GRAPHERY

A Biological Network Algorithm Tutorial Webservice

Heyuan Zeng¹, Anna Ritz²
1. Computer Science Department, Reed College
2. Biology Department, Reed College

Introduction

Network visualization is crucial for integrating and interpreting experimental and annotated evidence across different biological scales. Web-based platforms have emerged to enable collaboration among biological and computational researchers. As network-based visualization becomes popular in the scientific community, developing materials to train biologists to understand the outputs of network-based approaches are essential. We present Graphery, a web-based graph algorithm tutorial that is designed for biological researchers.

Administration System

Users will be able to register, suggest content, and receive notifications about new tutorials, graphs, and translations. Currently, administrators can add content, with plans for user contributions.

Tutorials and Graphs

The collection page lists all the published content in the database (namely, graphs and tutorials). It also supports filters and keyword searching.

Tutorial View

A tutorial has three sections: a text content section, a graph display section, and an editor section. The text content section contains a text description and discussion of a graph algorithm, and it will be translated into different languages.

Users interact with the tutorial on in the graph and editor sections. The graph display section allows users to interact with the graph by zooming in/out or dragging the graph elements. Users can run the tutorial on different graphs in the database.

The editor section, which includes a control unit and a code editor, is the core of the advanced interactions. This system executes the python implementation in a step-by-step debugger-like manner, highlighting nodes and edges in the graph display section. In this screenshot, execution is paused at the line with the arrow. The variable list to the right of the code editor will be updated if any variable specified by a special @tracer decorator is changed. Additionally, if the variable is referencing to a graph element, the graph element, and the variable name in the list will be highlighted.

With the aid of this system, users can know how the code is working on the graphs. And the interactions do not stop here. Users can change the code in the editor and run the code either on the cloud or their computers if they've downloaded a custom script. The execution result will be sent back to the website to be visualized. We also provide a playground page where users can edit existing code or write brand new code from on a basic template we provide. You can check out the docs for further usages.

https://graphery.reedcompbio.org
Docs: https://docs.graphery.reedcompbio.org