GRAPHERY

A Biological Network Algorithm Tutorial Webservice

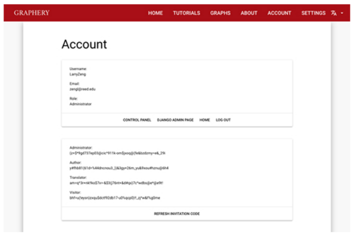
Heyuan Zeng[1], Anna Ritz[2]
1. Computer Science Department, Reed College
2. Biology Department, Reed College

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.



Account

Username:
LarryZeng

Email:
zengl@reed.edu

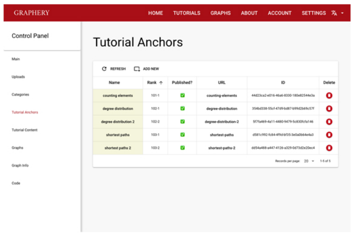
Role:
Administrator

Administrator:
[illegible]

Author:
[illegible]

Translator:
[illegible]

Visitor:
[illegible]



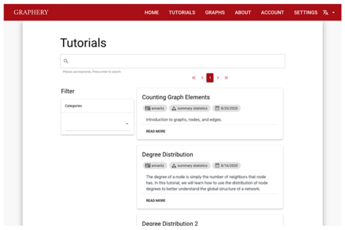
Tutorial Anchors

Name	Rank	Published?	URL	ID	Delete
counting elements	101-1	✓	counting-elements	44d23ca2-e016-46a6-8330-180e82544e3a	●
degree distribution	102-1	✓	degree-distribution	354bd338-55cf-47d9-bd87-b99d2b69c57f	●
degree distribution 2	102-2	✓	degree-distribution-2	5f7fa469-4a11-4480-9479-5c830fcfa146	●
shortest paths	103-1	✓	shortest-paths	d581c992-fc84-4f9d-bf35-3e0a0b64e4a3	●
shortest paths 2	103-2	✓	shortest-paths-2	dd54a488-a447-4126-a329-0d73d2e20ec4	●

Records per page: 20 1-5 of 5

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.



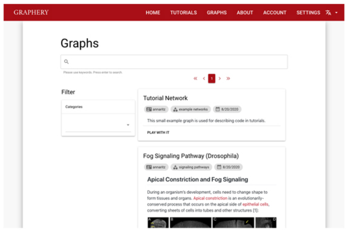
Tutorials

Please use keywords. Press enter to search.

Counting Graph Elements
annaritz · summary statistics · 8/20/2020
Introduction to graphs, nodes, and edges.
READ MORE

Degree Distribution
annaritz · summary statistics · 8/16/2020
The degree of a node is simply the number of neighbors that node has. In this tutorial, we will learn how to use the distribution of node degrees to better understand the global structure of a network.
READ MORE

Degree Distribution 2



Graphs

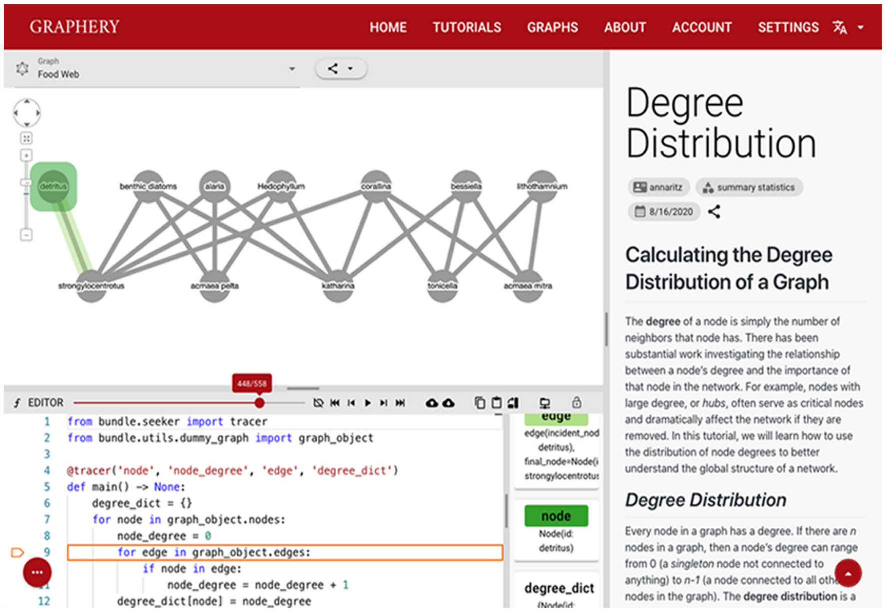
Please use keywords. Press enter to search.

Tutorial Network
annaritz · example networks · 8/20/2020
This small example graph is used for describing code in tutorials.
PLAY WITH IT

Fog Signaling Pathway (Drosophila)
annaritz · signaling pathways · 8/20/2020

Apical Constriction and Fog Signaling

During an organism's development, cells need to change shape to form tissues and organs. Apical constriction is an evolutionarily-conserved process that occurs on the apical side of epithelial cells, converting sheets of cells into tubes and other structures [1]:



```python
from bundle.seeker import tracer
from bundle.utils.dummy_graph import graph_object

@tracer('node', 'node_degree', 'edge', 'degree_dict')
def main() -> None:
    degree_dict = {}
    for node in graph_object.nodes:
        node_degree = 0
        for edge in graph_object.edges:
            if node in edge:
                node_degree = node_degree + 1
        degree_dict[node] = node_degree
```

edge
edge(incident_nod detritus), final_node=Node(i strongylocentrotus

node
Node(id: detritus)

degree_dict
(Node(id:

Degree Distribution

annaritz · summary statistics · 8/16/2020

Calculating the Degree Distribution of a Graph

The **degree** of a node is simply the number of neighbors that node has. There has been substantial work investigating the relationship between a node's degree and the importance of that node in the network. For example, nodes with large degree, or *hubs*, often serve as critical nodes and dramatically affect the network if they are removed. In this tutorial, we will learn how to use the distribution of node degrees to better understand the global structure of a network.

Degree Distribution

Every node in a graph has a degree. If there are *n* nodes in a graph, then a node's degree can range from 0 (a *singleton* node not connected to anything) to *n-1* (a node connected to all oth nodes in the graph). The **degree distribution** is a

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`