User-based online Graph editing tool

Utsav Sinha, Kundan Kumar, Vicki Anand

CS252 Project

August 2014 - November 2014

भारतीय प्रौद्योगिकी संस्थान कानपुर

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Outline

1. Project Overview
   - Cytoscape
   - Motivation
   - Objectives

Outline

1. Project Overview
   - Cytoscape
   - Motivation
   - Objectives
2. Implementation
   - Input
   - Functionalities
   - Querying Graph
   - Appearance
   - Version Control

Outline

1. Project Overview
   - Cytoscape
   - Motivation
   - Objectives
2. Implementation
   - Input
   - Functionalities
   - Querying Graph
   - Appearance
   - Version Control
3. Methods Used
   - MongoDB database
   - PHP and Javascript
   - Styling

Outline

1. Project Overview
   - Cytoscape
   - Motivation
   - Objectives
2. Implementation
   - Input
   - Functionalities
   - Querying Graph
   - Appearance
   - Version Control
3. Methods Used
   - MongoDB database
   - PHP and Javascript
   - Styling
4. References

What is Cytoscape ?

Cytoscape is an open source bioinformatics software platform for visualizing molecular interaction networks and integrating with gene expression profiles and other state data. Cytoscape has a JavaScript-centric project named *Cytoscape.js*.
Cytoscape.js is an open-source graph library written in JavaScript which allows you to easily display and manipulate rich, interactive graphs along with graph analysis and visualisation.

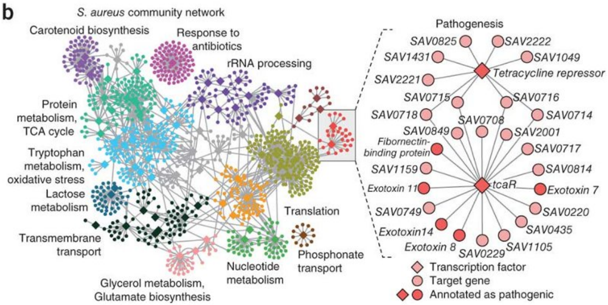


Figure: Graph using Cytoscape[Conc]

What this project is about ?

This projects creates a user based online graph editing tool using cytoscape.js and PHP with MongoDB as the backend database. Users can login into their accounts, create graphs and save their sessions which can be worked on later.

Motivation

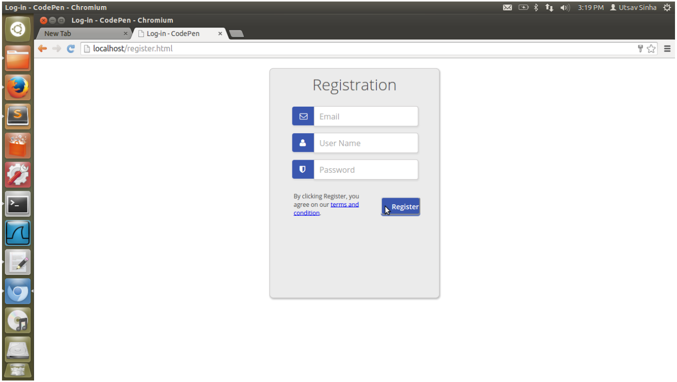
The project began with a quest to learn MongoDB and create a useful web application.
Thanks to our mentor and guide, *Prof Arnab Bhattacharya* for leading us to create a web plugin for *cytoscape.js*. The project was a perfect match for honing our new learnt skills and acquire knowledge about this wonderful cytoscape tool.

Objectives I

We are given a large graph, for eg. a DNA structure and individual researchers want to simulate the effects of their experiment by genetic recombination of the molecule.
Since the base graph remains the same with minor modifications (in terms of changes in the graph), we need a tool so that users can easily make changes using an interactive GUI. Also, a user may wish to have several parallel versions of the same molecule and save them seperately.

Objectives II

Users may want to revert back to their earlier versions of graph, just like in version control *git* or they may wish to work on other user's graph.
This project help realize all these goals using a simple easy to use graphical online tool. This tool uses cytoscape.js and all references to graph is made in cytoscape's context.



Log-in - CodePen - Chromium

localhost/register.html

Registration

Email

User Name

Password

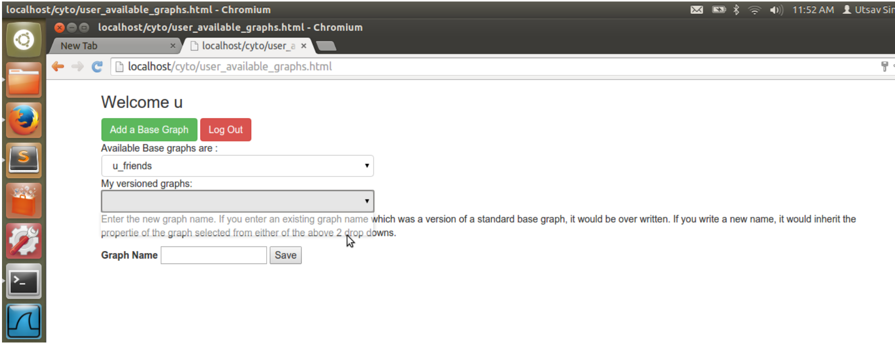
By clicking Register, you agree on our terms and condition.

Register

Figure: Register Interface

Inputs given

Users can switch from offline made graphs to this online tool by importing their graph data and style in *JSON* format. This will save and create the graph. All further work can be done on the online tool itself.
So after login, a user sees a list of all public projects (graphs) on which collaborative work can be done. A list of private versions of these graphs is also listed. A user can opt to overwrite an existing graph (of which he is the owner) or fork a new version of a previous work (both from the public and private list).


localhost/cyto/user_available_graphs.html - Chromium
11:52 AM
Utsav Sin
New Tab
localhost/cyto/user_a
localhost/cyto/user_available_graphs.html
Add a Base Graph
Log Out
Save

Welcome u

Available Base graphs are :

u_friends

My versioned graphs:

Enter the new graph name. If you enter an existing graph name which was a version of a standard base graph, it would be over written. If you write a new name, it would inherit the propertie of the graph selected from either of the above 2 drop downs.

Graph Name

Figure: Interface for selecting graph

Functionalities Overview I

After selecting the graph to work on, the user is redirected to the editing page where he can analyze the graph and edit the graph through the following operations:

- Add Node
- Add Edge
- Modify details
- Delete Node
- Delete Edge

Functionalities Overview II

The last 3 operations (modifying and deleting nodes and edges) can be performed by right clicking on the appropriate node/edge which is to be modified which opens an interactive interface:

- All Data: Used to display all data pertaining to the selected node through a popup
- Modify Details: Used to modify the data of the node
- Delete Node: Used to delete the node and all its incident edges
- Highlight Node: Used to hightlight the node with a different color; effect can be negated using de-hilight option

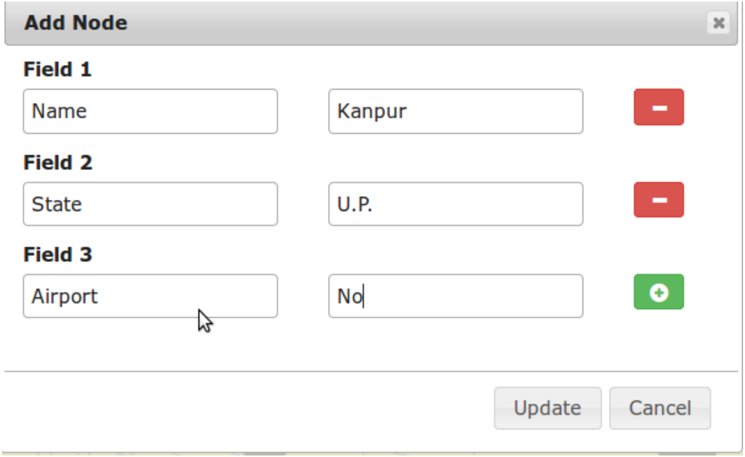
Simiarly, right clicking an edge offers similar functions but operational on edges instead of nodes.

Adding nodes and Edges

Add Nodes and edges open up a popup which can store data with as many key value pairs as required using add more fields button. Cytoscape stores nodes and edges as key value pairs. A typical node of a transportation graph with cities as nodes can be:

Table: City Node

Key	Value
Name	Kanpur
Pin Code	208016
State	U.P.
Airport	No



Add Node

Field 1

Name	Kanpur

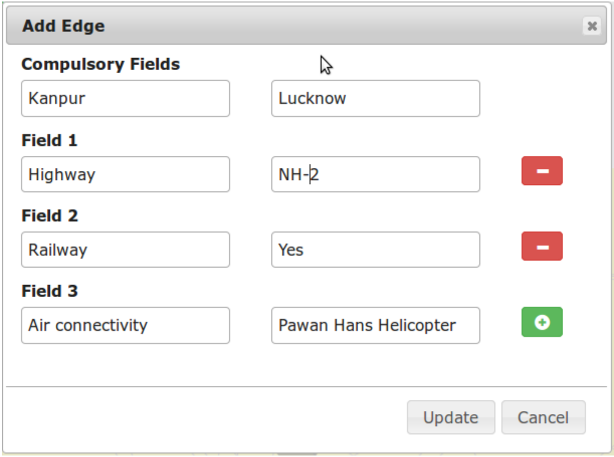
Field 2

State	U.P.

Field 3

Airport	No

Figure: Adding Node



Add Edge

Compulsory Fields

Kanpur	Lucknow

Field 1

Highway	NH-2

Field 2

Railway	Yes

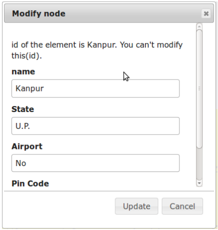
Field 3

Air connectivity	Pawan Hans Helicopter

Figure: Adding Edge

Modifying nodes and Edges

The values entered in the nodes and edges can also be changed later on by right clicking on the appropriate node/edge and selecting *Modify Details*. It opens up a popup like Figure 18



Modify node

id of the element is Kanpur. You can't modify this(id).

name: Kanpur

State: U.P.

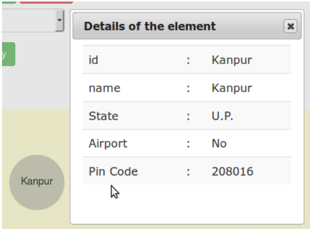
Airport: No

Pin Code

Figure: Modify Node

Seeing details of nodes and Edges

The data stored in the nodes and edges can be seen by right clicking on the appropriate node/edge and selecting *All Data*. It opens up a popup like Figure 19



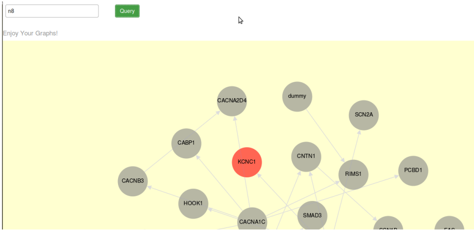
Details of the element

id	:	Kanpur
name	:	Kanpur
State	:	U.P.
Airport	:	No
Pin Code	:	208016

Figure: View Node Data

Querying Graph

Any node or edge in the graph can be focussed on in the editor by querying the element's unique id in the search box. This zooms the graph in or out depending upon the current zoom level and brings the element to the center of the screen. In case of node, color animation is also added for focussing. This feature is particularly useful when the graph becomes too large and finding the node/edge manually becomes difficult.



n8

Enjoy Your Graphs!

Appearance

Users can change the layout of the graph on the editor workspace in a variety of ways using the graph layout options. The available layouts include:

- Random
- Grid
- Breadth First
- Circle
- Preset
- Compound Spring Embedder

Depending upon the type of graph, different layouts can be used to best suit the data and provide it the most appealing visualization. Also, if the graph goes out of view due to zooming or other reasons, then these options helps to rescale the graph on the editor quickly.

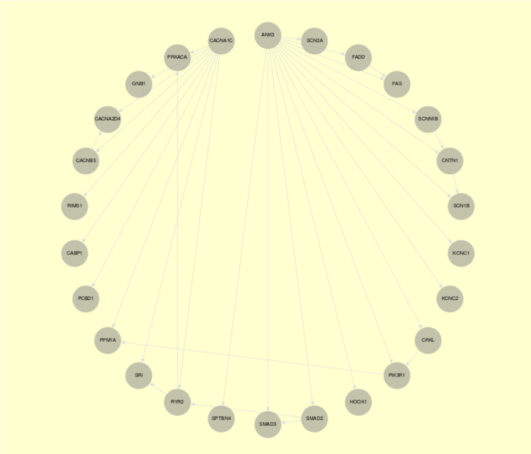
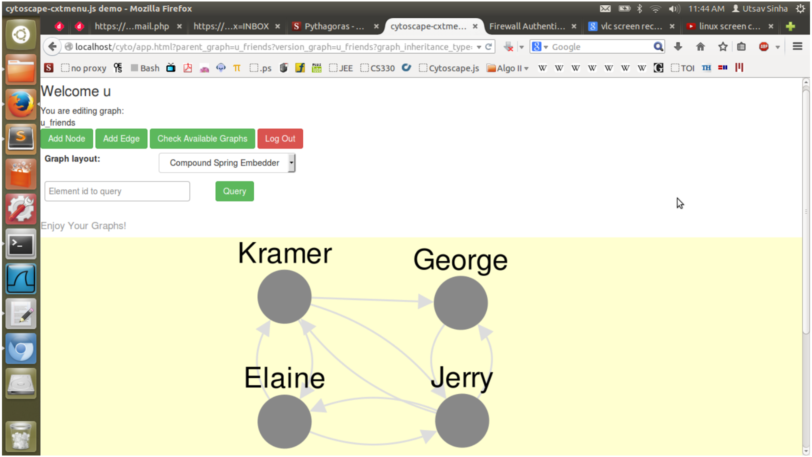


Figure: Circle Layout Graph



Welcome u

You are editing graph:
u_friends

Graph layout: Compound Spring Embedder

Element id to query

Enjoy Your Graphs!

Figure: Editor Interface

Version Control

There are 2 types of graphs:

- Public Graphs: Any user can create a graph (using Import Graph feature or otherwise) and share it as a base parent graph. Other users can create versions out of it.
- Private Versions: Users can create versions of public graphs which he can work on privately without sharing. These can later be made public.

Users can change create and store various versions of the same graph. At any time, the user can click on *Check Available Graphs* and see all pulic graphs as well as private versions of his modified graphs.

Database: MongoDB

MongoDB [Mon] was our database of choice since:

- It is schema-less and would handle all different types of graph representations (using key-value pairs in cytoscape) well
- cytoscape uses JSON for data representation and MongoDB can store the data in **JSON** format directly
- MongoDB also offers the facility of creating databases and tables (*collections*) on the fly which was very effective for version control

Backend: PHP I

PHP was used at the backend to make the editor tool along with creating the login and registration pages.

- The cytoscape graph was stored in a `<div>` tag and a `cytoscape()` element named *cy* was attached to it
- The element *cy* was later used for all references to the cytoscape graph
- HTML forms for adding nodes/edges were handled with PHP which queried the MongoDB for storing the changes
- Only log of changes were stored in accordance to version control
- The interactive interface on right-clicking graph elements or the context menu [Conb] was created using jQuery and CSS

Backend: PHP II

- Clicking on Delete Node/Edge deletes the element from the graph using the **cy** variable. The corresponding change is stored in the MongoDB database
- AJAX requests were made to respective PHP scripts for storing log in database whenever a add/modify/delete request of a graph element is made
- The changes on the graph were visible instantly and were handled by the cytoscape.js API [Cona]

PHP MongoDB Connection

MongoDB was integrated with PHP using `MongoClient()` method

- All versions of a graph created a seperate collection (table) in the database
- The list of all graphs were stored in a seperate `user_graphs` collection for lookup purposes
- Tables were uniquely itentified using the user login credentials, graph which is worked upon and the version information
- `find()`, `remove()`, `count()` and `insert()` commands were used for querying, deleting, inserting and updating information from the PHP script
- Javascript and jQuery was used to call the PHP scripts using *AJAX* POST method

Style

Bootstrap [Boo] was used to style the web tool.

- Custom made CSS was used to create animation effects using jQuery for focussing on graph elements
- Popup dialog boxes for filling forms for adding/modifying/viewing graph elements were created using jQuery [cod] and CSS
- The forms, the login/registration page and overall styling was done using bootstrap
- Graph Layouts were added using the pre-defined functions in *cytoscape.js* API[Cona]

- Bootstrap.
  Css style.

- code.jquery.com.
  code.jquery.com.

- Cytoscape Consortium.
  cytoscape.js 2.3.5 web-api.

- Cytoscape Consortium.
  cytoscape.js context menu.

- The Gene Ontology Consortium.
  Gene ontology: tool for the unification of biology.

- MongoDB.
  Mongodb documentation manual.