Graph Algorithms for Visualizing High Dimensional Data

Abhinav Shankaranarayanan Venkataraman
Directed by : Prof. Ricard Gavalda and Prof. Marta Arias

Universitat Politecnica de Catalunya (UPC), Barcelona

27 June 2016

Outline

1. Introduction
2. Community Detection
3. Visualization Module
4. Overall system
5. Conclusion

Project Research Group

- This project is carried out within the LARCA research group at UPC .
- Researchers within LARCA have in the last two years began collaborations with hospital and health agencies for the analysis of electronic healthcare records [EHR].
- In previous work within the group, they proposed to organize the information in EHR in the form of graphs and hyper- graphs, which can then be navigated by experts and mined with graph and network theoretic tools.

What is Community?

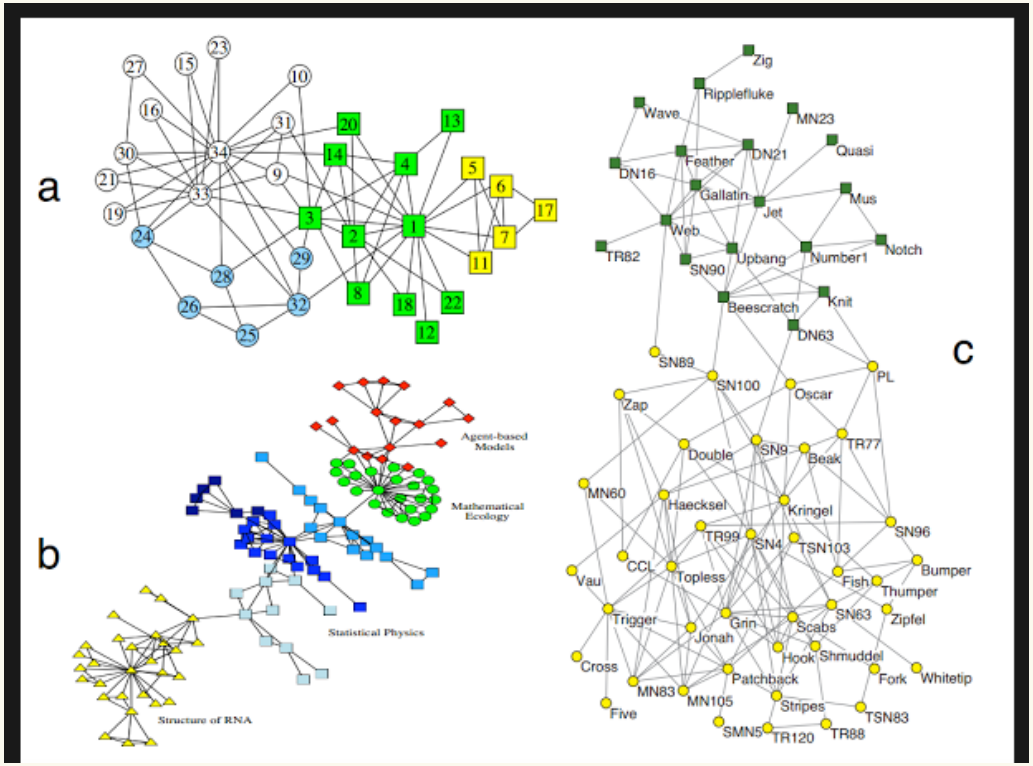


Image Source: Fortunato, Santo and Barthelemy, Marc 2007

Goal of the Project

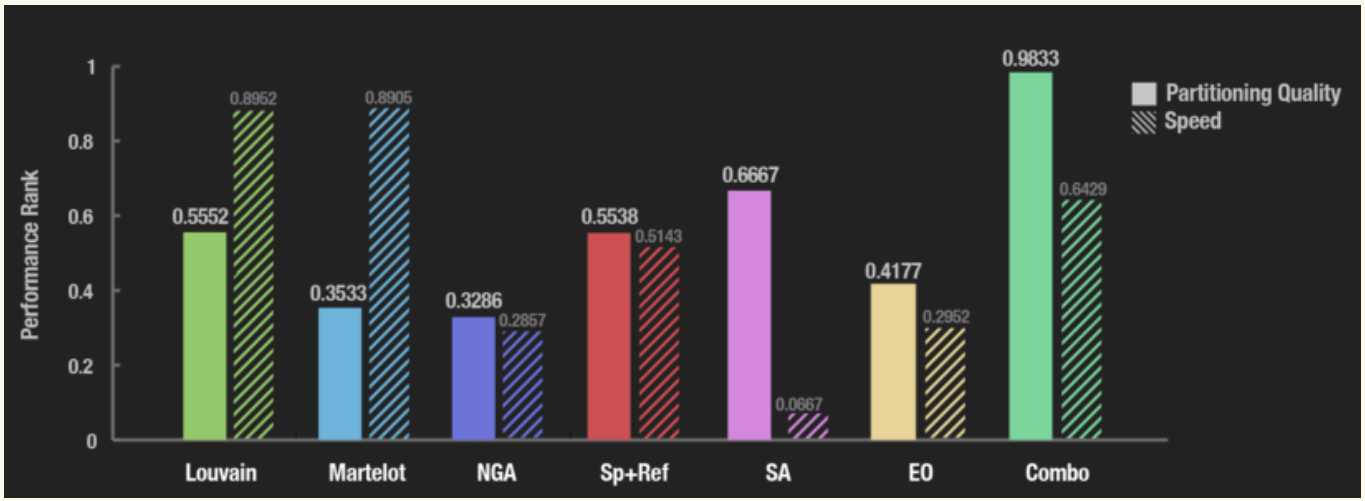
1. To survey a few algorithms that aim in community finding keeping in mind that the input is from the medical domain.
2. To choose an algorithms that benefit the purpose of organizing graphs from medical domain and for the purpose of visualization.
3. To implement the algorithms and test the efficiency of the algorithm using variety of graphs.
4. To build a Graphic User Interface (GUI) which enables visualization of the raw input on a web browser by drawing graphs.

Planning and Budget

1. Planning:
   - Required knowledge acquisition
   - Paper Analysis
   - Design and Implementation
   - Testing I
   - Testing II
   - Report Writing
2. Economic budget: Hardware budget, Software Budget, Human Resource Budget
3. Sustainability: Economically sustainable, Socially sustainable, Environmentally sustainable

State-of-the-art in Community Detection

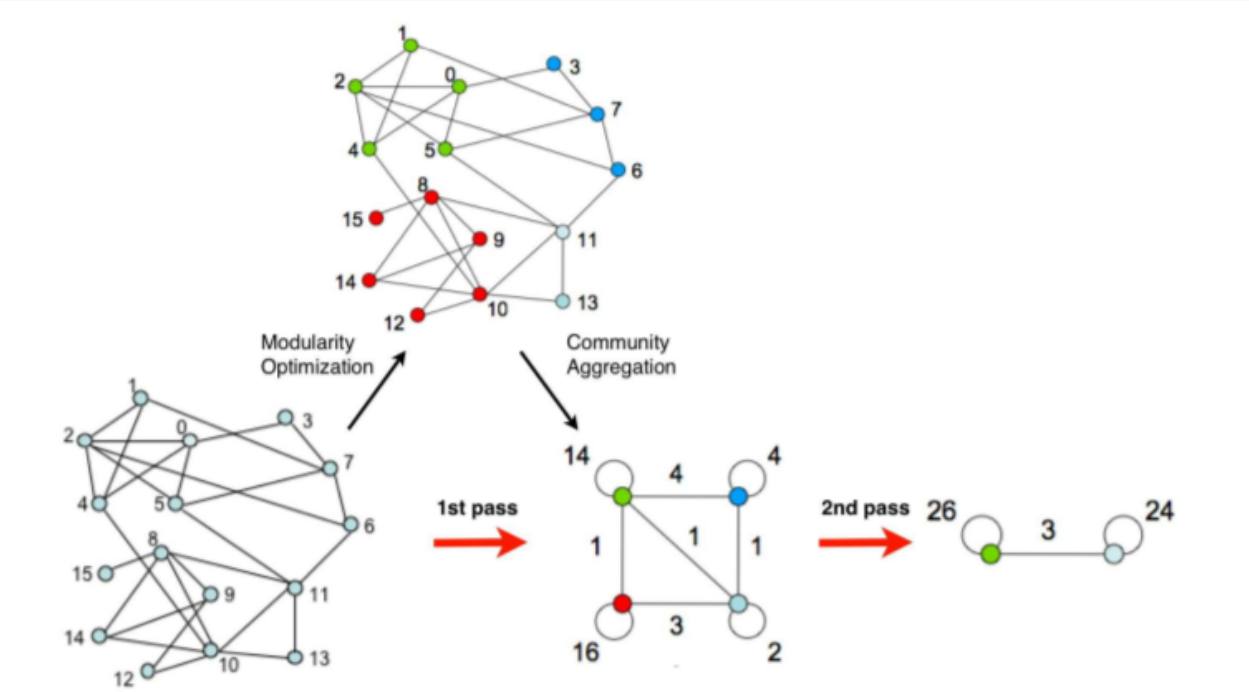
Communities are a part of the graph that have fewer ties with the rest of the system.
A community should be densely connected, well separated from the rest of the network and the members of the network should be more similar among themselves than with the rest.



Exploring state of the art: [3]

Louvain Algorithm [2]

Louvain algorithms is the state of the art community detection Algorithm. Louvain algorithm attempts to maximize modularity. This algorithm has two phases. The diagram shows the



Visualization of the steps of our algorithm. This was taken from the paper "Fast unfolding of communities in large networks" [1]

Louvain Algorithm Pseudocode

Louvain Algorithm Pseudocode:

1. Repeat until local optimum is reached:
   1. Phase1 : Split or partition the graph by optimizing modularity greedily
   2. Phase2 : Agglomerate the found clusters into new nodes

First phase in Louvain

Louvain Algorithm Pseudocode for Phase1:

1. Assign a different community to each node.
2. For each node v_i
   - For each $v_j \in N(v_i)$,consider removing v_i from community of v_i and place it in the community of v_j
   - Choose v_i into community of neighbour that leads to highest modularity gain (Greedy Choice).
3. Repeat until no improvement can be done

Second phase in Louvain

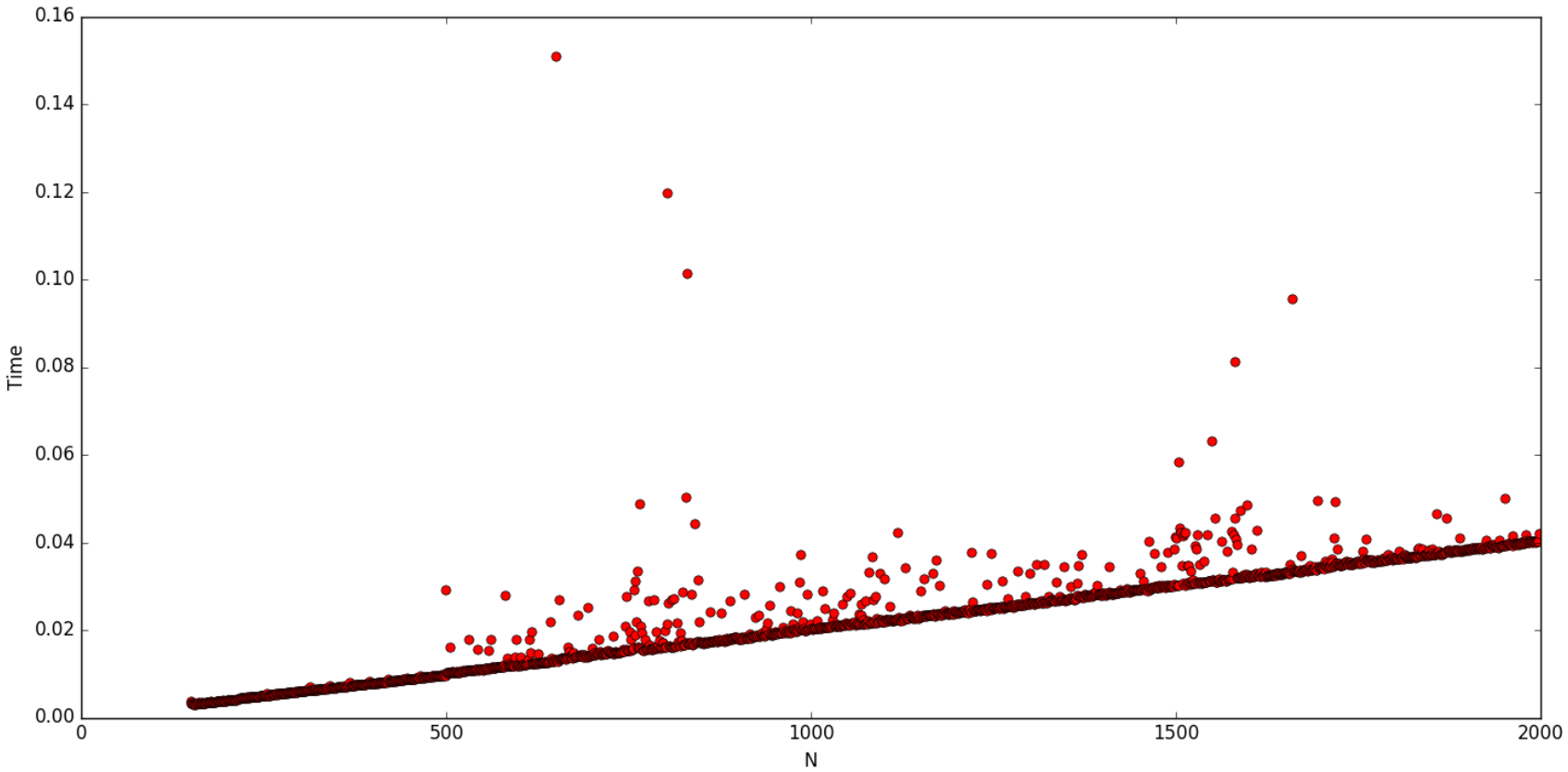
Louvain Algorithm Pseudocode for Phase2:

1. Let each community C_i form a new node v_i
2. Let the edges between new nodes v_i and v_j be the sum of edges between nodes in C_i and C_j in the previous graph

Mode of implementation

1. The implementation of the Algorithm is in Python. pyLouvain is code that is freely available although considering the task performed by the project it is tough to use pyLouvain directly. Hence modifications were made to pyLouvain and some part of the code was reused.
2. The input data structure was altered. The Input file is stored in a matrix and its transpose is used to get the node set. This is used to for a edge dictionary.
3. The first phase of pyLouvain is used as it is in the project and the second phase has been modified in the manner that relabelling is done.

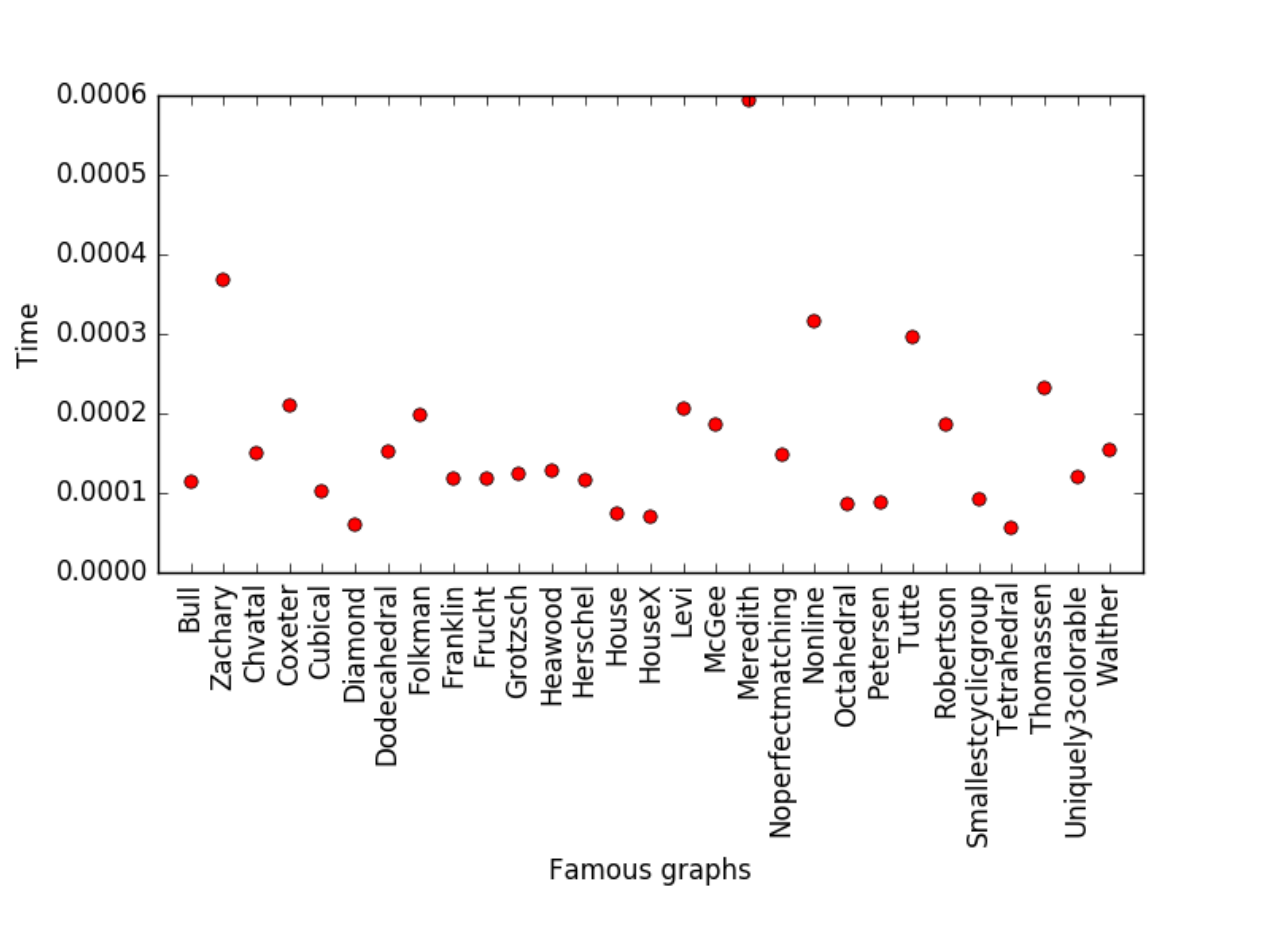
N until 2000 and Q=0.4, Scale-Free degree distribution Experiments



SNAP Experiments

Description	Nodes	Edges	Communities Known	Time	Note
Email network from a EU research institution	265214	420045	---	2.24	
DBLP collaboration network	317080	1049866	13477	5.58	Increase in number of edges
Amazon product network	334863	925872	75149	5.75	Increase in number of ground-truth communities
Youtube online social network	1134890	2987624	8385	17.54	Increase in number of edges by a factor of 3

Famous Graphs Experiments

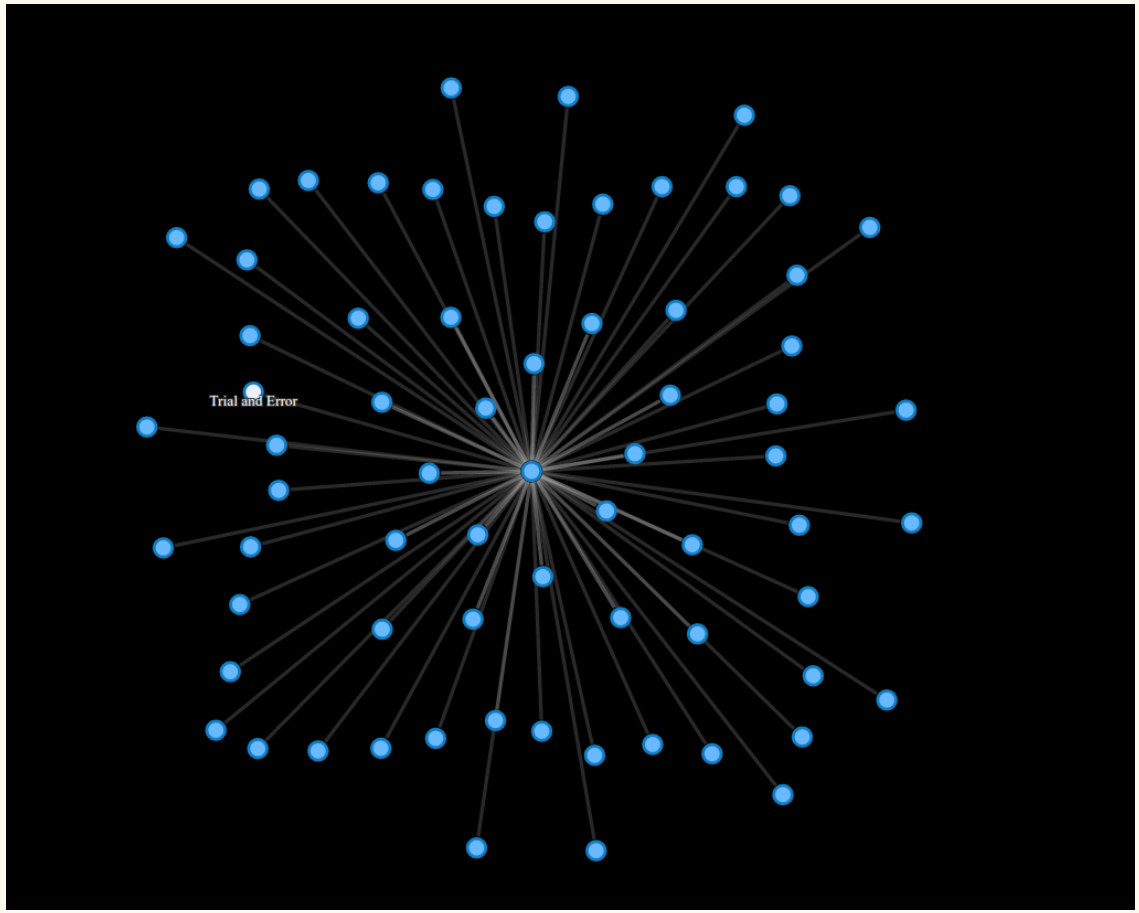


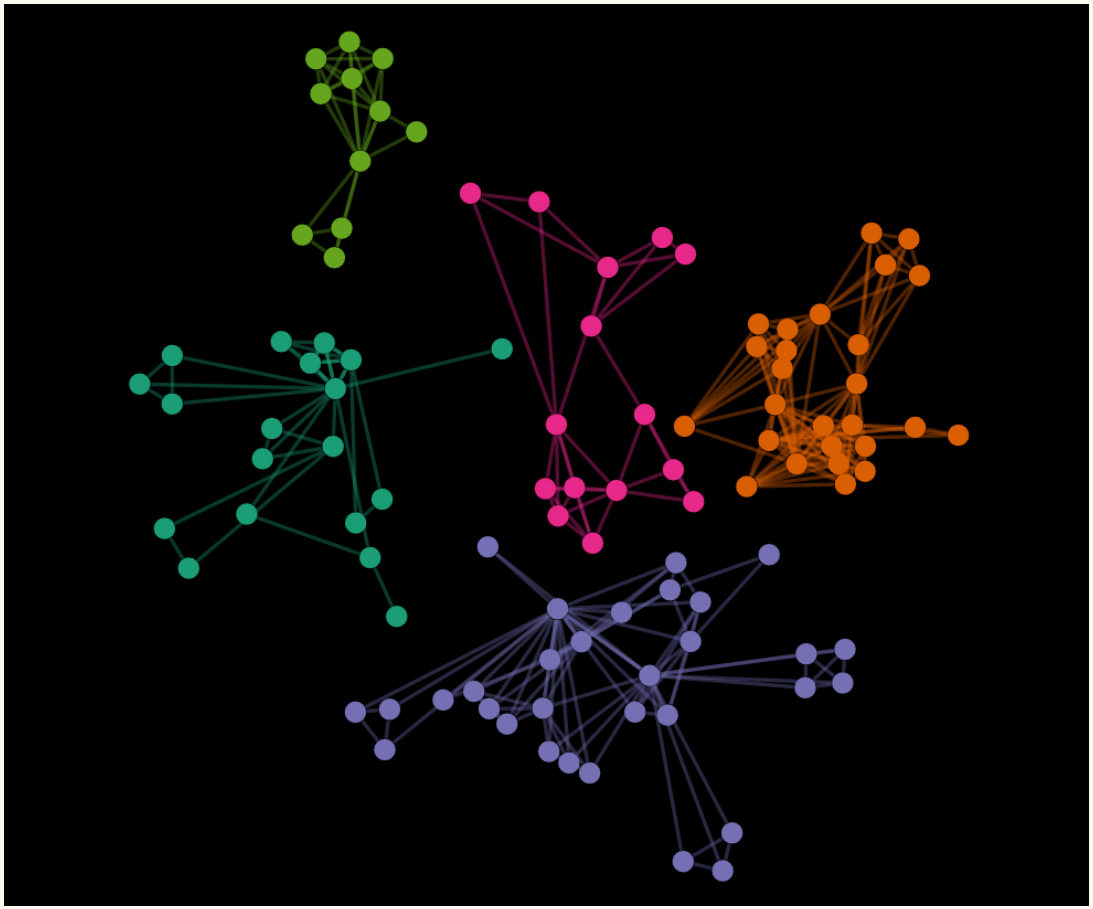
Visualization Libraries

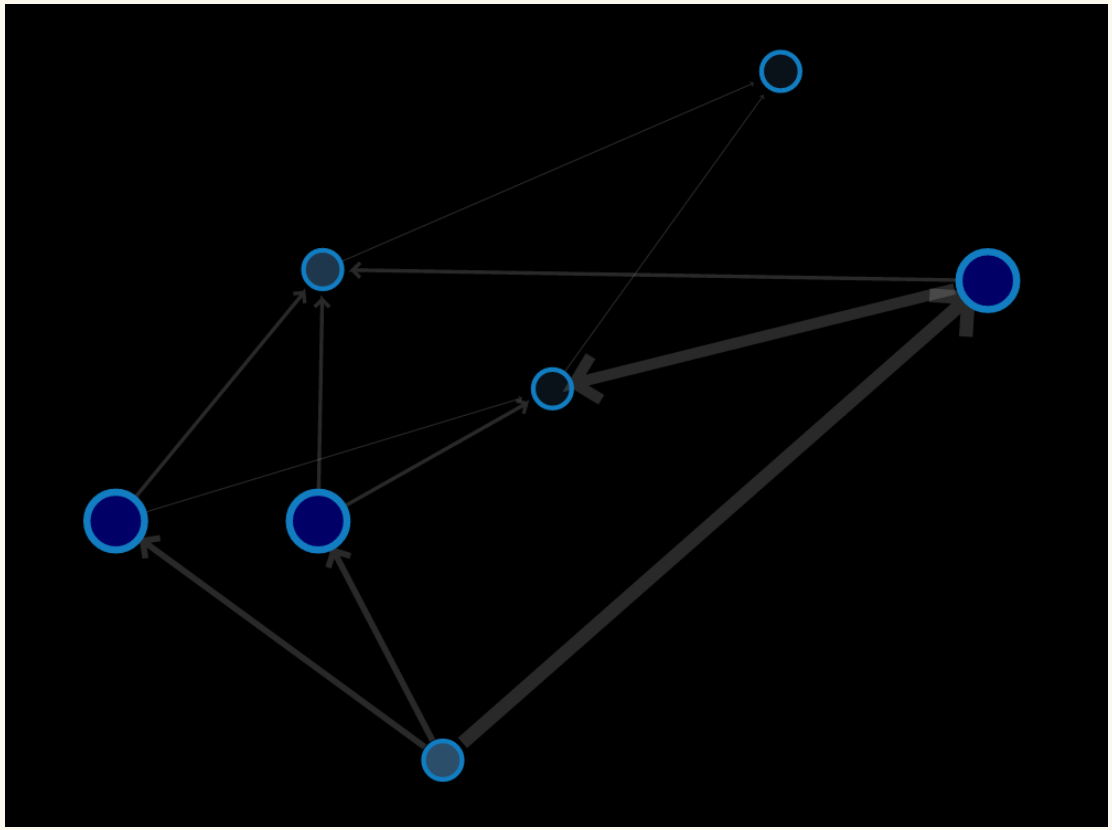
	Protovis.js	D3.js	Alchemy.js	Gephi
JavaScript	✓	✓	✓	
JSON Object	✓	✓	✓	
Robust		✓		✓
Less Overhead			✓	

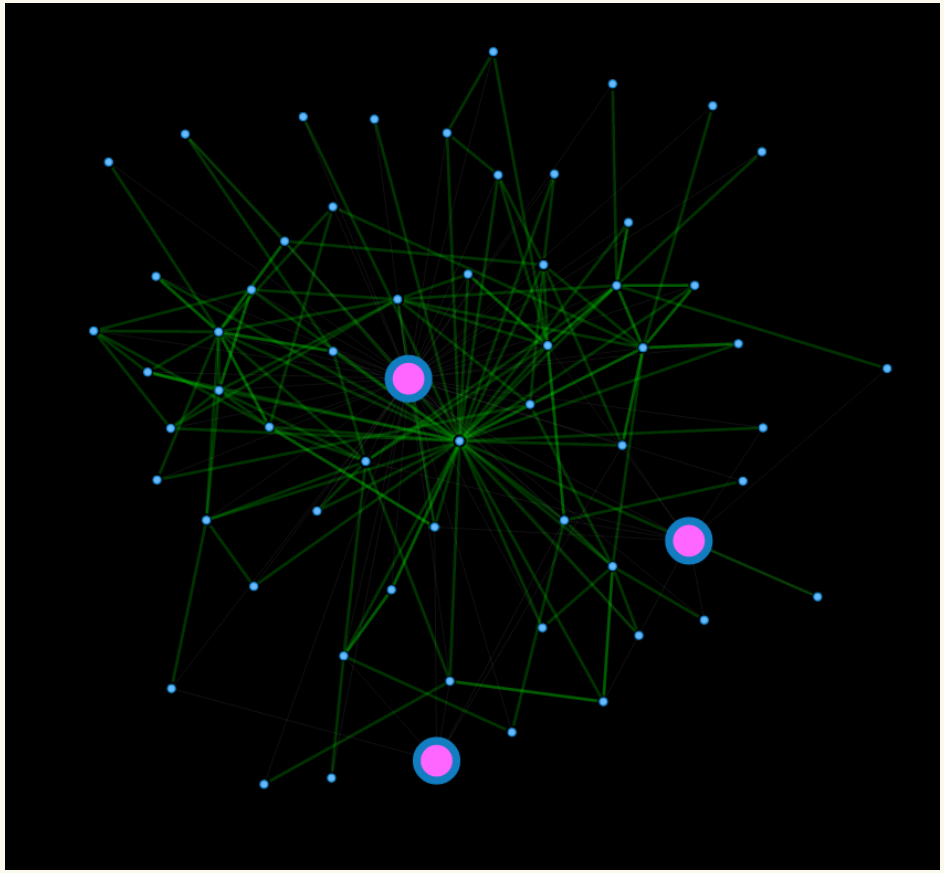
Table : Comparing Visualization methods

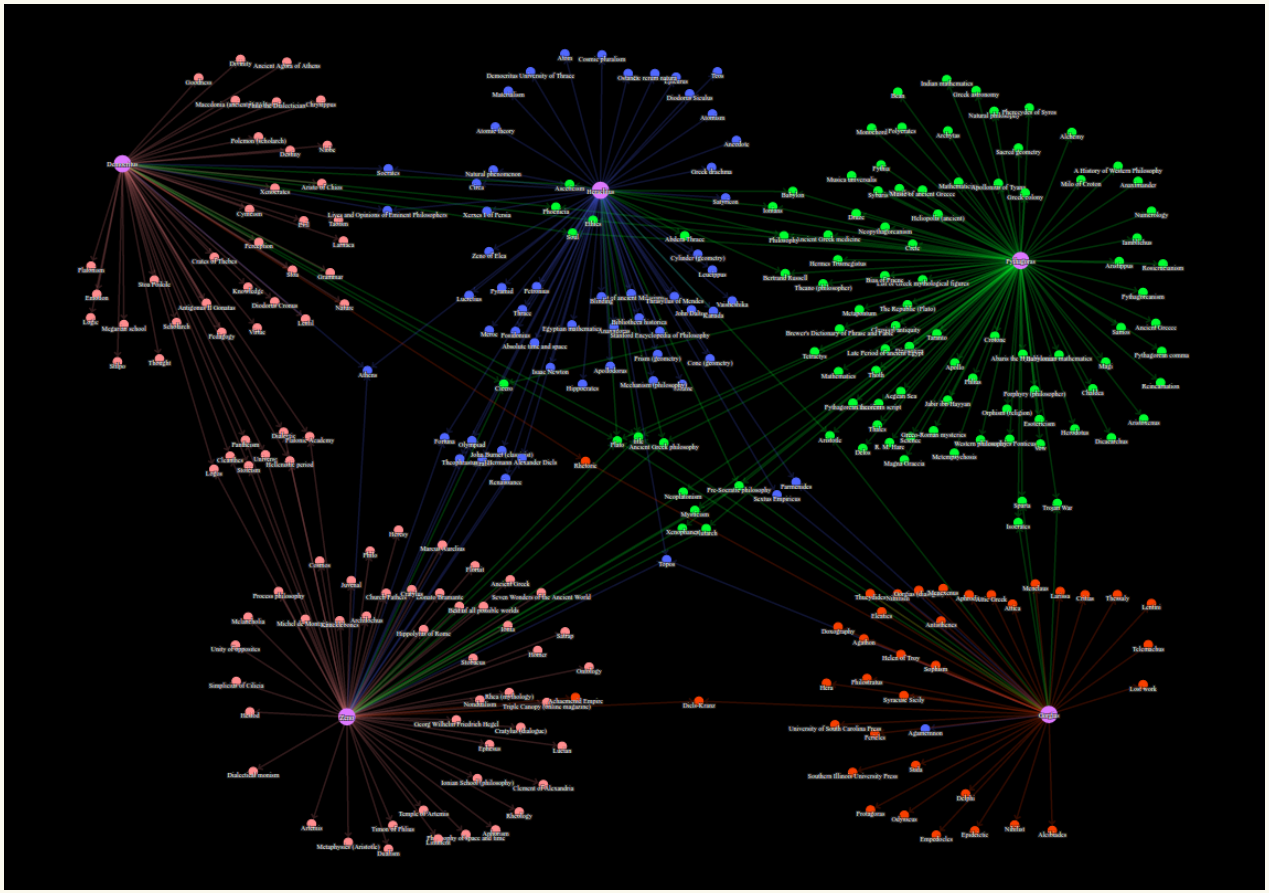
1. Alchemy needs three main units to form as an application namely: alchemy.css, alchemy.js and data.
2. Five simple steps to connect the JSON object to draw the graph.
3. Tests:

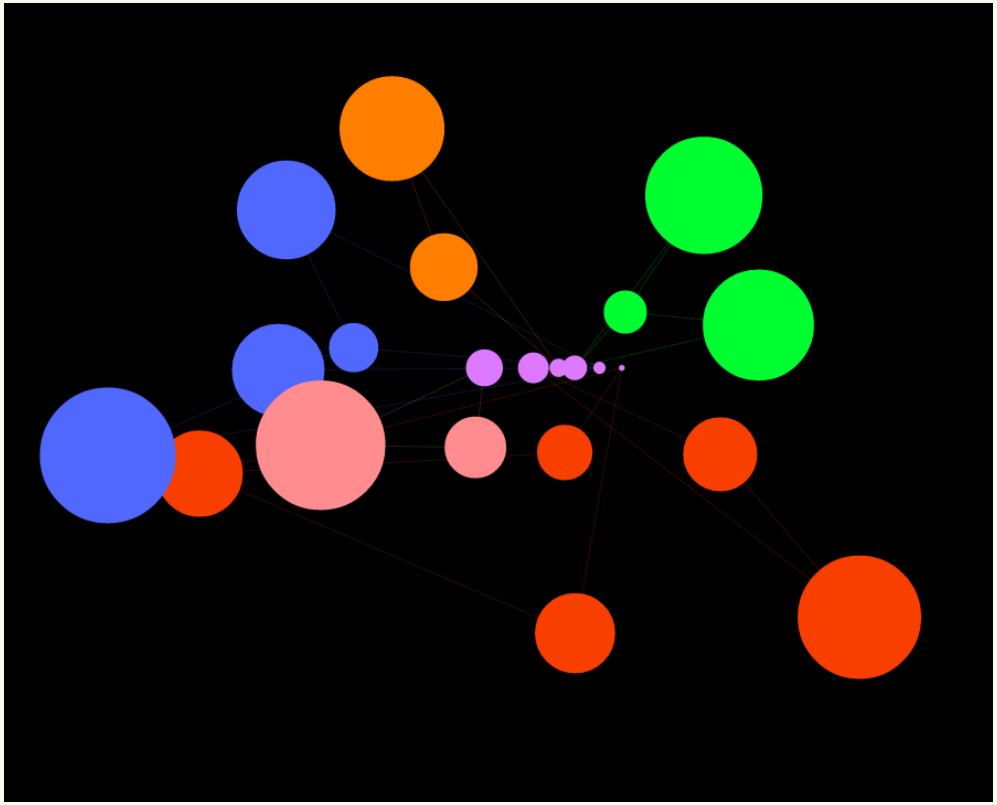








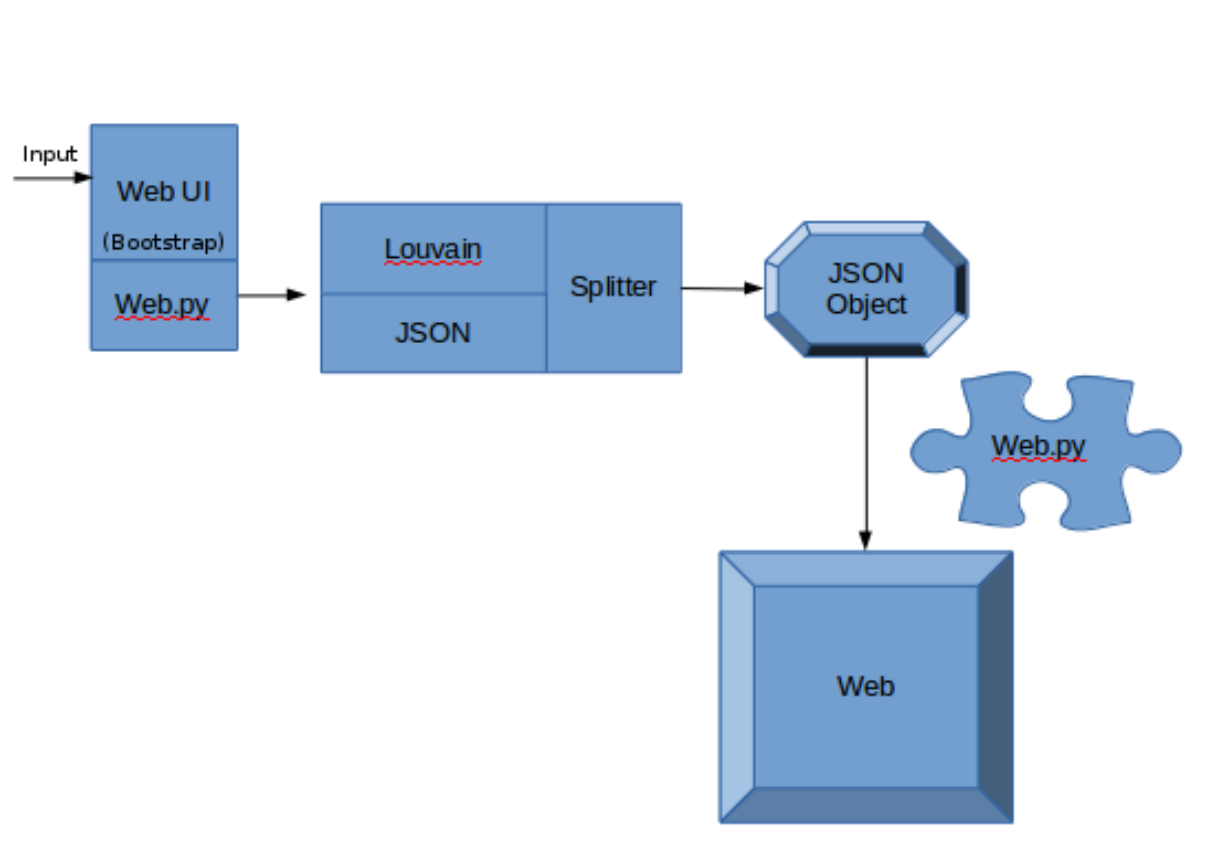




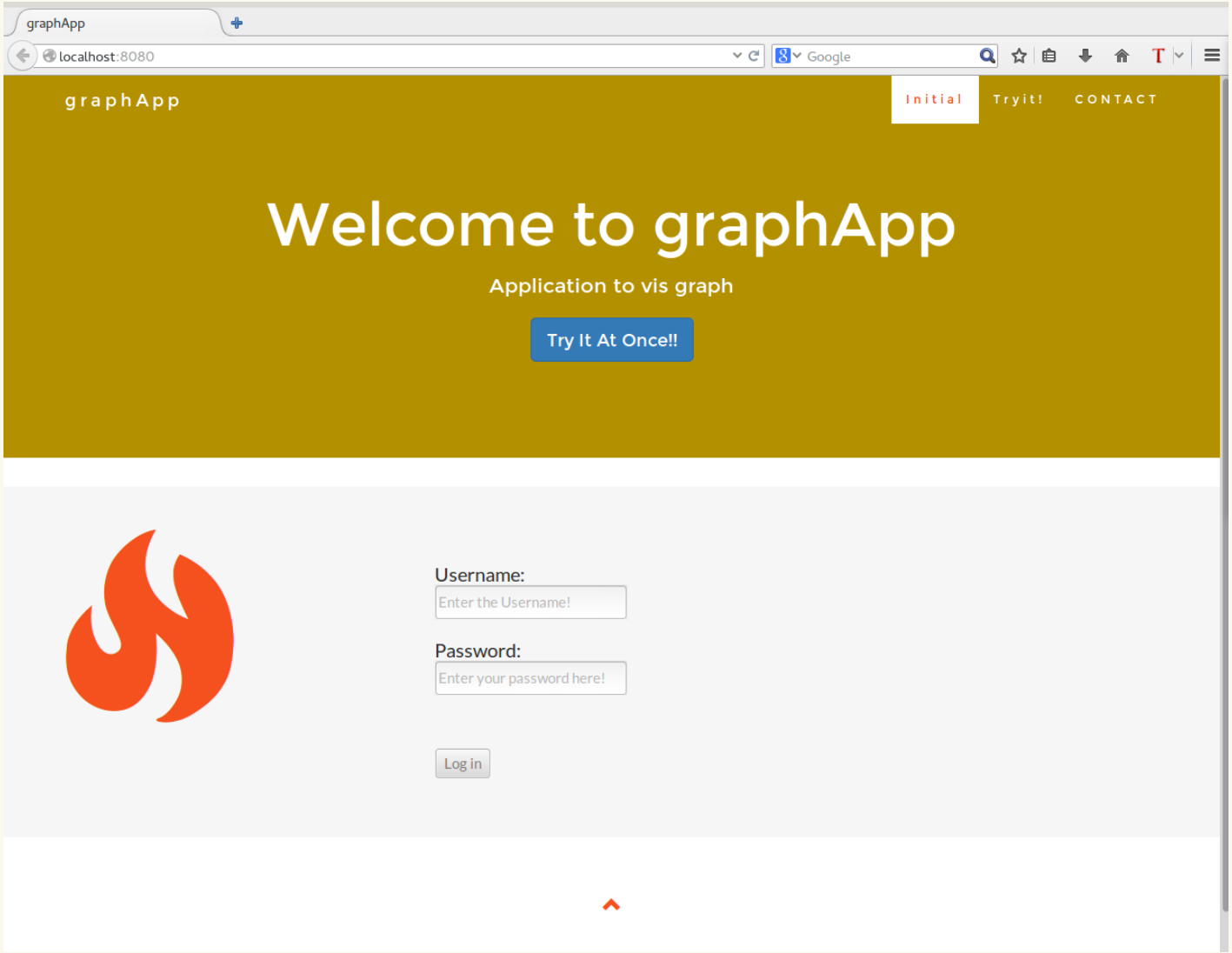
Web framework and Front-end

1. Angular.js was recommended by the main project
2. Django, Grok and Web.py were considered.
3. Bootstrap was used as the front-end framework. It was used for the front-end contrasting with web.py which was used behind.

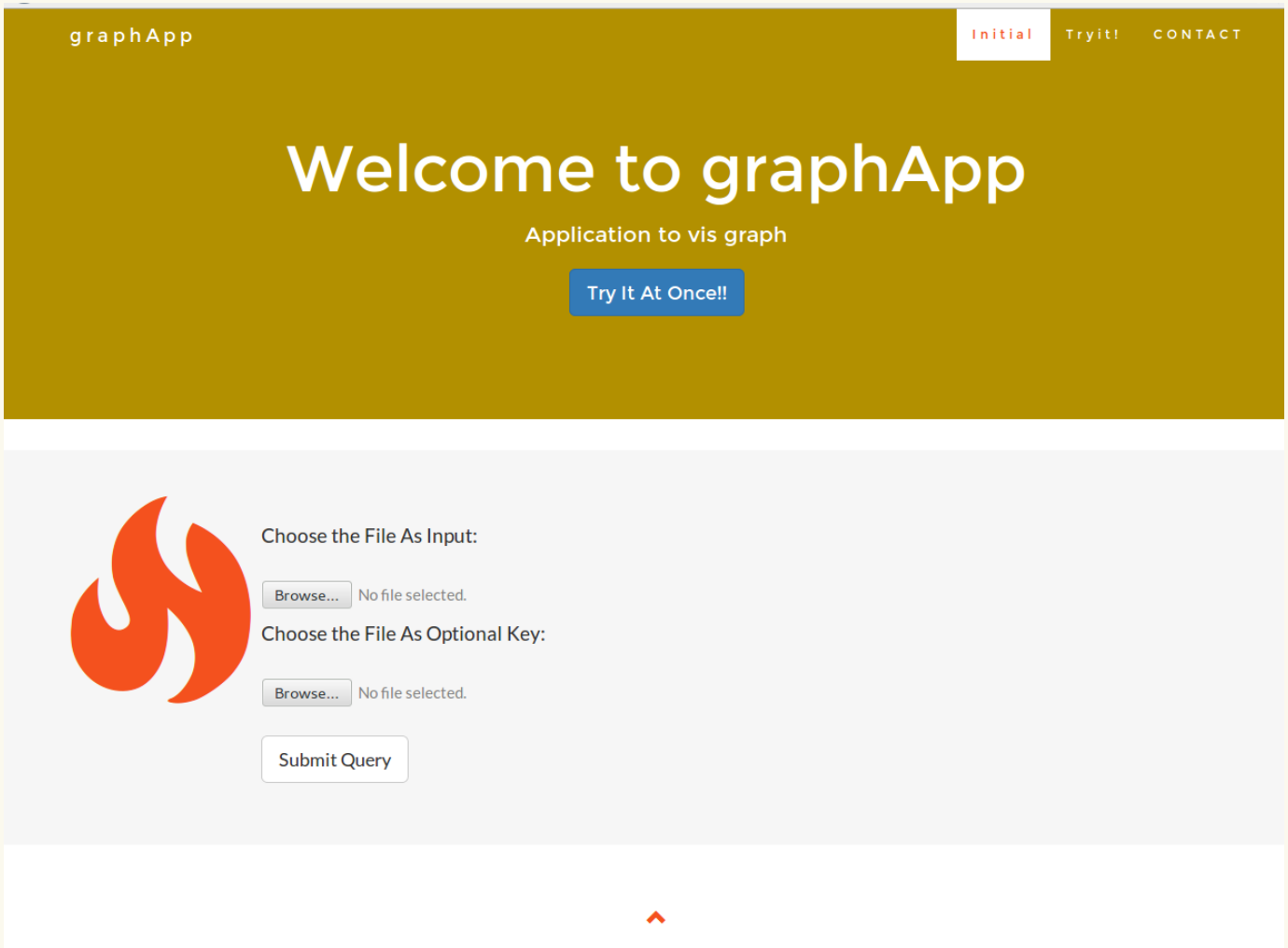
How it works?


Input
Web UI
(Bootstrap)
Web.py
Louvain
JSON
Splitter
JSON
Object
Web.py
Web

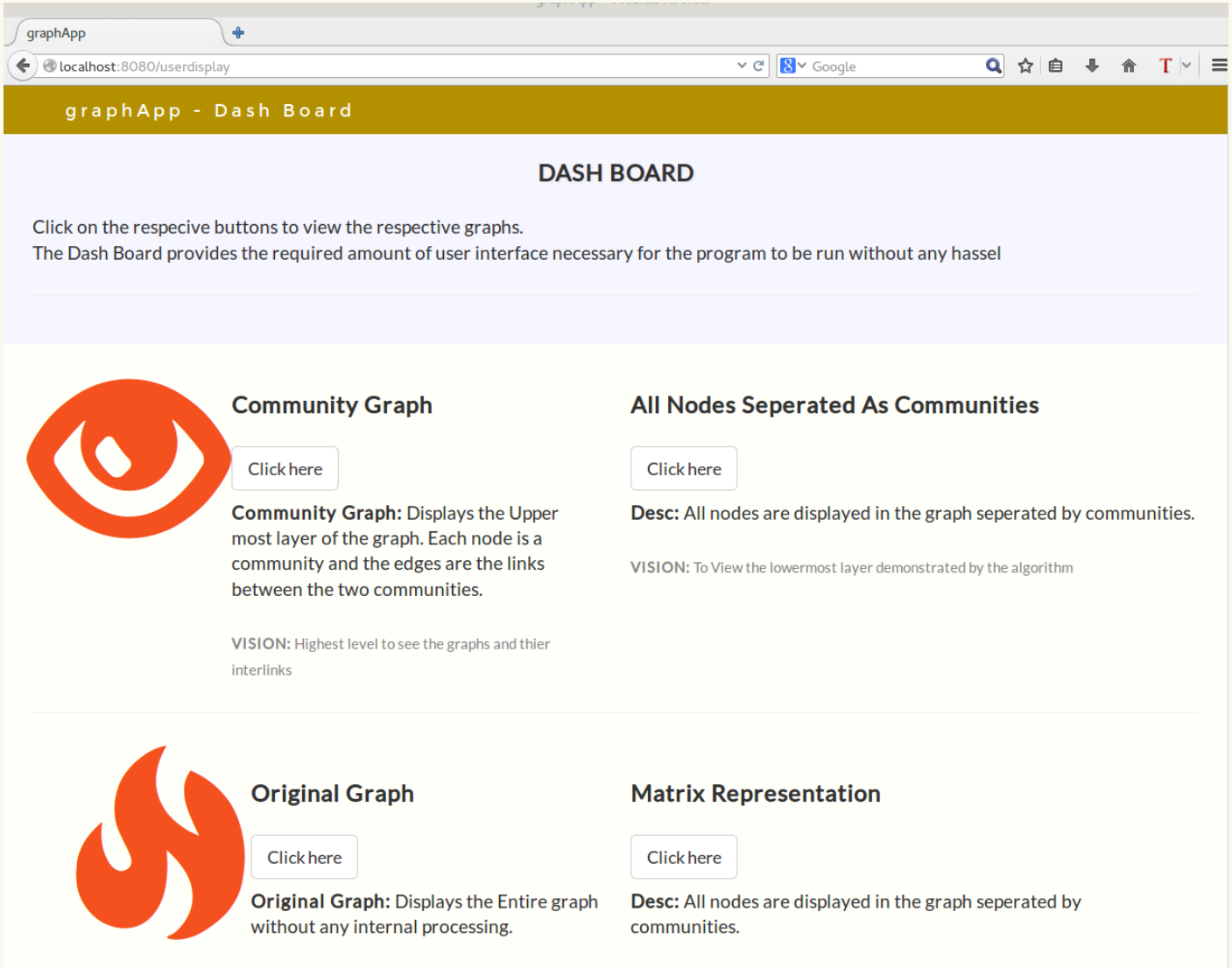
How it works?


graphApp
localhost:8080
Google
graphApp
Initial
Tryit!
CONTACT
Welcome to graphApp
Application to vis graph
Try It At Once!!
Username:
Enter the Username!
Password:
Enter your password here!
Log in

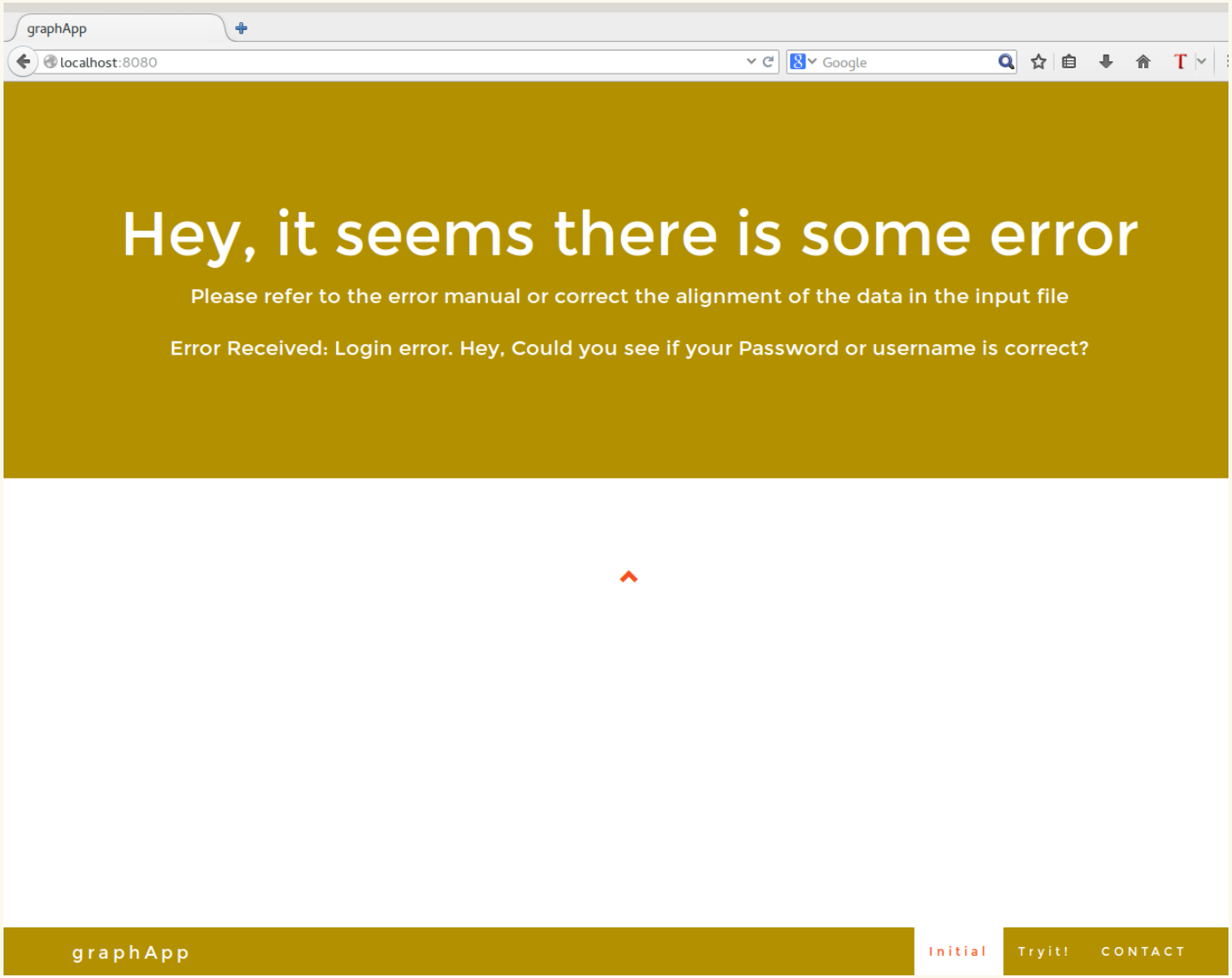
How it works?


graphApp
Initial
Tryit!
CONTACT
Welcome to graphApp
Application to vis graph
Try It At Once!!
Choose the File As Input:
Browse...
No file selected.
Choose the File As Optional Key:
Browse...
No file selected.
Submit Query

How it works?


graphApp
localhost:8080/userdisplay
Google
graphApp - Dash Board
DASH BOARD
Click on the respecive buttons to view the respective graphs.
The Dash Board provides the required amount of user interface necessary for the program to be run without any hassel
Community Graph
Click here
Community Graph: Displays the Upper most layer of the graph. Each node is a community and the edges are the links between the two communities.
VISION: Highest level to see the graphs and thier interlinks
All Nodes Seperated As Communities
Click here
Desc: All nodes are displayed in the graph seperated by communities.
VISION: To View the lowermost layer demonstrated by the algorithm
Original Graph
Click here
Original Graph: Displays the Entire graph without any internal processing.
Matrix Representation
Click here
Desc: All nodes are displayed in the graph seperated by communities.

How it works?


graphApp
localhost:8080
Google
Hey, it seems there is some error
Please refer to the error manual or correct the alignment of the data in the input file
Error Received: Login error. Hey, Could you see if your Password or username is correct?
graphApp
Initial
Tryit!
CONTACT

Conclusion

1. In the project we have surveyed a few algorithms that aim in community finding keeping in mind that the input is taken from health care domain.
2. Louvain Community detection algorithm was chosen for community detection and for visualization. Alchemy.js were selected after considering the input and a few state of the art algorithms.
3. The algorithms and frameworks thus found were implemented and tests were conducted for finding the efficiency of the algorithms.
4. A GUI implementing Web.py and Bootstrap was created combining the visualization and the computation.

Challenges

There were quite a few challenges in the project. I was from a pure theoretical background, so I had to learn all the following from scratch:

1. Python
2. JavaScript
3. WebFrame works - Django, web.py
4. Learning Alchemy.js and web.py
5. I had some experience in Linux, git and github

Personal Learning

The project gave a huge scope for exploring many softwares and trying out several of them. It gave an opportunity for me to learn Python to depth. My interest for Data visualization and graph algorithms has increased. By trying a web application I got accustomed to using web frameworks and web technologies.

List of References that were used

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre.
Fast unfolding of communities in large networks.
Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008.

Santo Fortunato.
Community detection in graphs.
Physics reports, 486(3):75–174, 2010.

Stanislav Sobolevsky, Riccardo Campari, Alexander Belyi, and Carlo Ratti.
General optimization technique for high-quality community detection in complex networks.
Physical Review E, 90(1):012811, 2014.

Thank you

Thank you for all those who supported me throughout the project.
It was a Great time at Barcelona working with Prof. Ricard and Prof. Marta.