalized in a metabolic model. Real life examples include *reaction1* and *reaction2*, and a theoretical example reaction is shown in *reaction3*. Note how *reaction2* contains the metabolite glucose with specific localization.

1. ATP + Glucose → ADP + Glucose 6-Phosphate + H+
   - An early step in glycolysis, catalyzed by hexokinase
2. Extracellular Glucose → Intracellular Glucose
   - Transport of glucose
3. A + B → C + D
   - Hypothetical reaction, catalyzed by *Enzyme1*

Metabolic constraints are layered onto these reactions. Constraints include *bounds* describing how much flux can go through the reaction in the forward or reverse direction. So, we can also constrain the direction of the reaction (forward, reverse, or both) and amount flux through the reaction (max, min). Another constraint is called the *'gene-protein-rule'*; this constraint summarizes what enzymes are associated with the reaction and which gene(s) encode the enzyme(s). *Gene1* must be present in the genome and then expressed as *Enzyme1* for *reaction3* to occur; this is an example of a biological constraint. So, *reaction3* can be constrained by its bounds or the availability of *metaboliteA, metaboliteB, metaboliteC, metaboliteD, Enzyme1*, or *Gene1*.

Returning to a media change experiment, we can simulate changes in metabolite availability by changing the bounds on these exchange reactions. After adding these constraints, FBA is performed to quantify the consequences of these changes (*i.e.* how they affect growth). The settings for three different media conditions are available: glucose-free media, microaerophilic media (*i.e.* oxygen limited), and lactate-rich media (left, black buttons, **Figure 3**). By simulating growth on these predefined medias, the user can see how media type affects growth both by visualizing flux on the network and by printing out flux values for each reaction. The user can also change the exchange limits on specific media components using a slider bar (left, 'Select the exchange limits', **Figure 3**).

Simulation 2: Simulate gene knockouts

Gene essentiality screens or targeted gene knockouts are also a common *in vitro* experiment that can be conducted *in silico*. Like in an *in vitro* knockout experiment, to simulate an *in silico* knockout, we remove a gene from the network. This will also remove the gene-associated reaction contained in the model. Following this, we often perform FBA to maximize an objective reaction. This analysis asks the question, can the metabolic system (cell) do something without that gene and gene-assiciated reaction? Typically, that *something* is *grow*, so we set the objective reaction to represent
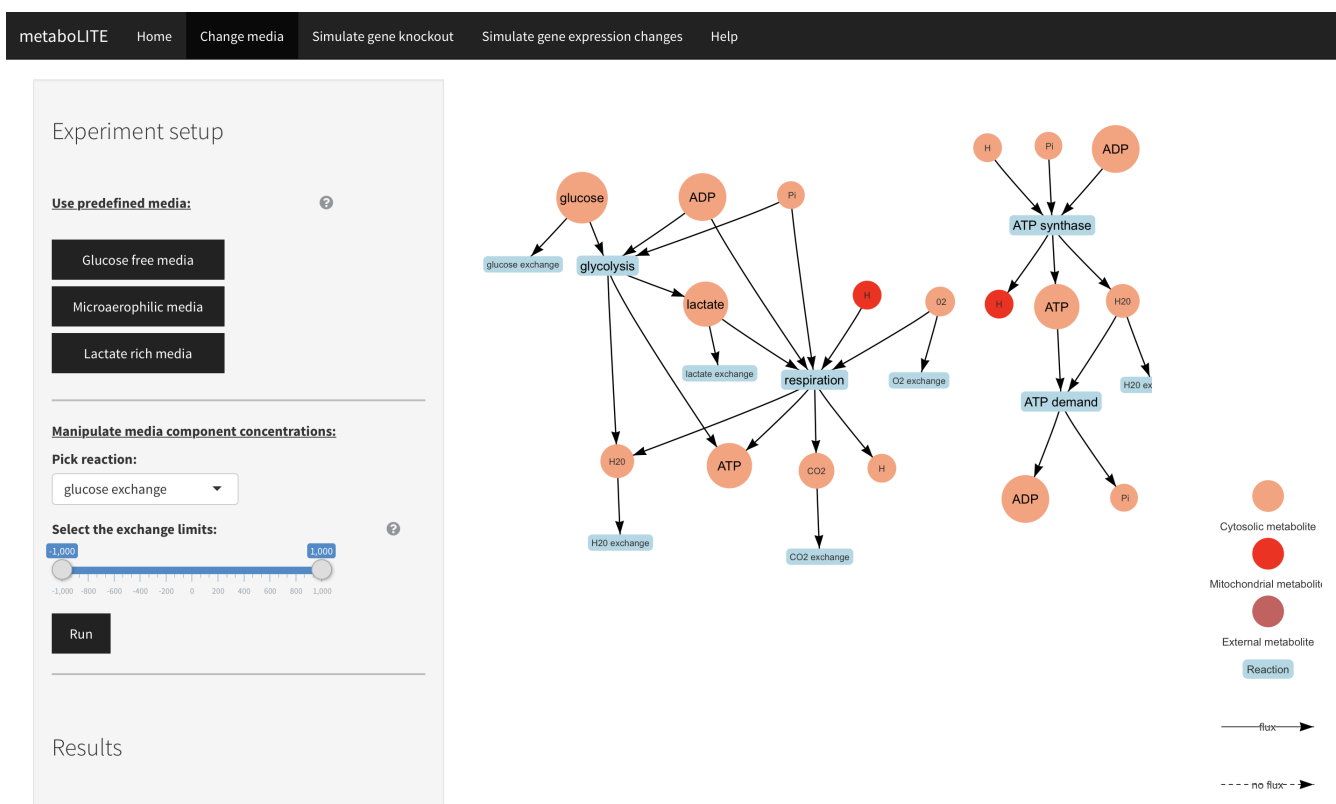
Figure 3: metaboLITE screen for changing media.

growth. However, we can set the objective reaction to be any measurable, interpretable biological observation. See **Box 3** to learn about objective reactions.

If the reaction is essential, the cell cannot grow (or produce whatever else is being measured). If the reaction is nonessential, the network will still be able to grow. Depending on how the model was built, you may be able to interpret the results quantitatively, rather than binary (growth v. no growth). For this simulation in metaboLITE, the user picks an enzyme from a drop down menu (left, **Figure 4**) to visualize and quantify flux throughout the network (right, **Figure 4**).
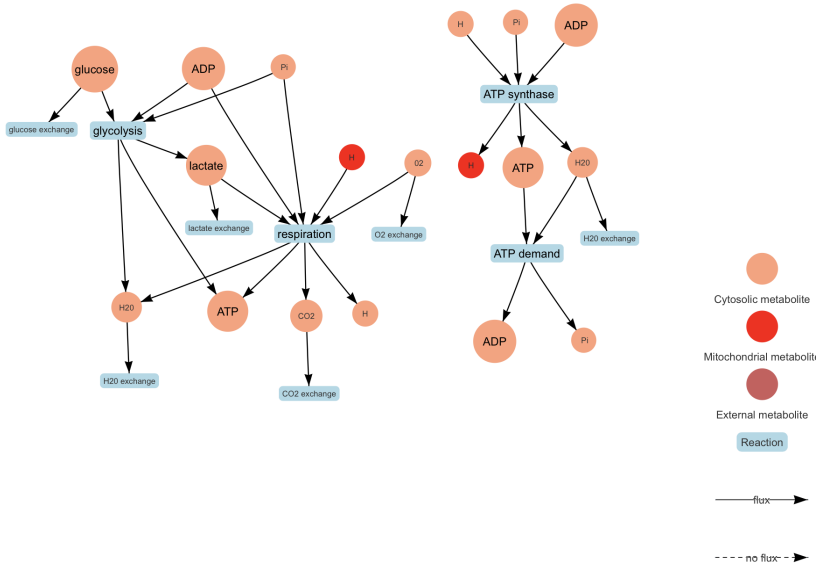
Figure 4: metaboLITE screen for performing gene knockouts.

Box 3: What is an objective reaction?

The objective reaction is the reaction you are trying to maximize or minimize. It is typically something you can measure. Imagine you have a system with three reactions:

1. A + B → C
2. A → D
3. D → C

Any of these reactions could be the objective reaction. More information is needed to make a logical choice. For example, in this system, you can measure *metaboliteC* and the system is grown in a media with *metaboliteA* and *metaboliteB*. The model for this system would have the following reactions represented:

1. A + B → C
2. A → D
3. D → C
4. → A (representing input from the media)
5. → B (representing input from the media)
6. C → (representing the measured output)

If you wanted to see how each reaction would be used to maximize production of *metaboliteC*, you would set *reaction* as your objective reaction and maximize. Oftentimes growth is used as the objective reaction, and growth is summarized into a reaction by adding up all of the metabolites required for the production of biomass. For the sake of discussion throughout this paper, we will assume the objective reaction used is a biomass reaction.

Simulation 3: Integrate transcriptomic data

The third simulation in metaboLITE is the integration of transcriptomic data. Expression data is often used to make a condition-specific model, or a model that represents a cellular phenotype in a particular condition. Expression data
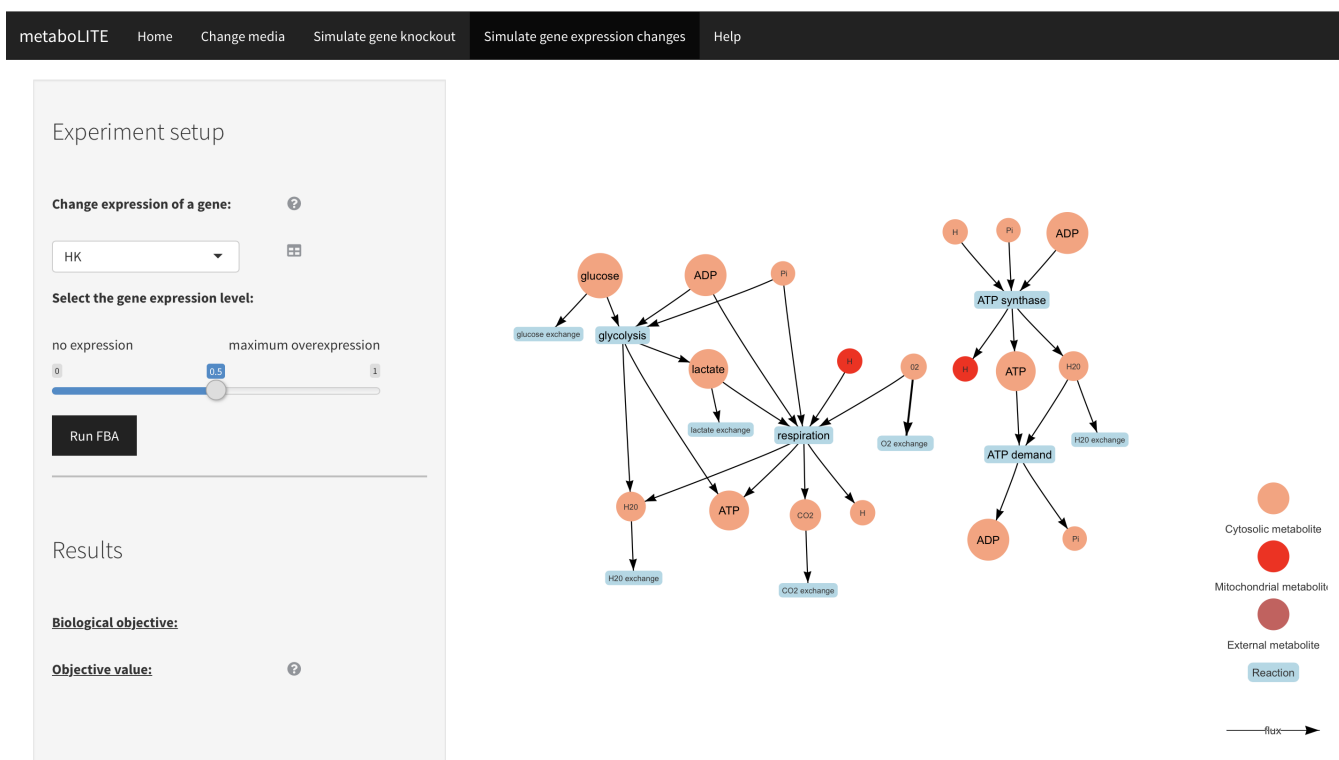
Figure 5: metaboLITE screen for integrating transcriptomic data.

is collected from a biological condition and the condition-specific model will be reduced to contain only the metabolic functions for which there is expression data to support that function. Many different algorithms are used to integrate expression data (*e.g.* (Jensen and Papin 2011; Becker and Palsson 2008; Zur, Ruppin, and Shlomi 2010)); each handles missing data, expression thresholding, and isozymes differently. However, all approaches are similar in intent: if a gene is expressed, the reaction that is catalyzed by the associated enzyme should be in the condition-specific model. If a gene is not expressed, the associated reactions are less likely to be used in that condition. This approach has worked well to represent the metabolism of tissue-specific cell types (*e.g.* (Shlomi et al. 2008)) or particular phenotypes like drug resistant pathogens (*e.g.* (Carey, Papin, and Guler 2017)).

Transcriptomic data allows a researcher to look at the expression of all enzymes at once, but single-target studies can also be integrated into these models. If one enzyme or gene is over or underexpressed, the associated reaction can be appropriated constrained to represent this phenotype. To do this, we constrain the maximum and/or minimum flux through a reaction; this is a constrain on the bounds of the reaction (*Box 2*). For this simulation, the user can pick a gene from a drop down menu, and set the expression level on a slider bar (left, **Figure 5**). The app performs FBA given the new constraint before visualizing and printing the results (right, **Figure 5**).

Conclusion

Here, we present an approach to communicate modeling concepts to aspiring computational biologists, inspired by both a simple metabolic model (iSIM) and the questions we often received when presenting our research to non-modelers. By using a simple reconstruction and web-based interface, we can demonstrate modeling concepts without the use of programming. By decoupling the acts of learning the technical concepts and learning to program, more researchers can be exposed to genome-scale metabolic modeling enabling them to evaluate the utility of this modeling framework for their own biological questions.

metaboLITE

Michal Stolarczyk mjs5kd@virginia.edu

What is this?

metaboLITE is an R shiny application that aids the basic understanding of concepts under genome-scale metabolic modeling. It implements two simple toy models that make it easy to learn the contepts of genome-scale metabolic modeling:

- *iSIM* central energy metabolism
- *E.coli* core metabolism

Functionalities

Visualize the model

- **Network layout visualization**, which is intended to graphically present the interconnections between metabolites and reactions in the genome-scale metabolic reconstruction (GENRE) [1]. Nodes of the graph represent metabolites and reactions whereas edges - connections between them. The network is functionally divided into two parts: ATP production and metabolism and glucose metabolism. The main reactions and metabolites are denoted by a bigger font size. In order to simulate the model growth a Flux Balance Analysis (FBA) simulation is performed [2].

- **Network stoichiometry visualization**, which is intended to depict the stoichiometric coefficients in view of the aforesaid network representation of the genome-scale metabolic reconstruction. The width of the edges correspond to the stoichiometic coefficients that determine ratios of each metabolite in the reactions. Additionally, the coefficients are presented in the table on the left.

Perform simple experiments with just a click of a button

KO reaction tab

- **Reaction knockouts (KOs) impact visualization**, which is intended to present the influence of the specific reaction KOs on genome-scale metabolic reconstruction architecture and fluxes in the model. By means of the FBA this functionality can be used to detect either essential (ones that removed render the model carry 0 flux through the objective function; objective value equals 0) or nonessential (when removed do not influence the effective flux value through the objective function) reactions [3]. Additionally, knockouts can simulate the complete enzyme inhibition that catalyzes the reaction being knocked out.

Change media tab

- **Media changes impact visualization**, which is intended to depict the influence of growth media changes on the model growth and fluxes through reactions. The media changes are performed by constraining the exchange reactions in the model during the FBA simulation. For example in order to check the influence of oxygen shortage on the model growth one needs to lower the upper (and lower) flux bound(s) of the oxygen exchange reaction.

Simulate gene expression changes tab

- **Gene expression influenece visualization**, which is intended to depict the impact of the pseudo-gene expression changes in the model, which directly influences the flux that is carried by the reaction catalyzed by the enzyme encoded by the gene in question. It is indented to give the user an idea of how the algorithms for gene expression integration influence the fluxes in in the model.

Model representation

The models are represented as a graph here. The nodes of the graph serve as metabolites and reactions in the form of orange/red circles and blue rectangles, whereas the edges of the graph define the relationship between them.

The converstion of the *iSIM* model is presented in the image below:
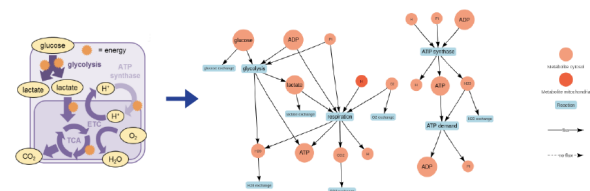
How to cite

TBD

References

1. Thiele, I. & Palsson, B. <98>. A protocol for generating a high-quality genome-scale metabolic reconstruction. Nature Protocols 5, 93<80><93>121 (2010).

2. Orth, J. D., Thiele, I. & Palsson, B. <98>. What is flux balance analysis? Nat Biotechnol 28, 245<80><93>248 (2010).

3. Garca Snchez, C. E. & Torres Sez, R. G. Comparison and analysis of objective functions in flux balance analysis. Biotechnol Progress 30, 985<80><93>991 (2014).

Figure 6: A help page is provided in the app (shown) and on its website

Supplementary information

Supplementary Video

Link to the sample video: https://www.dropbox.com/s/hmnfsebz79lfl4h/changeMedia.mov?dl=0

Installation

Users can download metaboLITE online; instructions for all operating systems (Mac, Windows, and Linux) are found at michalstolarczyk.github.io/metaboLITE.

Installation takes less than 10 minutes, depending on the user's internet connection. First, the user will download Docker and Kitematic, two freely available packages that bypass the manual installation of the app's dependencies (*i.e.* the software that runs 'under the hood' of metaboLITE). Kitematic will open as an application and users search for the metaboLITE by name. metaboLITE will then open a web browser window. A help page and list of relevant references for users are both on the website and in the app itself (**Supplemental Figure 6**).

Supplementary code

COnstraint-**B**ased **R**econstruction and **A**nalysis (or, COBRA) is an open-source software package used by metaboLITE to simulate the biological experiments. It is implemented in both MATLAB and Python programming languages and maintained by the openCOBRA community. Consider the following code examples (in python) to recreate the simulations presented in metaboLITE.

For more information, see the documentation for either CobraToolbox (the MATLAB software package) or CobraPy (the python software package), both at opencobra.github.io.

Loading a model

```python
import cobra
# working directory should be "path/to/model"
model = cobra.io.read_sbml_model("iSIM.xml")
# look at the reactions by reaction id
print([rxn.id for rxn in model.reactions])
# look at the reactions by reaction formula
print([rxn.reaction for rxn in model.reactions])
# look at the metabolites by name
print([met.name for met in model.metabolites])
# look at the genes
print([gene.id for gene in model.genes])
```

Change media

```python
import cobra
# working directory should be "path/to/model"
model = cobra.io.read_sbml_model("iSIM.xml")
# To change the media composition, first pick a metabolite
met_id = 'm04c'
# Look at the reactions associated with this metabolite
print(model.metabolites.get_by_id(met_id).reactions)
# Look at the bounds of the exchange reaction for this metabolite
rxn = model.reactions.get_by_id('E1')
print(rxn.bounds)
# Change upper bound to remove it from media
rxn.lower_bound = 0.
# Run FBA - simulate growth
result = model.optimize()
# Get growth rate
growth = result.f
```

Simulate gene knockouts

```python
import cobra
# working directory should be "path/to/model"
model = cobra.io.read_sbml_model("iSIM.xml")
# Pick a gene to knockout
gene_id = 'LDH'
```

```
# Knockout gene
cobra.manipulation.delete_model_genes(model, [gene_id])
# Run FBA - simulate growth
result = model.optimize()
# Get growth rate
growth = result.f
```

Integrate transcriptomic data

```
import cobra
# working directory should be "path/to/model"
model = cobra.io.read_sbml_model("iSIM.xml")
# Let's say you know gene 'LDH' is downregulated in a condition
gene_id = 'LDH'
# Decrease expression (max and min flux) by 50%
rxn.lower_bound = 0.5*rxn.lower_bound
rxn.upper_bound = 0.5*rxn.upper_bound
# Run FBA - simulate growth
result = model.optimize()
# Get growth rate
growth = result.f
```

Acknowledgments

Author contributions

MAC conceived of the project. MS coded metaboLITE. MAC wrote the manuscript. All authors provided feedback on the app design, edited manuscript drafts, and approved the manuscript.

Funding

References

Becker, Scott A, and Bernhard O Palsson. 2008. "Context-Specific Metabolic Networks Are Consistent with Experiments." *PLoS Comput. Biol.* 4 (5): e1000082. https://doi.org/10.1371/journal.pcbi.1000082.

Carey, Maureen A, Jason A Papin, and Jennifer L Guler. 2017. "Novel Plasmodium Falciparum Metabolic Network Reconstruction Identifies Shifts Associated with Clinical Antimalarial Resistance." *BMC Genomics* 18 (1): 543. https://doi.org/10.1186/s12864-017-3905-1.

Jensen, Paul A, and Jason A Papin. 2011. "Functional Integration of a Metabolic Network Model and Expression Data Without Arbitrary Thresholding." *Bioinformatics* 27 (4). academic.oup.com: 541–47. https://doi.org/10.1093/bioinformatics/btq702.

Shlomi, Tomer, Moran N Cabili, Markus J Herrgård, Bernhard Ø Palsson, and Eytan Ruppin. 2008. "Network-Based Prediction of Human Tissue-Specific Metabolism." *Nat. Biotechnol.* 26 (9): 1003–10. https://doi.org/10.1038/nbt. 1487.

Zur, Hadas, Eytan Ruppin, and Tomer Shlomi. 2010. "iMAT: An Integrative Metabolic Analysis Tool." *Bioinformatics* 26 (24): 3140–2. https://doi.org/10.1093/bioinformatics/btq602.