Metabolic Pathway Visualization

Documentation and Design

DRAFT: April 3, 2015

Contents

1 Visualization

Currently, all of the pathways between D-Glucose (C00031) and L-Tryptophan (C00078) are being displayed and can be accessed here: http://prb2.web.rice.edu/metapaths/Pathways.html

Pathways are represented as a graph consisting of compound nodes (green) and reaction pair nodes (black), connected by links. The start and end nodes are colored purple.

- Edge weight can represented with line thickness or opacity
- Edge arrows can be toggled on or off
- Pathways can be filtered by ignoring/including certain nodes or by specifying a maximum path length
- Nodes can be filtered by typing their ID into the appropriate field, or by right-clicking on the desired node and clicking the button.
- Right-clicking a node when viewing multiple paths will provide filtering options
- Right-clicking a node when view a single path will display more information about the selected compound or reaction pair
- Paths can be viewed one at a time, with the ability to navigate between each single path
- Nodes can be moved by clicking and dragging to a desired position. They can be released from this fixed position by double-clicking

2 Structure

The JavaScript code for the visualization consists of the following sections:

2.1 Data Loading

This section contains the functions which load the input files specified below, and make their data available to use for all functions. The start and end nodes are also noted and stored.

2.2 Graph Generation

This section initializes a `D3` force directed graph. `SVG` elements are created for the graph, nodes, and links. Properties of the graph are set, including: size, gravity, force, node and link colors, link weights, etc. Next, functions that enable interactivity with graph elements, such as dragging behavior, displaying tooltips, etc. are created and attached to mouse actions.

Field	Value	Description
Charge*	-100	The charge strength of nodes. A negative value indicates node repulsion, positive indicates node attraction.
Colors	Various	Node colors, highlight colors and edge colors are set here. These will be made user configurable in the future.
Gravity*	0.01	Inward force, directed toward the center of the graph. Serves to prevent nodes escaping the viewable area. This value is set very low to avoid excessive clumping of nodes, which hinders viewablity and access to nodes.
Size	Viewport size	Sets the size of the visualization.
Zoom*	[0.1, 10]	The zoomable scale range for the graph. In this case, zooming in to 1/10 of the starting scale and zooming out up to 10 times the original scale value is allowed.

* Detailed documentation on these, and all other `D3` properties that were used, can be found here: https://github.com/mbostock/d3/wiki/Force-Layout

2.3 Graph Operations

This section contains functions that manipulate the graph, including: adding/removing nodes, viewing single paths, viewing all paths, etc.

2.4 Filtering Functions

The functions in this section process the user's filter requests. The request is parsed and then a function corresponding to the filter action they requested is called to execute the filtering. The existing graph is cleared and the filtered pathways are then displayed.

2.5 Page Functions

This section contains functions that manage parts of the page's `HTML`. These functions are called when the graph changes, so that information on the page (such as path length, ignored nodes, etc.) can be updated to reflect the changes to the graph.

3 Input Files

3.1 AllPathways.json

`AllPathways` is the primary data structure that contains all of the information necessary to plot each pathway. It is an array of of JavaScript objects, each of which represents a single pathway. Each `pathway` object contains a `nodes` array and a `links` array. Each `node` in the `nodes` array contains the properties of that node. Each `link` in the `links` array contains a mapping of `source` and `target` compounds.

Node:

```
fixed: true,
index: 0,
x: 40,
y: 365
```

Link:

```
source: "C00031",
target: "RP00060",
```

3.2 NodesMap.json

`NodesMap` contains an entry for each unique node (compounds and reaction pairs) that are present in any of the pathways being visualized. Each entry contains contains information about the node, but some fields (ec, reacID and reacName) are only pertinent to reaction pairs.

C00031:

```
ec: ""
id: "C00031",
incoming: 1,
name: "D-Glucose",
outgoing: 1,
pathways: [0,1,2,3,4,...],
reacID: "",
reacName: "",
```

RP00060:

```
ec: "2.7.1.1"
id: "RP00060",
incoming: 5,
name: "",
```

```
outgoing:  5,
pathways:  [1,2,4,7,9],
reacID: "R00299",
reacName:  "ATP:D-glucose 6-phosphotransferase",
```

3.3 LinkWeights.json

Each entry in `LinkWeights` is a comma delimited string of the two compound IDs comprising that link, which is mapped to its weight. The weight is defined as the number of pathways that this link appears in.

Link Weight:

```
{ "RP02140,C00463":  3 }
```