SAMMI Documentation

Release 0.0.1

Andre Schultz

Apr 07, 2020

Contents

Welcome to SAMMI, the Semi-Automated Metabolic Map Illustrator. SAMMI is a browser based tool developed for the visualization of metabolic networks. In SAMMI, metabolic networks are illustrated by directed bipartite graphs, where reactions and metabolites represent two distinct groups of nodes. Substrate metabolites connect into reaction nodes, which in turn connect to product metabolites.

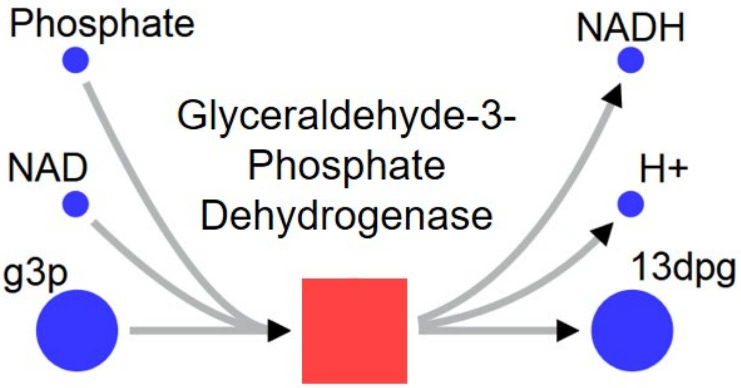


Fig. 1: Visual illustration of a metabolic reaction as represented by a bipartite graph.

SAMMI renders metabolic maps as interactive, constantly updated force-directed graphs, allowing for efficient automated node positioning. The tool has also been developed to allow a wide array of node editing and positioning functionalities, as well as easy graph parsing and navigation. SAMMI can be used for both context-specific visualizations, where users can upload their own reaction and/or metabolite data in order to visualize the context of their own network, or to generate visually appealing models that can be easily shared among SAMMI users.

SAMMI has been implemented as a standalone tool available at www.sammitool.com, and plugins have been developed for COBRA in MATLAB and COBRApy in Python.

CHAPTER 1

Introductory video

The following video provides a quick introduction to some of the main functionalities of SAMMI.

CHAPTER 2

Documentation

The following documentation describes all the tools available in SAMMI.

2.1 Loading a Map

Metabolic models annotated in the Systems Biology Markup Language (SBML) (Hucka et. al.) and BioPAX (Demir et. al.) formats are available in a number of different databases. These databases include, but are not limited to BiGG, MetExplore, Human Metabolic Atlas, Virtual Metabolic Human, BioModels, PathBank, and MetaCyc. These formats are used in the annotation of constraint-based metabolic reconstructions (O'Brien et. al.), a popular platforms for modeling metabolic networks. SAMMI accepts as input both SBML and BioPAX annotated models, and can also load metabolic networks annotated from KEGG. This section explains how to initially load these models.

2.1.1 Load SBML Model

To load a single map for the entire reconstruction, Click on **Load Single Model** at the home screen and select the metabolic reconstruction file.

This option will load every reaction and metabolite in the reconstruction. This functionality is recommended only for relatively small models, as large networks will yield slow-rendering and convoluted maps. For large reconstructions, users can parse the model into multiple subgraphs during the upload process. In order to do so,click on **Load Model to Parse** at the home screen. This will bring you to a second menu where you can select whether to parse the graph based on model annotation or user-defined input. To generate a map based on model data, select either **Reactions** or **Metabolites** from the dropdown menu. This will populate a new dropdown menu with annotation categories from the model. Select a category and SAMMI will build a subgraph for each value under that data type. For instance, select **Reactions** then **Subsystem** to make a subgraph for each subsystem defined in the reconstruction, or select **Metabolite** then **Compartment** to make a map for each cellular compartment.

Parsing a model through reaction annotation can lead to the inclusion of metabolites in multiple subgraphs, but no more than one copy of each metabolite is initially loaded in each subgraph. If parsing a model through metabolite annotation, all reactions containing a metabolite with that annotation will be included in each subgraph. This can lead to the inclusion of metabolites with a different annotation in the subgraph. For instance, if parsing a reconstruction

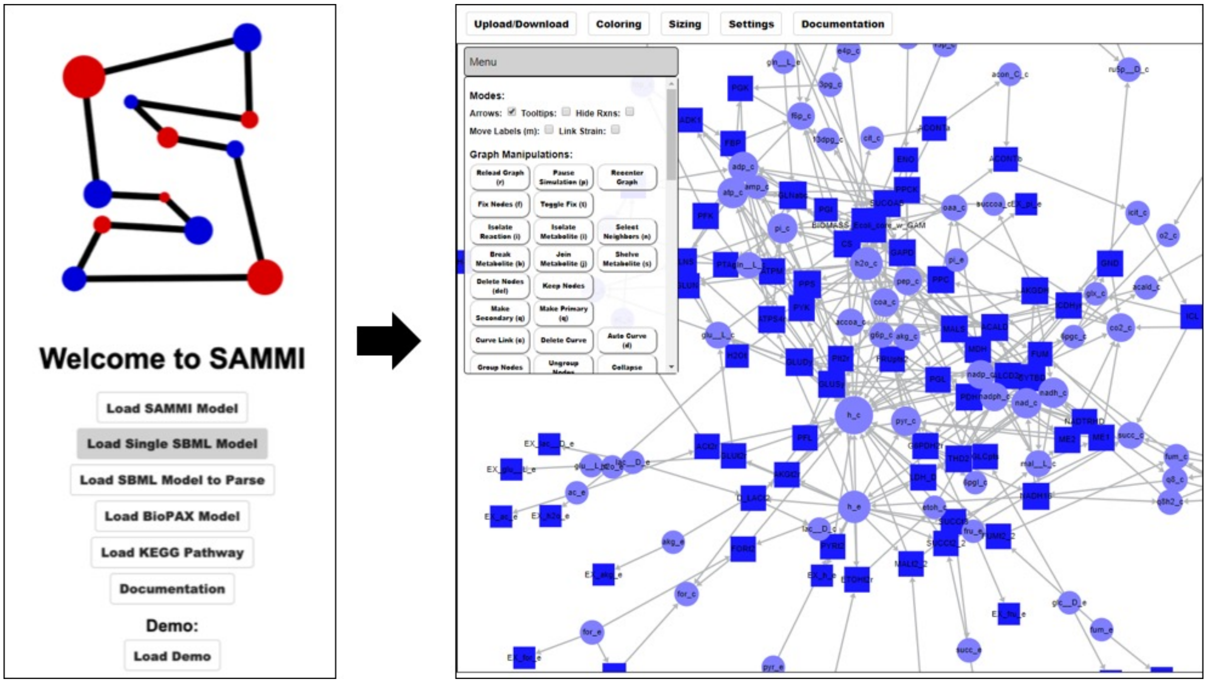
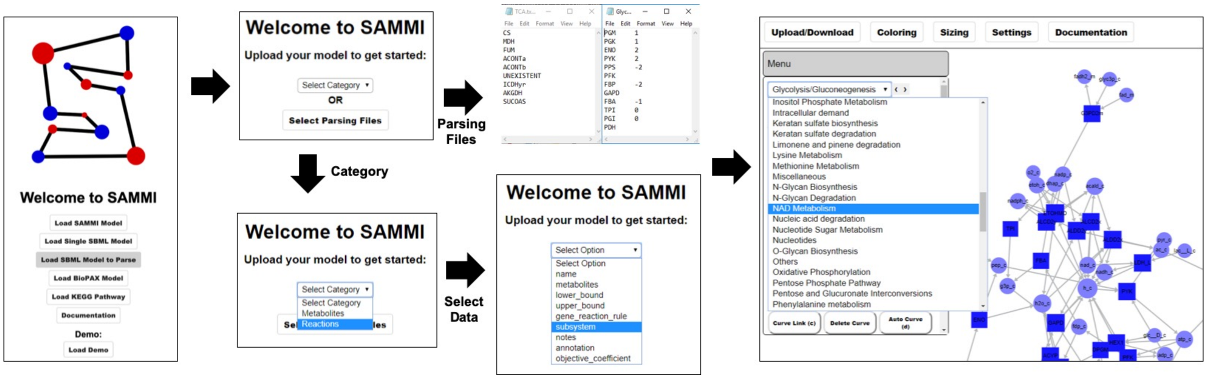


Fig. 1: In order to load the entire reconstruction on a single map choose *Load Single Model*.

based on metabolite compartments, the mitochondria compartment will also include transport reactions through the mitochondrial membrane, which in turn will lead to the inclusion of cytosolic metabolites.

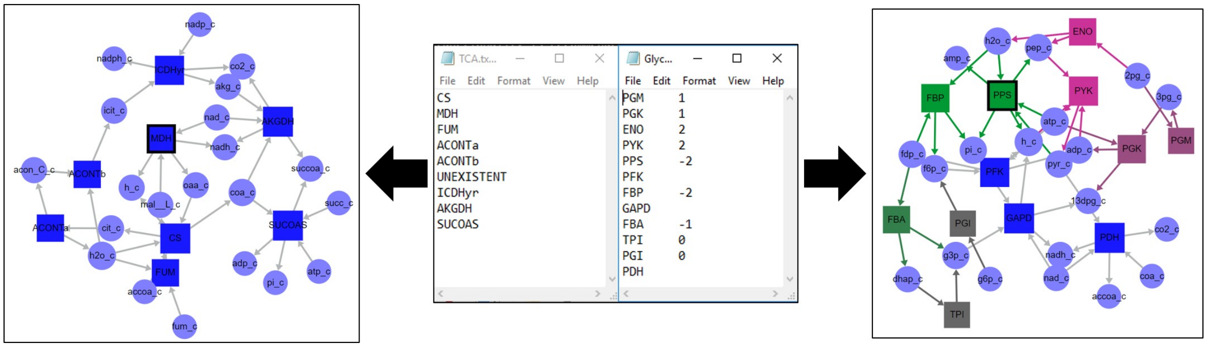


```
CS
MDH
FUM
ACONTa
ACONTb
UNEXISTENT
ICDHyr
AKGDH
SUCOAS
```

```
PGM     1
PGK     1
ENO     2
PYK     2
PPS     -2
PFK
FBP     -2
GAPD
FBA     -1
TPI     0
PGI     0
PDH
```

Fig. 2: Models can be parsed in two different ways. Either using reaction or metabolite data annotated in the reconstruction (bottom) or using user-defined lists of reaction IDs.

The second option is to parse the model based on user-defined inputs. To use this option, users should define any number of tab-delimited text files with two columns: reaction IDs and optional associated numerical values. No header should be included. After defining the files and selecting **Load Model to Parse**, click on **Select Parsing Files** and select all the desired files. SAMMI will make a subgraph for each file selected. Here, reactions can be included in multiple subgraphs or no subgraph at all. Reactions with an associated value will retain that value as reaction color data. Reactions with no associate value will be included in the model but will have no associated data.



```
CS
MDH
FUM
ACONTa
ACONTb
UNEXISTENT
ICDHyr
AKGDH
SUCOAS
```

```
PGM     1
PGK     1
ENO     2
PYK     2
PPS     -2
PFK
FBP     -2
GAPD
FBA     -1
TPI     0
PGI     0
PDH
```

Fig. 3: Two valid examples of model parsing files. The file on the left will load a map including only the reactions in that file, with no associated data. The line UNEXISTENT will be ignored since no such reaction is present in the model. The file on the right will load associated reaction data where those values are defined.

2.1.2 Load BioPAX Model

In oder to load BioPAX models select *Load BioPAX Model* in the SAMMI home screen. For this options users can select multiple files, each of which are loaded as a different subgraph. File names will be used to name each subgraph, although these names can be changed after models are loaded.

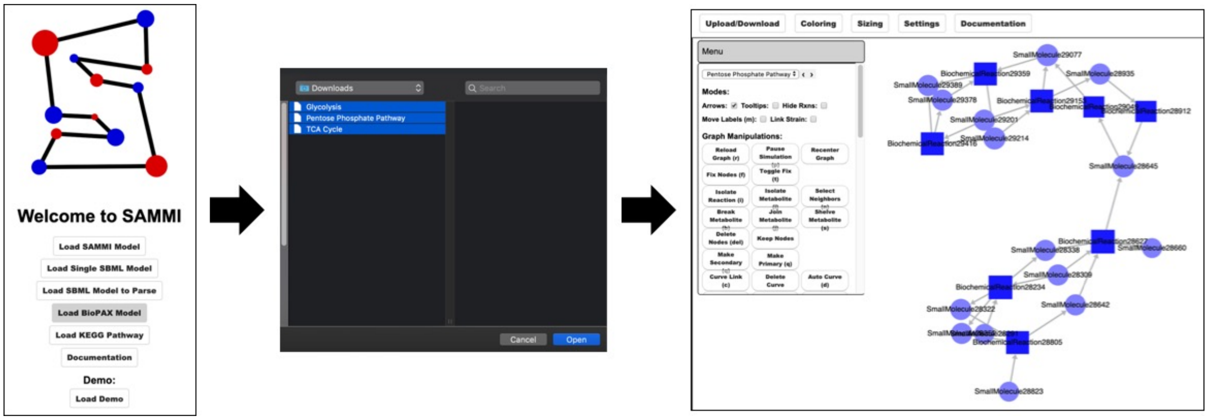


Fig. 4: BioPAX models can be loaded multiple at a time, and each loaded model will be rendered as a subgraph.

2.1.3 Load KEGG Pathway

The metabolic pathways for over 5,000 organisms have been made readily available in SAMMI. These pathways were obtained from KEGG in January of 2019 and contain associated metabolite and reaction annotations. To load a metabolic map from KEGG click on **Load KEGG Pathway** in the home screen. Next, type the name of the desired organism and matching names will be autocompleted. Organisms can be searched by binomial nomenclature, informal name, and organism KEGG IDs. Select the desired organism from the autocomplete list and hit enter to load the desired map.

2.2 SAMMI Interface

Once a model is loaded, the SAMMI working interface will be displayed with the uploaded graph. The SAMMI interface is composed largely of four parts:

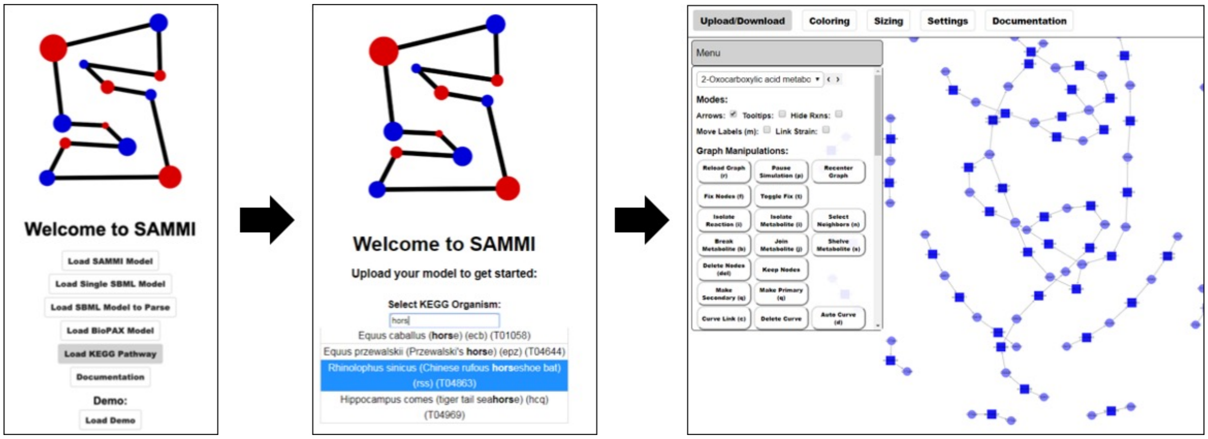
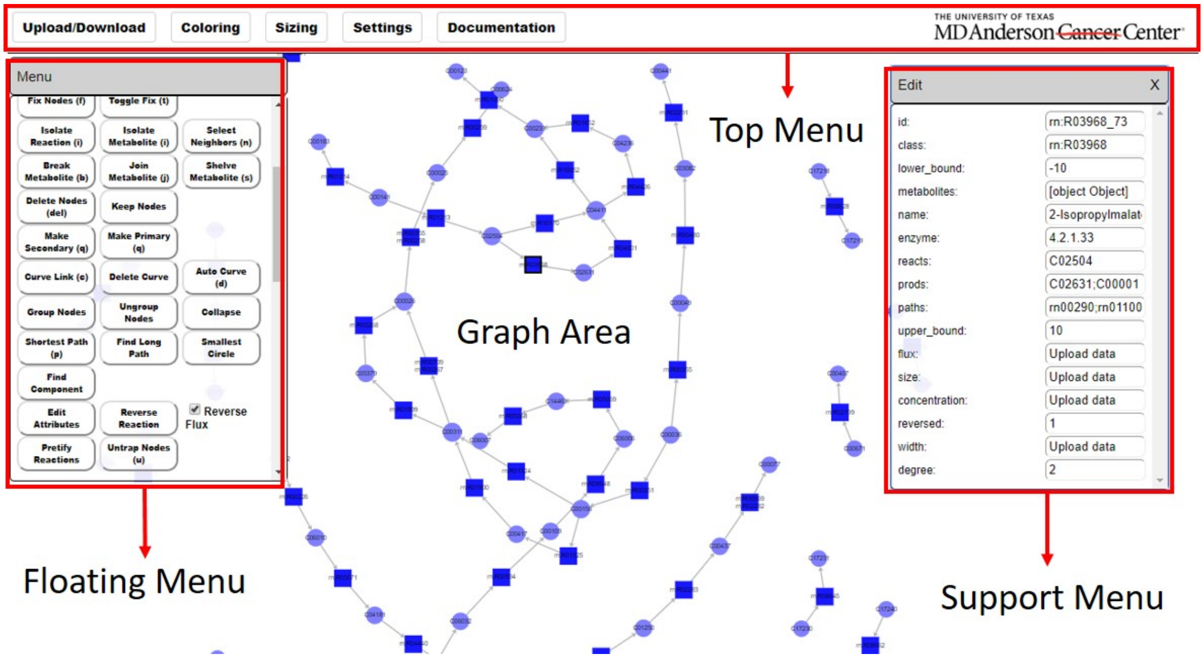


Fig. 5: To upload a metabolic map from KEGG choose *Load KEGG Pathway* from the home screen. Next, type the name of the organism, choose one of the autocomplete options, and hit *Enter*. This option will load all metabolic pathways available in KEGG for the selected organism.



id: rn:R03968_73
class: rn:R03968
lower_bound: -10
metabolites: [object Object]
name: 2-Isopropylmalat
enzyme: 4.2.1.33
reacts: C02504
prods: C02631;C00001
paths: rn00290;rn01100
upper_bound: 10
flux: Upload data
size: Upload data
concentration: Upload data
reversed: 1
width: Upload data
degree: 2

Fig. 6: The SAMMI working interface is composed of four main parts: (1) Top Menu, (2) Graph Area, (3) Floating Menu, and (4) Support Menu.

- **Top Menu**: The top menu bar is displayed across the top of the browser window. This menu has multiple tabs with a wide variety of functionalities which will be covered throughout this documentation
- **Graph Area**: This is the area where the nodes and edges of your current graph are displayed. Nodes in this area can be dragged and modified, shapes and text annotations can be added, and the area can be zoomed in and out and panned.
- **Floating Menu**: The floating menu is constantly displayed inside the graph area. This menu can be dragged to different parts of the graph area using the top title bar or empty areas of the window. This menu displays a series of functionalities that can be used to modify and navigate the current graph. See the *Floating Menu* section of the documentation for or a full description of all functionalities available in this part of the interface.
- **Supporting Menu**: The supporting menu is an additional menu window, similar to the floating menu, which is loaded for specific functionalities such as combining subgraphs, changing texts, or altering node data. Like the floating menu this window can be dragged around the map area, but it can also be closed by clicking the top right corner of the window.

2.3 Graph Settings

Several visual and functional graph properties can be changed as desired to alter the graph layout. While many of these functionalities will be covered later in the documentation, this section will highlight some of the standard simulation and visualization options.

2.3.1 Coloring

Node and edge coloring can be changed under the **Coloring** tab in the top menu. For a full description of these option please refer to the *Node and Edge Coloring* section of the documentation. Here we will provide a summarized description of some of the option:

- **Metabolite**: Standard color of floating metabolite nodes with no associated data.
- **Reaction**: Standard color of floating reaction nodes with no associated data.
- **Fixed Metabolite**: Standard color of fixed metabolite nodes with no associated data.
- **Fixed Reaction**: Standard color of fixed reaction nodes with no associated data.
- **Edge**: Standard color of edges with no associated data.
- **Metabolites with size data**: Color of metabolite nodes with size data but no color data. If node size data is present, but no color data, setting this option to a different color allows users to identify which nodes are a certain size due to associated data, and which ones are the default node size.
- **Reaction with size data**: Color of reaction nodes with size data but no color data. Similar to previous option.
- **Links with width data**: Color of edges with width data but no color data. Similar to previous two options.
- **Position node color**: Some SAMMI functionalities add a temporary reference node to the graph. This option defines this node color. If users wish to use the default reference node color to color metabolite or reaction nodes, this option can be changed for easy identification of reaction nodes.
- **Link Strain Color**: The link strain option in the floating menu colors graph edges by their length. This option defines the color of longer edges.

2.3.2 Sizing

Node, edge, and text sizing can be changed under the **Sizing** tab in the top menu. For a full description of these option please refer to the *Node and Edge Sizing* section of the documentation. Here we will provide a summarized description of some of the option:

- **Label Size**: Size of text of node labels.
- **Added Text Size**: Text annotations can be added to the graph. This option defines the size of the font in these annotations.
- **Link Width**: Default width of edges with no associated data.
- **Node Size**: Default size of nodes with no associated data.
- **Reference Node Size**: Size of reference node. Reference nodes are added for position reference during the execution of some functionalities, such as node positioning and edge curving. If users want to change large areas of the graph, the default node size might be too small to click if the graph is zoomed out. This allows users to increase the size of the reference node.
- **Arrow Size**: Size of edge arrows.

2.3.3 Settings

Several parameters used in the force-directed graph simulation, as well as other SAMMI functionalities, can be tailored for better visualization and execution purposes. These parameters can be changed under the **Settings** tab in the top menu. Node labeling can also be changed here using model data.

- **Link Repulsion Strength**: Force-directing layout aims to make each edge a given length defined by this parameter. Higher values will lead to overall longer edges, and lower values to overall smaller edges.
- **Node Collision Buffer**: The graph layout simulation aims to make the nodes not overlap with each other. Setting this value to a positive number will add an additional space between nodes. This will not increase the repulsion between each node, but simply add space between nodes that would otherwise be touching or overlapping.
- **Node Repulsion Strength**: Increase the strength with which nodes repeal each other. Higher values will lead to more spread-out graphs, while lower values will lead to more compact maps.
- **Inertia**: Speed with which the simulation takes place. Lower values will lead the simulation to respond very quickly to node dragging and rearranging, while high value will make the force-directing take place very slowly.
- **Center Strength**: In order for graph components to not drift too far from each other, all nodes are attracted to the center of the graph with a force relative to this parameter. Setting this value to zero will allow any independent components to drift far away from our view, and setting this parameter to high values will bring all nodes to the center of the graph.
- **Secondary Strength**: Metabolite nodes tagged as secondary are not attracted to the center of the graph, but to a point calculated relative to their associated reaction. This parameter defines the strength of this attraction.
- **Path Execution Timeout (s)**: One of the options in the floating menu calculates a long path in the graph between two nodes. The calculation of the longest path between two nodes in a graph is an NP hard problem, however. In SAMMI, this long path is calculated by a modified Breadth First Search algorithm that finds increasingly longer paths. While this algorithm does have a stopping condition, it can take an extremely long time to reach it, and it can lead to memory overload in the browser if allowed to run for a long time in large, highly connected graphs. This parameter defines in seconds how long this algorithm can run for. Higher values can return longer paths, but will take longer to run.
- **Center Size**: This option adds a black dot to the center of the graph with size relative to this parameter. This point is added solely for the user's reference, and does not affect the force directed layout. Make this parameter a positive value to visualize the center point to which nodes are attracted to in the force-directing.

- **Pretify Size**: The pretify option organizes orphan reactions into an aesthetically pleasing format. This option defines the size of the area in which they are organized. Higher values will lead reactions to be arranged in a more spread out manner.
- **Naming**: These options are populated based on data available in the uploaded reconstruction. Changing this option will re-label the nodes in the graph. The search function will search for nodes based on the values in the field defined here. That is, if nodes are labeled by ID, the search function will search the node IDs, if they are labels by name, it will search for names.
- **Reversibility**: Defines how the edge arrows are drawn to indicate reversibility. Reactions are considered reversible if they have a lower bound value of less than zero. Different metabolic reconstructions may use different field names to defined reaction lower bound (e.g. *lower_bound* or simply *lb*). To define the field that should be used for reaction lower bound change the value of the *LB Field* field under the *Upload/Download* tab.
  - **None**: Indicates no reversibility. Arrows are drawn to indicate the directionality in which the reaction is annotated.
  - **Both Ways**: Draws arrows in both ends of the edges related to reversible reactions. While this option indicates reversibility, it does not indicate the direction in which the reaction is annotated.
  - **Diamond Arrows**: Draws a diamond shape at the target end only of edges related to reversible reactions. This option allows for both the identification of reversibility and directionality of annotation.

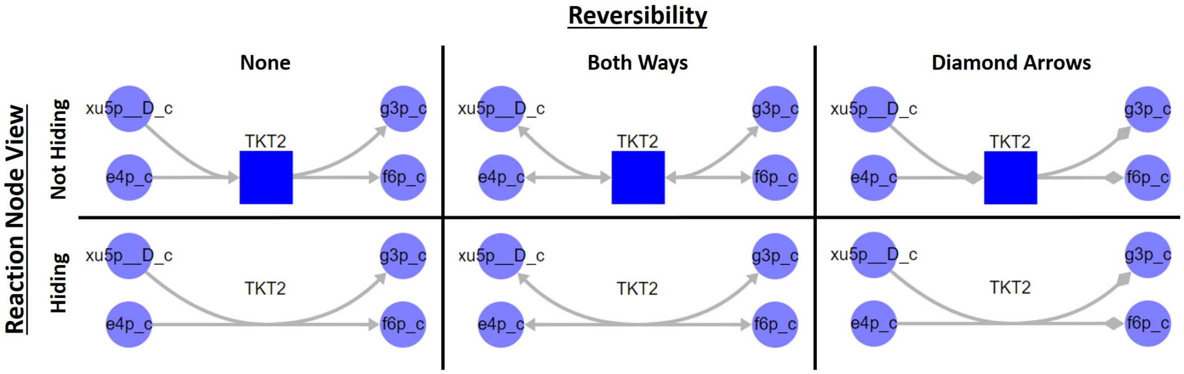


Fig. 7: Six reaction visualization formats. Columns indicate the reversibility visualization type, available in the top menu *Settings* tab, and rows indicate the visualization of reaction nodes, available in the floating menu.

- **Node Shape**: Defines whether nodes are represented as circles or squares. By default, reactions are usually represented as squares and metabolites as circles in order to easily differentiate between the two node types.

2.4 Associated Graph Data

Reaction nodes can be associated with three types of data:

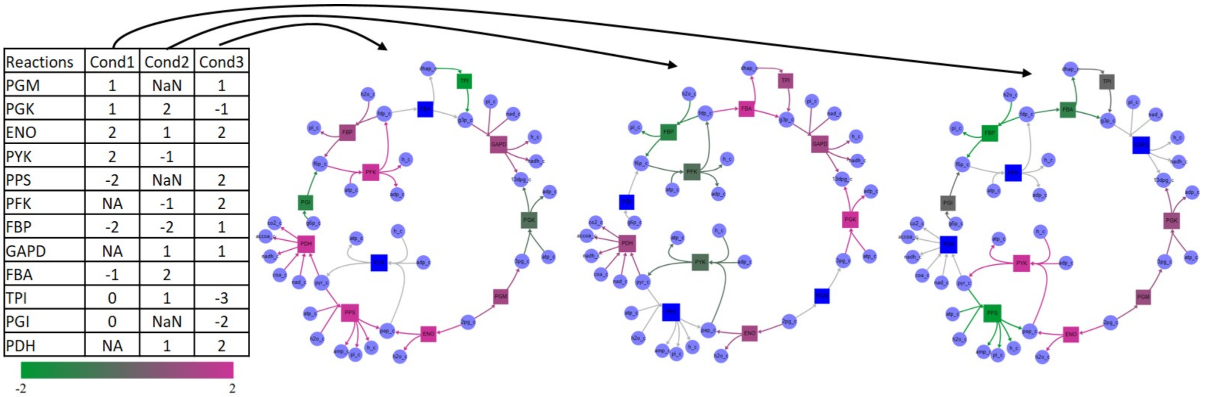
- **Color Data**: Referred to here as *flux* data, these values are used to color the nodes and the edges associated with the given reaction. When reversing reaction directionality, these values may also be reversed (See *floating menu* section). To upload reaction color data click on *Load Reaction Color Data* in the *Upload/Download* tab in the top menu.
- **Size Data**: Defines the size of the reaction node when present. To upload reaction size data click on *Load Reaction Size Data* in the *Upload/Download* tab in the top menu.
- **Edge Width Data**: Defines the thickness of the edges associated with the given reaction node. To upload edge width data click on *Load Link Width Data* in the *Upload/Download* tab in the top menu.

Metabolite nodes can be associated with two types of data:

- **Color Data**: Referred to here as *concentration* data, these values are used to color metabolite nodes when present. To upload metabolite color data click on *Load Metabolite Color Data* in the *Upload/Download* tab in the top menu.
- **Size Data**: Defines the size of the metabolite node when present. To upload metabolite size data click on *Load Metabolite Size Data* in the *Upload/Download* tab in the top menu.

Color and size scales can be modified following the data upload, and this functionality will be explained in further detail in the following sections. Uploaded data propagates to all nodes in all subgraphs in the given map. That is, if a reaction or metabolite in the file loaded are present in a subgraph different than the one currently displayed, that data will still be assigned to that node. To view the data associated with a given node, turn on the *tooltip* function on the floating menu and hover over the node. Alternatively, if you wish to change any data after it has been uploaded, use the *Edit Attributes* function also in the floating menu. These changes will not be propagated, however. That is, changing the concentration value of pyruvate in the current graph does not change it in other graphs. To remove any data type simply click the **Remove** button next to the given data type in the *Upload/Download* tab in the top menu.

Data files to be uploaded should be formatted as tab delimited text files. The first column should define reaction or metabolite model IDs. Every subsequent column will define values for a different condition. Non-numeric or empty fields are skipped and no data is associated with the node in that condition. Size values loaded are taken as absolute value. If the number of non-empty entries in the header row matches the number of conditions, those are taken to be condition names. If they define one extra entry, it is taken as the header for the ID column and the first entry is discarded. Once data is loaded, a dropdown menus is loaded in the floating menu window where users can navigate the different conditions. The following figure exemplifies a sample data file loaded as reaction color data.



Reactions	Cond1	Cond2	Cond3
PGM	1	NaN	1
PGK	1	2	-1
ENO	2	1	2
PYK	2	-1	
PPS	-2	NaN	2
PFK	NA	-1	2
FBP	-2	-2	1
GAPD	NA	1	1
FBA	-1	2	
TPI	0	1	-3
PGI	0	NaN	-2
PDH	NA	1	2

Fig. 8: Example of a data file loaded as reaction color data. The file is a tab-delimited text file where the first column defines reaction IDs and every subsequent column defines different conditions. Non-numeric and empty fields are ignored and no data is associated with that node in that condition.

Edge width files are structured similarly to the files described above, and should be defined using reaction IDs. All edges associated with the same reaction will have the same width value. Width values are also loaded as absolute values.

2.5 Node and Edge Coloring

Node and edge coloring properties can be managed in the **Coloring** tab in the top menu. Colorscales are defined by a minimum and a maximum value, and optional breaks can be added using the **Add Break** button. Minimum and maximum values are first defined by the minimum and maximum values in the uploaded data, and additional break values must be between these two values. After upload, minimum, maximum, and break values, as well as associated color, can be modified in the **Coloring** tab. Nodes with associated data below the minimum are assigned the minimum color, and nodes with associate value above the maximum are assigned the maximum color.

Colorscales are defined for each data condition loaded (i.e. each column in the data file). This allows users to appropriately color conditions with different meaning (e.g. flux and gene expression) or significantly different ranges of value. To make all reaction or metabolite colorscales the same, simply navigate to the condition with the desired colorscale and click on the **Make Color Scale Global** button under the desired node type.

Nodes with no associated color data but that are given size data are colored according to the **Reaction/Metabolite with size data** color specified in the coloring menu. This allows users to know which nodes are sized according to the loaded data and which nodes are the default size. If no color or size data is present, the nodes are colored according to whether the node is fixed or floating. These options are defined at the top of the coloring menu. The following figure summarizes the decision making process in coloring nodes in the graph.

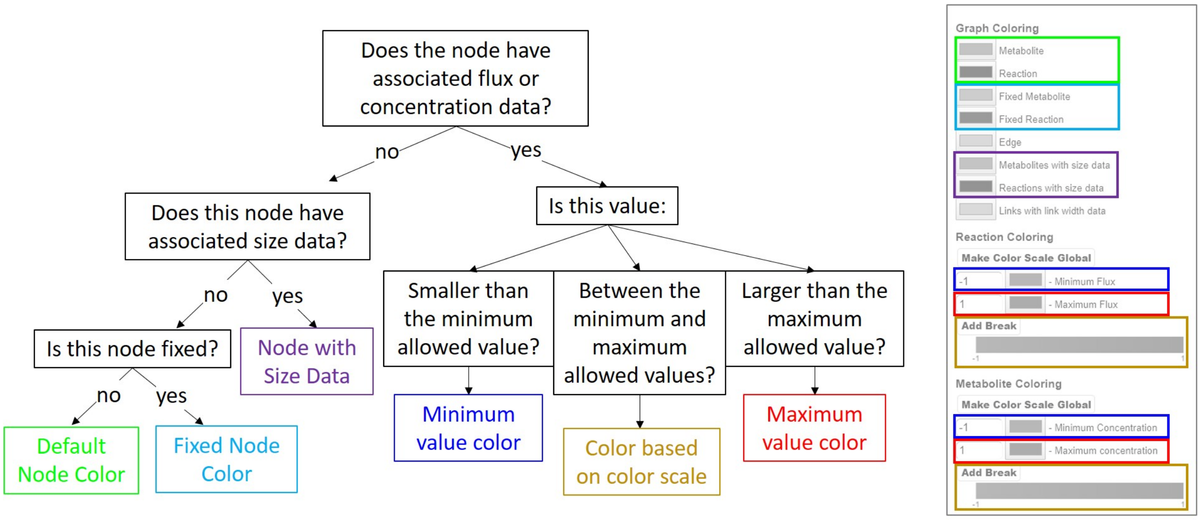


Fig. 9: Flowchart summarizing the decision making process for node coloring. The color of the box at the end of each branch can be matched with the box positioned over the coloring menu on the right, indicating the color of the node. Nodes with color data are colored according to the colorscale. Nodes with no color data but with size data are colored according to the given option, and nodes with no associated data are colored according to whether the node is fixed or not.

Edges can also be colored according to length. This functionality can help identify which edges are being most elongated and are therefore convoluting the map. This functionality can be activated by checking the **Link Stain** box inside the floating menu, which will color all links using a uniform colorscale ranging from the default *Edge* color to the *Link Strain Color*, both defined under the coloring menu. If this functionality is not activated, edges are colored the same color as their associated reaction node when that node has associated flux data. If the associated node has no flux data but the edge does have width data, the edge is colored according to the *Links with link width data* option. Otherwise, links are colored according to the default *Edge* option.

2.6 Node and Edge Sizing

If a map has no associated size data in any of its subgraphs, nodes are sized relative to their degree and the scaling factor defined by the *Node size* option in the **Sizing** tab in the top menu. That is, nodes with a higher number of connected edges will be slightly larger. This allows users to quickly identify bottleneck, highly connected metabolites immediately upon loading the model. If any size data is loaded, however, that functionality is removed. Nodes with associated size data will then be sized according to the data, and nodes with no size data are sized according to the default value *Node Size* under the sizing menu.

Edge width will be defined according to associate width data for edges where the data is present, or according to the default *Link Width* under the sizing menu for edges where width data is missing. The default width for edges with no associated width data can be modified using the *Link Width* option in the sizing menu.

The scale of node sizes and edge widths can be defined also under the sizing menu. Each of the three size scales (reaction and metabolite node size and edge width) are defined by three options: *Maximum Size*, *Minimum Size* and *Size Scale*. Nodes and edges with size data below the minimum size will be sized using the minimum value, and nodes and edges with size data above the maximum value will be sized using the maximum value. The size of nodes and edges with associated size data between the minimum and maximum values will be scaled linearly between those two values.

The third parameter *Size Scale* is a multiplying factor that defines the final node size and edge width. That is, large scale parameters will lead to overall large nodes or thicker edges, while low scale parameters will lead to overall smaller nodes and thinner edges. This allows users to size nodes and edges appropriately independently or the magnitude of the values loaded. These scales are set globally for all data conditions loaded.

The final node size and edge thickness displayed depends on the zoom level of the graph. To load a size reference scale, click on the checkbox *View Size Reference* under the sizing menu. This option will load a reference scale on the top left portion of the graph area which will adjust according to the zoom level.

2.7 Mapping Other Types of Omics Data

SAMMI offers the functionality of converting gene or protein expression data into associated reaction data directly within the SAMMi interface. This function can be accessed using the *Load Gene Data* in the *Upload/Download* tab in the top menu. Since gene-reaction rules can be expressed in several different ways between different metabolic models, SAMMI offes a number of different tools to convert gene to reaction data. Once this options is selected the supporting menu window will open with the following options:

- **Gene Expression Rule Field**: This option defines which reaction field is to be used in this data mapping (e.g. grRules).
- **Regular Expression to Split**: This field defines a series of regular expressions that can be used to split the string in the field defined above into substrings. These regular expressions should be separated by a semicolon.
- **Regular Expressions to Remove**: This field defines a series of regular expressions that should be removed from the string in the defined field. These should also be separated by semicolons.
- **Select Mapping Function**: Define the function to be used in mapping the associated gene or protein data.

Once these fields are defined, users can push the **Map** button to select the data file to be mapped. The uploaded file should be similar to the reaction and metabolite tab delimited files, where the top row defines multiple conditions and the first column defines the genes to be mapped. Once the file is loaded a series of steps convert the defined field into a vector of associated gene names. These steps are:

1. Remove any characters or substrings matching any of the regular expressions defined under *Regular Expressions to Remove*.
2. Split the resulting string in any of the regular expressions defined under *Regular Expression to Split*.
3. Trim trailing and leading empty spaces.
4. Take unique elements.

For instance, given the gene-reaction rule *100760573 or (100760573 and 100765276)* from the reaction *MAN1PT* in the iCHOv1 reconstruction(Hefzi et. al.), this string can be processed as:

1. Defining *"\(;\)"* as the expression to remove we can remove the parenthesis leaving *100760573 or 100760573 and 100765276*.

2. Using *“and;or”* as the splitting expression we can split the string into *[“100760573 ‘’,” 100760573 ‘’,” 100765276’‘].*
3. Trimming extra spaces we get *[“100760573’‘,’‘100760573’‘,’‘100765276’‘].*
4. And taking unique entries, finally, we get *[“100760573’‘,’‘100765276’‘].*

Once the final vector of gene entries is obtained, gene names are replaced by uploaded numerical values, genes for which no value was uploaded are removed, and the defined function is used to obtain a final value for the reaction.

2.8 Secondary Metabolites

Many metabolites in metabolic networks, such as water, hydrogen and ATP, are considered “secondary” due to their participation in a large number of reactions and ubiquitous presence in the network. In drawing metabolic networks, these nodes can lead to high connectivity and clutterness in the graph. For that reason, these nodes are often either omitted or repeated several times in the metabolic map. Omitting nodes, however, is most often undesirable since secondary metabolite nodes are important to provide completeness in the network representation. Repeating a node multiple times is most often a more desirable option, and metabolites defined as secondary are often represented by a smaller node to reduce cluttering. This approach, however, can still lead to a significant increase in the number of nodes, which can in turn hinder the process of arranging the nodes for a visually pleasing representation.

While the functionalities of deleting, making multiple copies, and representing secondary metabolites by smaller nodes are all available in SAMMI, an additional functionality of **Shelving** metabolites is also available. By shelving a metabolite, that node is temporarily removed from the graph, but can be re-introduced at a later point. By using this functionality, secondary metabolite nodes can be temporarily removed while primary nodes are arranged in an aesthetically pleasing layout, then reintroduced, split into multiple copies, and labeled as secondary.

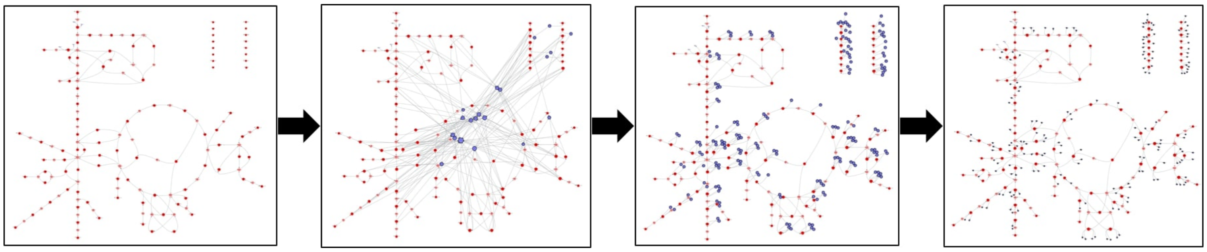


Fig. 10: On the far left, graph with shelved secondary metabolites has been arranged in a visually pleasing layout. Following that, secondary metabolite nodes are re-introduced, one copy of each secondary node is made for each reaction, and the nodes are labeled as secondary.

Metabolites can be shelved one of three different ways:

- *One at a time*: Selected metabolites can be shelved one at a time using the **Shelve Metabolite** functionality in the floating menu, or using the keyboard shortcut **s**. Both of these actions will shelve all selected metabolite nodes in the current graph.
- *From user defined input*: Metabolites can also be shelved in a more high throughput manner by uploading a text file containing a list of regular expressions (one expression per line) using the **Load Metabolites to Shelve** option in the **Upload/Download** tab in the top menu. By using this functionality, all metabolites, in all subgraphs, with class matching any of the uploaded regular expressions will be shelved.
- *From standard List*: Under the **Shelve Standard Metabolites** option in the **Upload/Download** tab, users can also select from a list of metabolites commonly considered to be secondary. Once this option is selected, the supporting menu window opens. First, select the compartment annotation syntax in your model. Next, select all metabolites you wish to shelve and click *Shelve* at the bottom of the window. This option also shelves

metabolites with class matching any of the options in all subgraphs. If you wish to add to to this standard list of secondary metabolites or to the list of compartment annotation syntaxes please contact us.

Metabolites can be returned to the current graph under the **Shelved Metabolites** section of the floating menu.

2.9 Graph Navigation

While the generation multiple subgraphs can lead to more digestible visualizations, and making multiple copies of selected nodes can help produce more visually appealing maps, these representations can hinder the visualization of data continuity. For instance, if a metabolite is produced in a given reaction in a given subgraph, and consumed in another reaction in another subgraph, the continuity of the production and consumption of the given metabolite can be lost in the given visualization. To ameliorate this issue, SAMMI offers two map navigation options.

2.9.1 Node Searching

The search functionality in the floating menu allows users to quickly locate nodes throughout the graph. To search for nodes, input the desired text in the input field and push the *Enter* key. All nodes with current labels matching the inputed text will be selected in the graph. That is, searches are performed using the field currently used to label the nodes.

If multiple subgraphs are present, the support menu will be populated with all other subgraphs containing nodes matching the search, as well as the number of matching nodes. To navigate to a different subgraph, simply click on that subgraph in the support menu. To quickly search for a specific node, hold down the *Ctrl* key and click on the node.

Two functionalities can be turned on and off during searches using the checkboxes under the search input field. The **focus** option hides all nodes not associated with a node matching the search. To view all nodes again, simply click anywhere on the graph area. The second option **RegExp** determines whether the inputted text will be taken as a regular expression or as standard text. If this option is on, any node matching the regular expression inputted will be selected. Otherwise, any node containing the inputted text, not considering capitalization, will be selected.

2.9.2 Highlight Metabolite

To visualize all the reactions where a metabolite is present, select the desired metabolite and click on the **Highlight Metabolite** button in the floating menu. Alternative, use the keyboard shortcut **h**. This option will load a temporary graph where the selected node will be in the center, and all reactions where that node participates will be represented by a connected reaction node. Subgraph names will be appended to the name of each reaction. This function automatically arranges the associated reaction nodes in a circle. Drag the reference node to increase the circle size and view the reaction names clearly. To navigate to a different subgraph, double-click on the reaction node. To return to the current graph, double click on the graph background.

2.10 Downloads

SAMMI maps can be downloaded in a SAMMI specific format, static images, as well as Escher and MetExploreViz compatible formats. All of these options are available in the **Upload/Download** tab in the top menu.

- **Download SAMMI**: Download all subgraphs, as well as node positioning, associated data, colorscales, and simulation parameters of the current SAMMI map. This file can be uploaded using the **Load SAMMI Model** option in the home screen once SAMMI is restarted. Note that map commits will NOT be saved under this option. To do so, please download each commit as a separate file. This is done in order to limit file size.

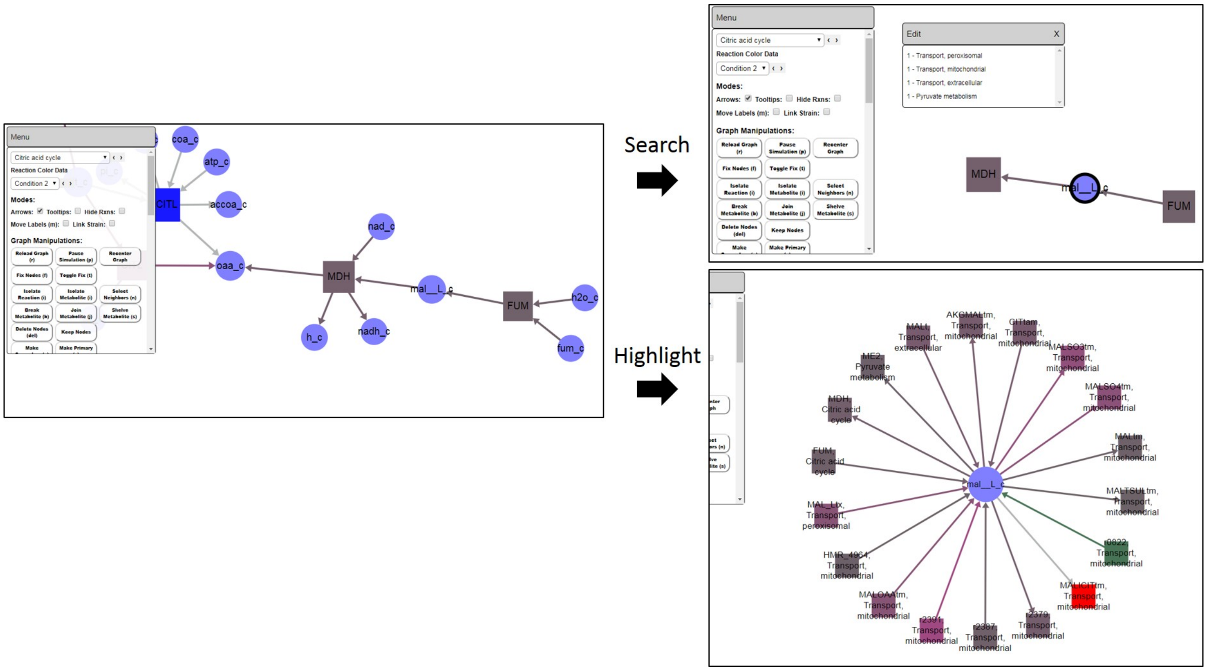

Menu
Citric acid cycle
Reaction Color Data
Condition 2
Modes:
Arrows: ☑ Tooltips: □ Hide Rxns: □
Move Labels (m): □ Link Strain: □
Graph Manipulations:
Reload Graph (r)
Pause Simulation (p)
Recenter Graph
Fix Nodes (f)
Toggle Fix (t)
Isolate Reaction (i)
Isolate Metabolite (i)
Select Neighbors (n)
Break Metabolite (b)
Join Metabolite (j)
Shelve Metabolite (s)
Delete Nodes (del)
Keep Nodes
Make Primary
coa_c
atp_c
CITL
accoa_c
nad_c
oaa_c
MDH
mal__L_c
FUM
h2o_c
h_c
nadh_c
fum_c
Search
Highlight
Edit
1 - Transport, peroxisomal
1 - Transport, mitochondrial
1 - Transport, extracellular
1 - Pyruvate metabolism
MALt, Transport, extracellular
AKGMALtm, Transport, mitochondrial
CITtam, Transport, mitochondrial
MALSO3tm, Transport, mitochondrial
ME2, Pyruvate metabolism
MALSO4tm, Transport, mitochondrial
MDH, Citric acid cycle
MALtm, Transport, mitochondrial
FUM, Citric acid cycle
MALTSULtm, Transport, mitochondrial
MAL_Lx, Transport, peroxisomal
r0822, Transport, mitochondrial
HMR_4964, Transport, mitochondrial
MALICITtm, Transport, mitochondrial
MALOAAtm, Transport, mitochondrial
r2370, Transport, mitochondrial
r2301, Transport, mitochondrial
r2387, Transport, mitochondrial

Fig. 11: Plotting multiple subgraphs and making multiple copies of nodes can hinder data visualization continuity. (Left) Visualization of a metabolite balanced flux distribution assures that a reaction in the model must be producing the malate consumed here by MDH, but this reaction must be present in a different subgraph as it is not represented here. (Top right) Searching a node shows in the supporting menu which other subgraphs that node is also present. (Bottom right) Highlighting a node loads a temporary graph presenting all reaction where that node is present. Double-clicking a reaction node takes you to that subgraph, and double clicking the background takes you back to the original graph.

- **Download Current**: Downloads only the current graph, and not all other subgraphs present in the current map. This single map can also be uploaded under the **Load SAMMI Model** option once SAMMI is restarted.
- **Download ESCHER**: Downloads a JSON file of the current graph compatible with Escher. To view this file in Escher, click on *Load Map* in the Escher home page. Next, select *Load Map JSON* under the *Map* tab and upload the JSON file. To set parameters for this download please refer to the end of this section.
- **Download MetExploreViz**: Download a version of the current graph compatible with MetExploreViz. Once you have a working version of MetExploreViz, under *Load network from* in the home screen, select *JSON file* and uploaded the JSON file. MetExploreViz also offers force-directing layout with an option of fixing nodes. To download a dynamic version of the current graph, check the *Animated* checkbox next to this option. If left unchecked, nodes will be fixed to the position displayed in SAMMI. To set other parameters for this download please refer to the end of this section.
- **Download PNG/PDF**: Downloads a PNG or PDF image of the current graph view. This option will download the graph as it currently displayed on the screen, and not an image of the entire map. If downloading a PDF, the *Download Scales* option will download a second PDF page displaying size and color scales of the current graph.

Some parameters might need to be defined when downloading versions of the map compatible with Escher or MetExploreViz. These parameters are displayed under **Model Download Parameters**, also in the *Upload/Download* tab. These are:

- **Scale**: Scaling factor for node positioning. These different platforms might operate under different ranges of values for node positioning. Define this parameter to expand or contract the network in the next platform. Used by Escher and MetExploreViz downloads.
- **LB Field**: Reactions are determined to be reversible or not using the lower bound value of each reaction. That is, a reaction is considered reversible if it has a negative lower bound, and is considered irreversible otherwise. Different model annotations may define the lower bound by different fields (e.g. *lower_bound* or *lb*). Define in this option the name of the field used to determine the reaction lower bound in the current map. This field will also affect how reaction arrows are drawn in SAMMI. To visualize all node fields use the *tooltip* option (refer to *floating menu* documentation). Used by Escher and MetExploreViz downloads.
- **Name Field**: Name of the field used to define reaction and metabolite names in MetExploreViz. Used by MetExploreViz download only.
- **Pathway Field**: Name of the field used to define reaction pathway in MetExploreViz. Used by MetExploreViz download only.
- **Compartment Field**: Name of the field used to define metabolite compartment in MetExploreViz. Used by MetExploreViz download only.

2.11 Floating Menu

The floating menu defines most of the functionalities available for graph manipulations. This section describes all the options available in the order they appear:

2.11.1 Modes

- **Arrows**: Defines whether or not to add arrows to the ends of edges indicating reaction directionality. Besides the visual component, removing arrows from the map will allow the force-directing to render faster, which might be desirable for larger graphs.
- **Tooltips**: If this option is turned on, hovering over nodes will display a tooltip with all the data related to that node.

- **Hide Rxns**: Hide reaction nodes. This option makes reaction nodes invisible, but they can still be clicked, dragged, and rearranged.
- **Move Labels**: By default, node labels are not clickable in order to make the rearranging of node easier. By turning this option on, labels can be dragged and repositioned. The label will then remain in the same position relative to the node once the node is dragged. Toggle this functionality with the shortcut key **m**.
- **Link Strain**: Color all edges relative to their size. Longer edges will be colored relative to the color defined in the *Coloring* tab in the top menu. This option allows users to quickly identify which edges are longer, identifying nodes and connections that might be straining the graph.

2.11.2 Graph Manipulations

- **Reload Graph**: Re-renders the current graph. This option also removes reference nodes and resets SAMMI variables. Use this option in case of unexpected graph behavior.
- **Pause Simulation**: Pauses the force-directing rendering. Graph rendering is resumed once the graph is clicked.
- **Recenter Graph**: Returns graph to original zoom level and centers the visualization. Use this option to reset the panning and zooming options.
- **Fix Nodes**: Fixes all selected nodes. These nodes will no longer be re-arranged in the force-directing, and all floating nodes will be arranged considering its fixed position. Shortcut key **f**.
- **Toggle Fix**: Toggles the fixing status of selected nodes. Fixed nodes will become floating and floating nodes will become fixed. To release all selected nodes, first fix them with the options above then toggle their fixing status. Shortcut key **t**.
- **Isolate Reaction**: Isolates the selected reaction. This function only works if a single reaction node is selected. An additional copy of each metabolite associated with the selected reaction will be made, and will become the metabolite node associated with the reaction node, isolating the reaction in its own component. See figure below. Shortcut key **i** if a single reaction node is selected, otherwise this shortcut will refer to the *Isolate Metabolite* option described below.
- **Isolate Metabolite**: Works for any set of selected metabolite and reaction nodes. An additional copy of each selected metabolite node will be made, and will become the associated copy of each selected reaction node in which the metabolite takes place. See figure below. Shortcut key **i** if a single metabolite node or multiple nodes are selected. If a single reaction node is selected this shortcut refers to the Isolate Reaction option above.
- **Select Neighbors**: Select all nodes connected to all nodes already selected. shortcut key **n**.
- **Break Metabolite**: Makes one copy of all selected metabolites for each reaction the metabolite participates. That is, if a selected metabolite node is connected to five reaction nodes, five copies of the metabolite will be made and connected to each reaction node it participates in. This functionality is meant to split up secondary metabolites such as hydrogen, water, and ATP. Shortcut key **b**.
- **Join Metabolite**: Joins into one node all selected metabolites with matching *class*. For the difference between node *id* and *class* please refer to the *SAMMI Map Format* section of the documentation. For instance, if two glucose metabolites are selected, these will be joined into a single node which will be connected to all reactions connected to the original two glucose nodes. Shortcut key **j**.
- **Shelve Metabolite**: Temporarily removes metabolites from the graph. These metabolites can be added back to the map under the *Shelved Metabolites* subsection of the floating menu described below. Metabolites can also be shelved by uploading a list of regular expressions under the *Load Metabolites to Shelve* option, or from a standard list of metabolites under the *Shelve Standard Metabolites* option, both in the *Upload/Download* tab in the top menu. For more details please refer to the *Secondary Metabolites* section of the documentation. Shortcut key **s**.
- **Delete Nodes**: Remove from the graph selected nodes. Removed nodes cannot be re-introduced, but action can be undone immediately after deletion with the *Ctrl + z* shortcut. Shortcut key **Delete**.

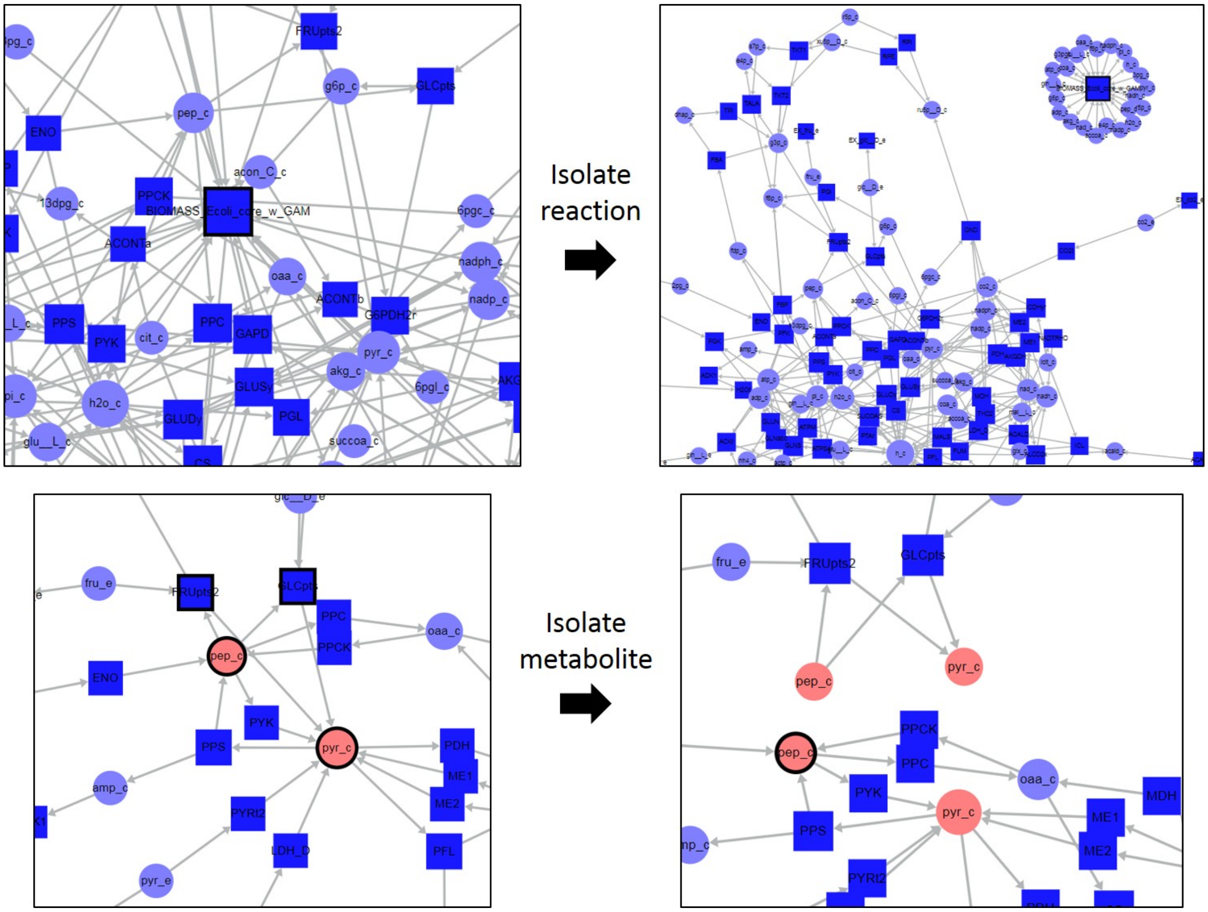


Fig. 12: (Top) *Isolate Reaction* example. The Biomass reaction is connected to multiple metabolites in the map. By isolating the reaction, an additional copy of each metabolite present in the biomass reaction is made, and that copy becomes the copy associated with the Biomass reaction node, isolating the reaction in its own component. (Bottom) *Isolate Metabolite* example. Metabolites *pep_c* and *pyr_c* as well as reactions *FRUpts2* and *GLCpts* are selected. By isolating the metabolites, a second copy of each metabolite is made, and that copy becomes the node associated with the selected reactions. For both isolating functions, all unselected reactions where the copied metabolites take place remain connected to the original metabolite node.

- **Keep Nodes**: Keeps all selected nodes. All unselected nodes are deleted. Removed nodes cannot be re-introduced, but action can be undone immediately after deletion with the *Ctrl + z* shortcut.
- **Make Secondary**: Label selected metabolite nodes as secondary. Metabolite nodes must have a degree of one to be marked as secondary. Secondary nodes are drawn smaller and are not drawn to the center of the map, but to a position calculated relative to the substrates and products of the reaction. Shortcut key **q** if any of the selected metabolite nodes are primary. If all selected metabolite nodes are secondary, this shortcut makes all selected metabolite nodes primary.
- **Make Primary**: Label selected metabolite nodes as primary. Shortcut key **q** if all selected metabolite nodes are secondary. Otherwise, this shortcut makes all selected nodes secondary.
- **Curve Link**: Adds curve to selected node. This function only works if a single node is selected. This function adds a reference node to the graph, which can be dragged to define the curve appearance. All incoming links are aligned in one end of the node and all outgoing links are aligned at the opposite end. While this function works with metabolite nodes, its use is recommended only for reaction nodes. Curving links related to metabolite nodes can overlay arrowheads related to different reaction reversibility. Shortcut key **c**.
- **Delete Curve**: Removes curving of links of selected nodes, returning the associated edges to straight lines.
- **Auto Curve**: Curves the links of selected reaction nodes to a position calculated relative to primary substrate and products related to that reaction. Applies only to reaction nodes. Shortcut key **d**.
- **Group Nodes**: Group selected nodes. After grouping, once a node in that group is selected, all remaining nodes in that group are also selected.
- **Ungroup Nodes**: Remove grouping assignment of selected nodes.
- **Collapse**: Combines selected reaction nodes into a single node. Selected metabolite nodes are ignored in this function. Metabolites connected only to selected reaction nodes are removed from the graph. For metabolites connected to unselected nodes and multiple selected nodes, only only one edge will be kept. Following this function the editing window will open so the user can rename the resulting node.
- **Shortest Path**: Selects nodes in the shortest path between consecutively selected nodes. For instance, if nodes A then B then C are selected, this function will select the nodes in the shortest path from A to C passing through B. Shortcut key **p**.
- **Find Long Path**: Finds a long path between first two selected nodes. Finding the longest path in a graph is an NP hard problem, so this function does not guarantee the longest path. However, it performs a modified Breadth First Search algorithm which return sequentially longer paths. The algorithm does have a stopping condition, but can also exit once the running time is above the parameter defined in *Path Execution Timeout (s)* under *Settings* in the top menu. If nodes A then B then C are selected, for instance, this function will return the longest path between A and B found using the modified algorithm in the given time limit.
- **Smallest Circle**: Finds the smallest circle between adjacent nodes. Only works if two adjacent nodes are selected. In practice, this function searches for the shortest path between the two selected nodes in a graph where their connected link has been removed.
- **Find Component**: Selects additional nodes to form a component containing the selected nodes. In practice, this function selected all nodes in pairwise shortest paths, including ties, between the selected nodes.
- **Edit Attributes**: Opens node attributes in the supporting menu window and allow for editing of such attributes.
- **Reverse Reaction**: Reverses the directionality of the selected reactions. All substrates become products and all products become substrates. This function is performed independently of reaction directionality, so it is up to the user to select only reversible reactions to maintain feasible reversibility. If the box *Reverse Flux* is checked, all flux data associated with the selected reaction nodes are also reversed (multiplied by -1). If you wish to reverse the data in some conditions but not other use the *Edit Attributes* function.
- **Pretify Reactions**: Organized the selected reactions in a visually pleasing format. Meant to be used only with orphan reaction (reactions defining their own component).

- **Untrap Nodes**: Removes selected nodes from added shapes. View *Text and Shapes* subsection below.

2.11.3 Subgraph Manipulations

- **Join Subgraphs**: This function allows users to join different subgraphs. Once this function is activated, the support menu is opened. A text input field is displayed at the top of the support menu with the name of the current subgraph. This value can be changed to rename the joined subgraph. Select all other subgraphs you wish to join with the current one by clicking their associated checkbox. At the bottom of the window an option to *Join similar metabolites* is available. If selected, this option will join metabolites of the same class in the resulting subgraph. That is, if pyruvate is present in multiple subgraphs being joined, this option will combine those into a single pyruvate node. If unchecked, the node defined in each subgraph will be displayed separately. Finally, click *Join* to combine the selected subgraphs with the current one. All nodes from the last joined subgraph will be selected upon joining.
- **Rename Subgraph**: This function will open the support menu with a text input field. Edit that field and hit *Enter* to rename the current subgraph.

2.11.4 Search

Type in the text field and press *Enter* to search for nodes. Nodes matching the search will be selected. If multiple subgraphs are present, the supporting menu will also be displayed showing how many nodes in each other subgraph also match the search. To navigate to a different subgraph, click on that subgraph in the supporting menu. The following options are also available for the search function:

- **Focus**: If this option is selected, all nodes matching the search and their associated nodes are brought into focus by hiding all remaining nodes. To view all nodes again just click anywhere in the graph. If this box is not checked the nodes matching the search will still be selected but no nodes will be hidden.
- **RegExp**: Determines whether the search is done by regular expression or not. If checked, the inputed text is taken as a regular expression, otherwise it is simply taken as plain text. If this option is not checked capitalization is not considered.

The **Highlight Metabolites** option is also available under this subsection. This functionality only works if a single metabolite node is selected. This function loads a temporary graph displaying the selected node, and creates a reaction node for all reaction in which that node participates, appending the subgraph in which that reaction participates to the node name. Nodes are also arranged in a circle with the metabolite node in the center. Drag the reference node to edit the size of the circle and view reaction names clearly. To navigate to a different subgraph double-click the reaction node. To return to the current graph double click on the graph background.

2.11.5 Arranging

This section allows users to arrange the selected nodes in different shapes. All selected nodes are fixed and edges curved according to the desired shape when using these functions. Two options are associated with these arranging function.

- **Reverse**: Each arranging function organizes the nodes in a certain order. Check this option to reverse this order. For instance, the circle option arranges the nodes in a clockwise order by default. Select this option to arrange the nodes in a counterclockwise order instead.
- **Selection Order/Position**: If *Selection Order* is selected, nodes will be arranged in the order in which they were selected. Use this option if you selected the nodes in the desired order, or if the path options were used to select nodes. If *Position* is selected, nodes will be arranged in an order similar to the one they are in at the beginning of the arranging process. This function might be more suited if nodes are selected using the panning selection or the component function, or if you wish to change the order of arranged nodes.

The available arranging function are:

Arrange selected nodes in a vertical line. Nodes are evenly distributed along the line. Drag the reference node to change the size of the line.

Arrange selected nodes in a horizontal line. Nodes are evenly distributed along the line. Drag the reference node to change the size of the line.

Arrange selected nodes in a diagonal line. Nodes are evenly distributed along the line. Drag the reference node to change the size and angle of the line.

Arrange selected nodes in a circle. Nodes are evenly distributed along the circle. Drag the reference node to change the size and rotation of the circle.

Arrange the selected nodes in a rectangular shape. This function works in two steps. When first activated, nodes will be arranged along a rectangle and a reference node will be added. Drag this reference node to adjust the size of the rectangle. Once this node is dragged it will be reloaded slightly lower and to the right. Next, drag the desired graph nodes along the rectangular shape to the desired location. Nodes can be dragged inside but not outside the rectangular area. When done arranging the nodes, click the reference node to exit this functionality. If you still wish to edit the size of the rectangle use the scale function below.

Rotate, pivot, and scale selected nodes. This function also works in two steps. When first activated, a reference node will be added. Drag this node to define the center of rotation. After dragging this node, a second reference node will be added to the left of the defined center. Drag this second node to rotate and scale the selected nodes around the defined center.

Scale the position of selected nodes. A reference node is added to the bottom right of the selected nodes. Drag this node to scale the position of the selected nodes.

2.11.6 Text and Shapes

This function adds annotation text to the graph. Click this button then on the graph area to add a text box displaying *New Text* to the graph. To edit this text, double-click the added text and a text input field will be displayed in the support menu for editing. The input text can also be dragged to different locations. Font size can be changed in the *Sizing* tab in the top menu under *Added Text Size*.

T

Rectangular shapes can be added to the graph by clicking this button then clicking on the graph area. A standard sized shape is initially added, but the edges of the rectangle can be dragged to adjust and move the box. By dragging and dropping nodes into the rectangle, nodes can be trapped into the rectangular area. To allow trapped nodes to leave the rectangular shape simply select the node and click on *Untrap Nodes* under *Graph Manipulations* in the floating menu, or use the shortcut key **u**.

2.11.7 Shelved Metabolites

Metabolite nodes can be shelved by using the *Shelve Metabolite* function under *Graph Manipulations*, or by uploading a list of regular expressions in the *Upload/Download* tab in the top menu. All shelved metabolites will be displayed in the dropdown list in this section. Select individual metabolites from the list to return these metabolites to the map. Alternatively, click on *Put All Back* to reintroduce all shelved metabolites.

2.11.8 Commits

The state of the current graph can be saved by making a commit of the current map. This commit will save the state of all subgraphs. To keep file sizes down, commits are not downloaded in the SAMMI format file. Up to 20 commits can be made of any given graph. If more commits than that are made, the oldest commits are over-written. Node positioning and associated data are saved with every commit, but not graph settings such as force-directing setting, colorscales, and font sizes. To save commits and map settings users can download the current graph in SAMMI format.

To make a commit, type in an optional description of the current state of the map in the *Description* field and click *Commit Current*. If this field is left blank, the current date and time will be used as the commit name. The commit is then added to the dropdown menu in this section. To return to a given commit simply select that option in this dropdown menu.

2.11.9 Add Nodes and Edges

Functions in this section allow for the addition of new reaction and metabolite nodes, and for the addition of new connections. The two types of functions in this section are:

- **Reaction/Metabolite**: To add a new reaction or metabolite, type the name of the desired node in the appropriate field. Both fields will autocomplete with the class of existing nodes. If the inputed name matches the class of an existing node, in any subgraph, that node will be imported to the current graph with all associated data. Reaction nodes are also imported with their associated metabolite nodes. If the name inputed does not match the class of any existing node, in any subgraph, a new node is created and added to the graph with no connections.
- **Connect as Substrate/Product**: These functions will connect all selected metabolite nodes to all selected reaction nodes as substrates or products accordingly. To use this function select at least one metabolite and one reaction node simultaneously by holding down the *Shift* key and clicking multiple nodes. Then select whether metabolites are to be added as reaction substrates or products. All selected metabolites will then be connected to all selected reactions as selected. Previously connected nodes will be ignored.

2.12 Keyboard Shortcuts

A number of keyboard shortcuts are available for easy editing and navigation. These are:

- **Ctrl + click**: Holding the *Ctrl* key and clicking on a node will bring that node into focus. This shortcut performs the same function as searching for that node specifically.
- **Shift + click**: Holding down the *Shift* key allows for the selection of multiple nodes by clicking each node. Holding down *Shift* and clicking the background also allows for the selection of nodes in a rectangular area by clicking and dragging over that area.
- **Alt**: Holding down the *Alt* key allows users to select nodes by hovering over them. This allows for fast selection of a large number of nodes.
- **Ctrl + z**: Undo previous action.
- **Ctrl + y**: Redo undone action.
- **Delete**: Delete selected nodes.
- **Arrows**: Move selected nodes in the arrow's direction.
- **a**: Select all graph nodes.

- **b**: Break selected metabolite nodes.
- **c**: Curve edges associated with selected node.
- **d**: Automatically curve edges associated with selected nodes to estimated position.
- **f**: Fix selected node.
- **h**: Highlight selected metabolite node.
- **i**: If a single reaction node is selected, isolate the reaction. If multiple nodes, or a single metabolite node, are selected, isolates metabolites.
- **j**: Join Metabolites.
- **m**: Toggle move labels functionality.
- **n**: Select neighbors of selected nodes.
- **p**: Finds shortest path between selected nodes.
- **q**: If all selected metabolite nodes of degree one are secondary, makes those nodes primary. Otherwise makes all metabolites nodes of degree one secondary.
- **s**: Shelve selected metabolite nodes.
- **t**: Toggle fix status of selected node.
- **u**: Untrap selected node from added shape.
- **p**: Pause simulation.
- **r**: Reload graph.

2.13 SAMMI Map Format

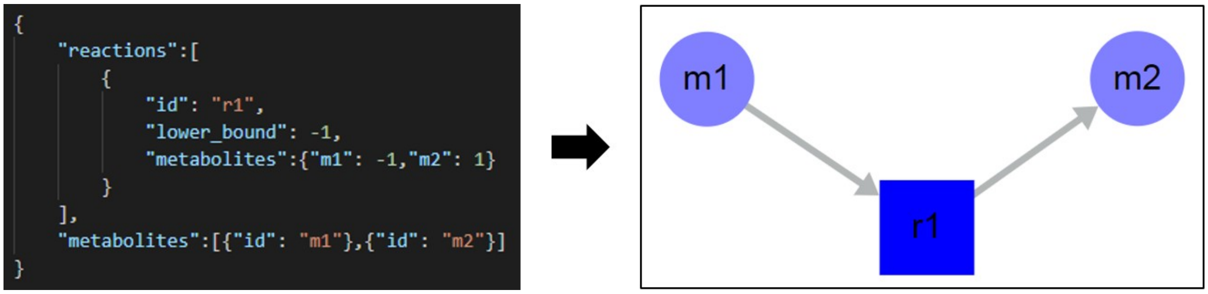
As described in the *Loading a Map* section, SAMMI takes files in the SBML format, either in XML, JSON, or SBML file extensions, a well as BioPAX annotated models. To load a personalized file, however, users can generate a JSON file with minimal requirements. The file must define a single object with two fields: **reactions** and **metabolites**, each defining a vector of objects. Each object in each of these vectors define a single reaction or metabolite respectively.

Metabolite objects may contain a single field **id** defining a unique metabolite ID. These fields will be extended to metabolite classes when the model is first uploaded into SAMMI. As metabolite nodes are duplicated, unique IDs will be generated for new nodes, but the class field will remain the same defined here. This unique ID generation holds true for all uploaded models. Additional fields may be added to each metabolite (e.g. name or database ID), which will be imported as metabolite data.

Reaction objects must contain two fields. The first is **id**, defining a unique reaction ID. The second is **metabolites**, defining an object where each field is defined by a metabolite ID, and specifies the stoichiometric coefficient of that metabolite in that reaction. Negative values indicated substrates and positive values indicate products. It is strongly recommended to add a third field for reaction lower bounds in order to define reaction reversibility. The name of this field can be changed in SAMMI in the *Upload/Download* tab in the top menu. Additional fields can also be added to reaction objects to be imported as associated reaction data.

2.14 COBRA Compatibility

The COnstraint-Based Reconstruction and Analysis (COBRA) toolbox is one of the most popular toolboxes for modeling metabolic networks. COBRA has been developed for MATLAB (Heirendt et. al.), Python (Ebrahim et. al.) and Julia (Heirendt et. al. (b)) programming languages, all of which can be accessed through this link.



```
{
    "reactions":[
        {
            "id": "r1",
            "lower_bound": -1,
            "metabolites":{"m1": -1,"m2": 1}
        }
    ],
    "metabolites":[{"id": "m1"},{"id": "m2"}]
}
```

Fig. 13: Minimal requirements and format definition for models to be uploaded into SAMMI.

SAMMI users can visualize metabolic models directly from the MATLAB and Python command line using the SAMMIM and SAMMIpy plugins respectively. Please refer to the documentation of each of these tools for more information.

The MATLAB and Python versions of COBRA also offer tools for exporting metabolic models in versions compatible with SAMMI. In this section we cover these options and offer alternative scripts to do so for users who wish to use the standalone version of SAMMI at www.sammitool.com. It is worth noting that all of these options export models in version that should first be loaded as SBML models, not SAMMI specific formats (e.g. using the "Load Single Model" or "Load Model to Parse" options in the SAMMI home page).

2.14.1 MATLAB

To visualize maps directly from MATLAB please refer to the SAMMIM plugin.

MATLAB COBRA models are defined by a struct variable. The COBRA toolbox offers the function `writeCbModel` to export models in a variety of formats. Files can be exported in the SBML format for SAMMI compatibility:

```
writeCbModel(model,'format','sbml')
```

This will generate an XML file that can be uploaded in the SAMMI interface. To define other function call arguments please refer to the `writeCbModel` documentation.

While this option provides an SBML compatible model, user defined fields are not exported, and reaction and metabolite IDs may be altered in the export process. To account for these issues, we provide the following script defining the `sammiJson` function, which exports the model to a SAMMI compatible JSON file. Here, all struct fields with the same size as the `rxns` field are exported as reaction-associated data, and all fields with the same size as the `mets` field are exported as metabolite associated data.

Download script here: `sammiJson.m`

You can test the given function with following bit of code that will generate the `e_coli_core.json` file:

```
%Get COBRA directory
global CBTDIR;
%Load models to test
load([CBTDIR '/test/models/mat/ecoli_core_model.mat'])
%Add random field
model.newMetField = rand(length(model.mets),1);
model.newRxnField = rand(length(model.rxns),1);
%Write to file
sammiJson(model,'e_coli_core')
```

2.14.2 Python

To visualize models directly from Python please refer to the SAMMIpy plugin.

Metabolic models is COBRApy are defined by a toolbox specific `cobra.DictList` class variable. COBRApy offers the function `save_json_model` that exports models to a SAMMI compatible JSON file with the function call:

```
cobra.io.save_json_model(model, "test.json")
```

This function call also does not export user-defined fields, however. For that we provide the following alternative script defining the `sammiJson` function that exports to a JSON file all reaction and metabolite fields defined by a string, float, int or bool. It is worth noting that this approach will generate larger files with perhaps unwanted fields.

Download script here: `sammiJson.py`

Once the function is loaded you can test it with the following bit of script.

```
#Read in model
model = cobra.test.create_test_model("salmonella")
#Add random reaction fields
for rx in model.reactions:
    rx.addedFieldNum = np.random.rand()
    rx.addedFieldString = ''.join(random.choice(string.ascii_uppercase + string.
↪digits) for _ in range(8))
#Add random metabolite fields
for met in model.metabolites:
    met.addedFieldNum = np.random.rand()
    met.addedFieldString = ''.join(random.choice(string.ascii_uppercase + string.
↪digits) for _ in range(8))
#Write to file
sammiJson(model,'core.json')
```

2.15 Video Tutorials

This section provides several video tutorials exemplifying a number of different functionalities present in SAMMI.

2.15.1 Upload Maps

Upload SBML and BioPAX annotated models as well as metabolic pathways from KEGG.

2.15.2 Download Maps

Download SAMMI maps in SAMMI specific format, as well as formats compatible with other platforms and static images.

2.15.3 Shelving Metabolites

This tutorial covers different ways to shelve, or temporarily remove nodes from the map, as well as how to put them back.

2.15.4 Positioning Nodes and Edges

How to fix nodes in positions, arrange them in different shapes, and curve edges.

2.15.5 Adding, Removing and Editing Nodes

How to split, duplicate, and join nodes, and how to add new or existing nodes.

2.15.6 Navigating Graph

How to navigate between different subgraphs, search for nodes, see which reactions nodes participate in, and track metabolic pathways between different subgraphs.

2.15.7 Visualizing Data

How to plot data onto the model and change color and size scales.

2.15.8 Selecting Paths and Components

How to automatically select relevant pathways and components within your map.

2.15.9 Combining Subgraphs

How to combine and rename subgraphs.

2.15.10 Changing Default Settings

Change default look node and edge settings as well as simulation parameters to change the map appearance and automated node positioning.

2.15.11 Undo Map Edits

How to save different map states and return to different iterations of your visualization.

2.15.12 Annotating Text and Shapes

Add annotation text and rectangular shapes to your graph.

2.15.13 COBRA Plugins

Introduction to COBRA plugins. Load SAMMI models directly from within the Python and MATLAB command lines.

2.16 Copyright and License

SAMMI is a tool for visualization of metabolic networks

2.17 Acknowledgments

SAMMI was developed using a number of JavaScript packages we would like to acknowledge. These are:

- D3js.
- jQuery.
- jQuery UI.
- Tippy.
- jscolor.
- jsPDF.
- svg2pdf.

2.18 Contact Us

The SAMMI developers are committed to maintaining and advancing this tool. If you have questions, suggestion, comments, concerns or bug reports please do not hesitate to submit an issue report to our GitHub repository.

2.19 Citation

If you use SAMMI for your project, please cite the following publication:

> Schultz, A., & Akbani, R. (2019). SAMMI: A Semi-Automated Tool for the Visualization of Metabolic Networks. *Bioinformatics*.