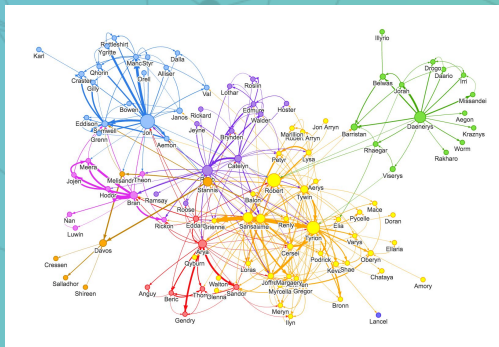


Leveraging Graph Algorithms In Visualizations With Neovis.js

William Lyon
@lyonwj
lyonwj.com



NODES 2019

Neo4j Online Developer
Expo and Summit

bit.ly/neovisalgos

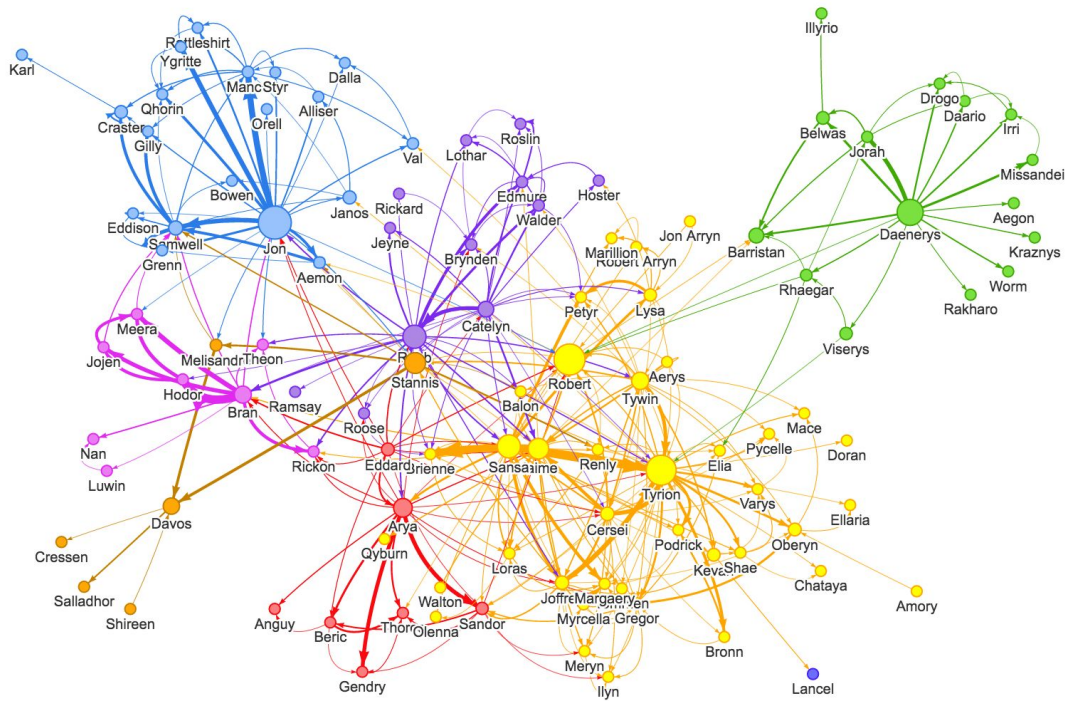


William Lyon

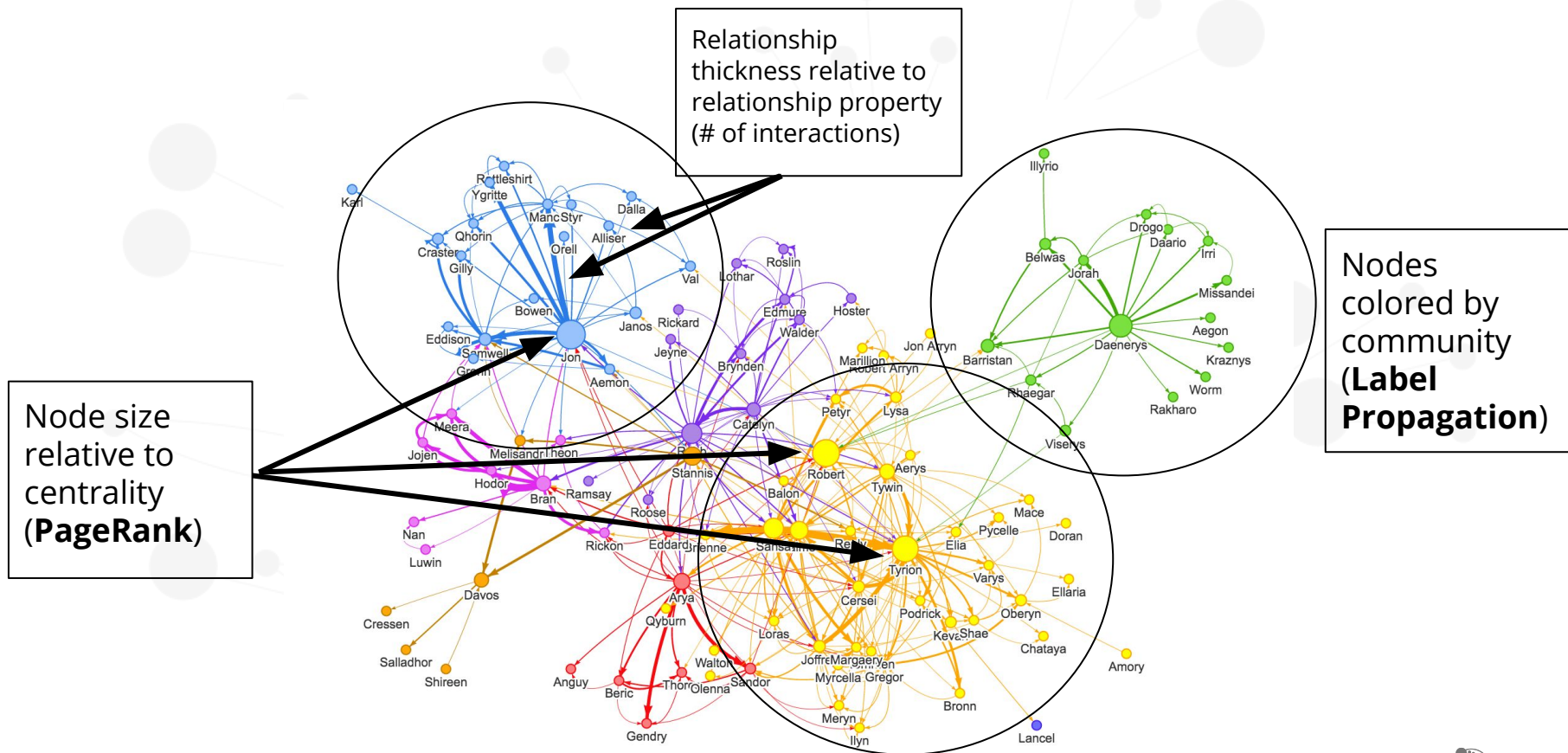
@lyonwj

Neo4j Labs Engineer

Graph Visualization + Graph Algorithms



Graph Visualization + Graph Algorithms



+40 Graph & ML Algorithms in Neo4j



Pathfinding & Search

Finds optimal paths or evaluates route availability and quality



Centrality / Importance

Determines the importance of distinct nodes in the network



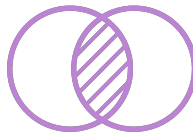
Community Detection

Detects group clustering or partition options



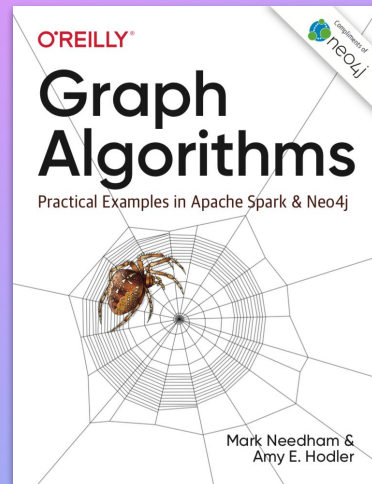
Link Prediction

Estimates the likelihood of nodes forming a future relationship



Similarity

Evaluates how alike nodes are



neo4j.com/graph-algorithms-book/

Graph and ML Algorithms in Neo4j



Pathfinding & Search

- Parallel Breadth First Search & DFS
- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- Minimum Spanning Tree
- A* Shortest Path
- Yen's K Shortest Path
- K-Spanning Tree (MST)
- Random Walk



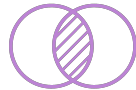
Centrality / Importance

- Degree Centrality
- Closeness Centrality
- CC Variations: Harmonic, Dangalchev, Wasserman & Faust
- Betweenness Centrality
- Approximate Betweenness Centrality
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality



Community Detection

- Triangle Count
- Clustering Coefficients
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity – 1 Step & Multi-Step
- Balanced Triad (identification)



Similarity

- Euclidean Distance
- Cosine Similarity
- Jaccard Similarity
- Overlap Similarity
- Pearson Similarity



Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

neo4j.com/docs/graph-algorithms/current/

Updated April 2019

How To...

1. Call as Cypher procedure

2. Pass in specification (Label, Prop, Query) and configuration

3. stream variant returns (a lot) of results

```
CALL algo.<name>.stream('Label','TYPE',{conf})  
YIELD nodeId, score
```

4. non-stream variant writes results to graph returns statistics

```
CALL algo.<name>('Label','TYPE',{conf})
```



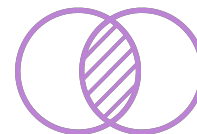
Pathfinding
& Search



Community
Detection



Centrality /
Importance



Similarity

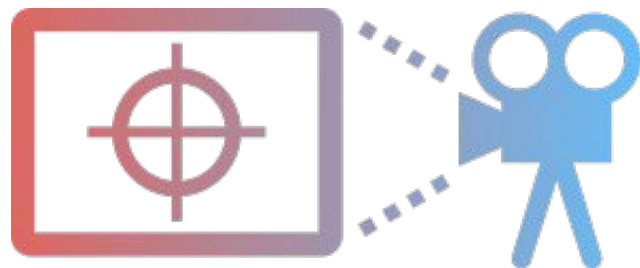


Link
Prediction

Cypher Projection

Pass in Cypher statement for node- and relationship-lists.

```
CALL algo.<name>(
  'MATCH ... RETURN id(n)',
  'MATCH (n)-->(m)
  RETURN id(n) as source,
         id(m) as target', {graph:'cypher'})
```



Cypher Projection Example

Russian Twitter Trolls

TECH DEC 20 2017, 11:11 AM ET

Russian trolls went on attack during key election moments

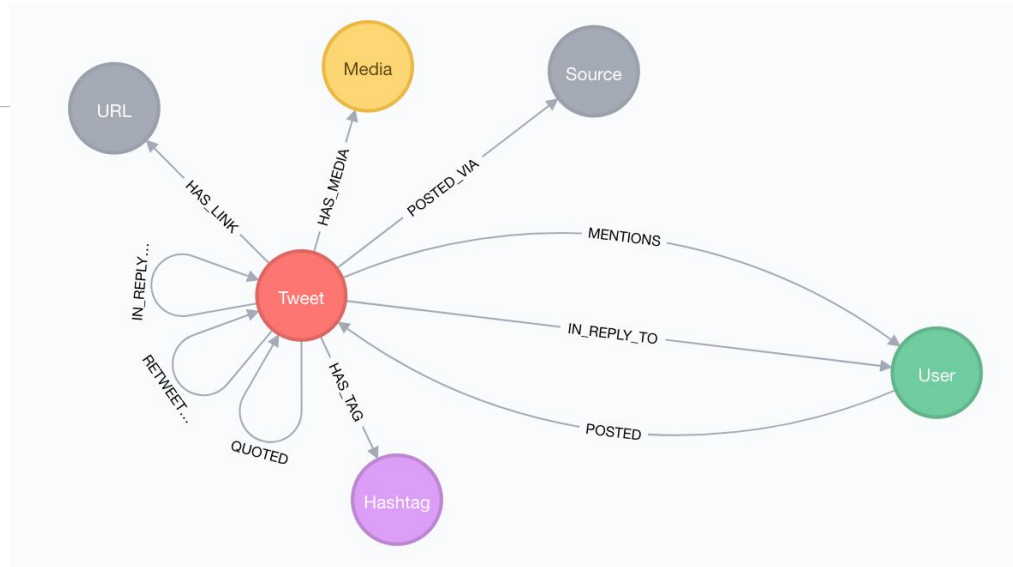
by BEN POPKEN



Ben Popken
@bpopken

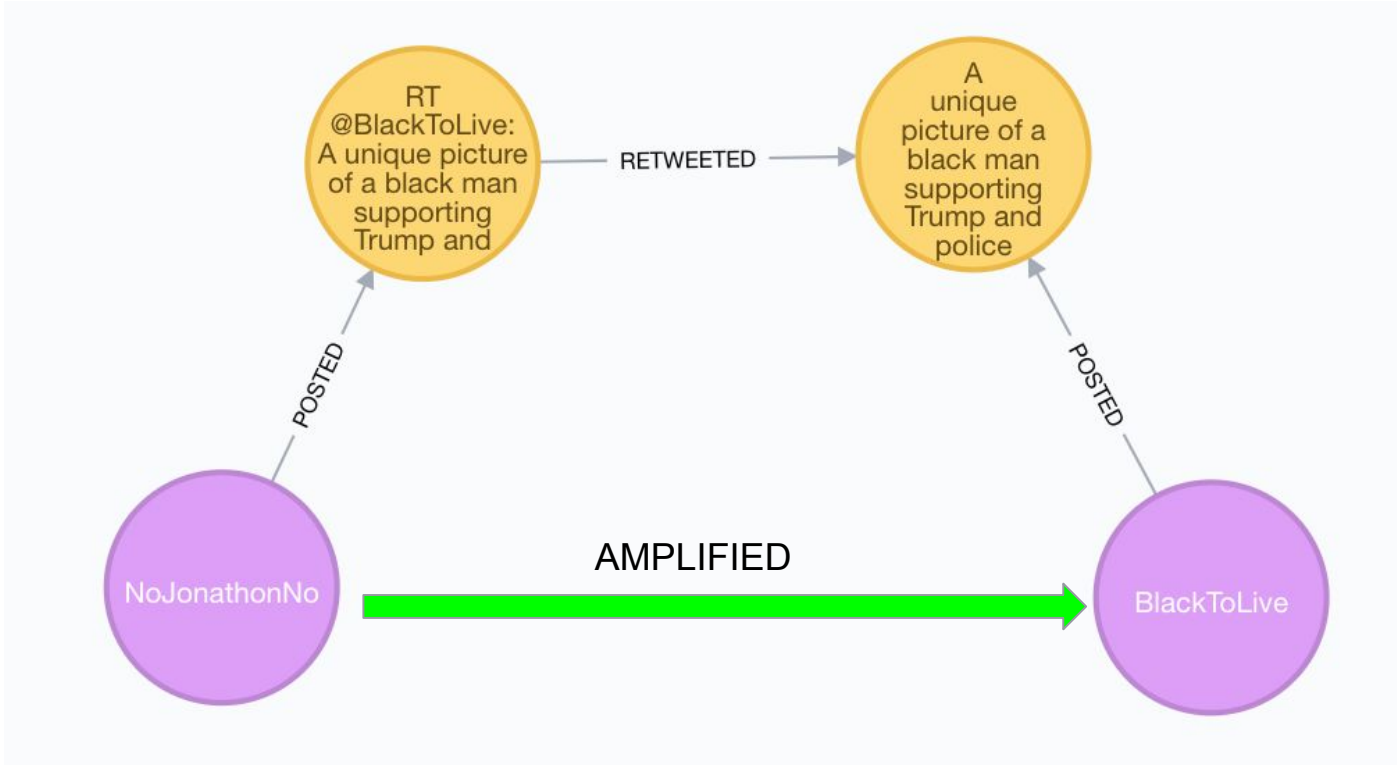
Following

Huge props and thank you to @neo4j and their @mdavidallen and @lyonwj for helping compile and analyze the deleted twitter data, surfacing trends and uncovering new angles.



Inferred Relationships

```
1 MATCH (r1:Troll)-[:POSTED]->(t1:Tweet)<-[:RETWEETED]-(t2:Tweet)<-[:POSTED]-(r2:Troll)
```

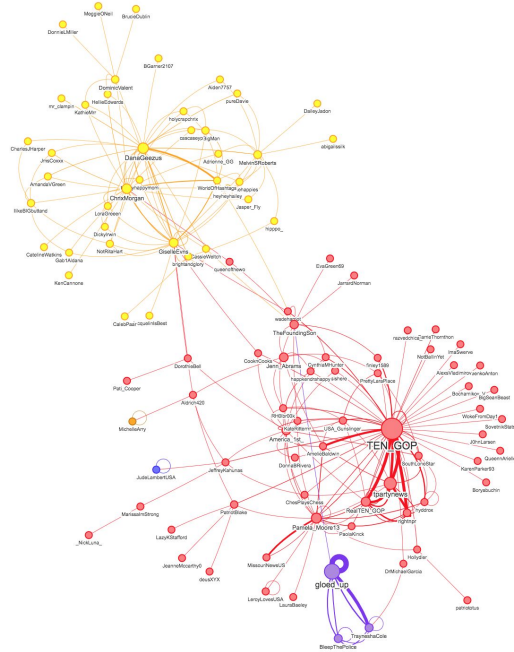


PageRank on Inferred AMPLIFIED Graph

```
CALL algo.pageRank("MATCH
```

```
(r1:Tro11)-[:POSTED]->(:Tweet)<-[:RETWEETED]-(:Tweet)<-[:POSTED]- (r2:Tro11)
```

```
RETURN id(r2) as source, id(r1) as target", {graph:'cypher'})
```




neo4j\$

neo4j\$:play https://guides.neo4j.com/sandbox/twitter-trolls/index.html

Russian Twitter Trolls


As part of the House Intelligence Committee investigation into **how Russia may have influenced the 2016 US Election**, Twitter released the screen names of almost 3000 Twitter accounts believed to be connected to Russia's Internet Research Agency, a company known for operating **social media troll accounts**. Twitter immediately suspended these accounts, deleting their data from Twitter.com and the Twitter API.

A team at NBC News including **Ben Popken** and **EJ Fox** was able to reconstruct a dataset consisting of a subset of the deleted data for their investigation and using Neo4j were able to **show how these troll accounts went on attack during key election moments**. **NBC News open-sourced the reconstructed dataset** and released it as this Neo4j database.



This interactive Neo4j Browser guide will help you to explore the dataset of Russian Troll tweets by guiding you through:

- an overview of the datamodel
- how to explore the data using Cypher, the query language for graphs
- some of the investigative queries used to make sense of the the dataset

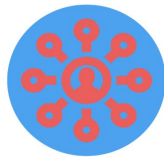


Read the NBC News story about the data [here](#)

NOTE: Much of this content is offensive and explicit. Query with caution.

1 / 16

Russian Twitter Trolls Get Started **Details** Data Model Code Advanced



Neo4j Browser: <https://10-0-1-230-37029.neo4jsandbox.com/>

Direct Neo4j HTTP: <http://100.26.17.167:37029/browser/>

Username: neo4j

Password: lapse-narcotics-tumarounds

JavaScript Bolt URL: bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443

IP Address: 100.26.17.167

HTTP Port: 37029

Bolt Port: 37028

Expires: 8 days, 20 hours, 43 minutes

neo4jsandbox.com

neovis.js

npm install neovis.js

neovis.js

1.2.1 • Public • Published a day ago

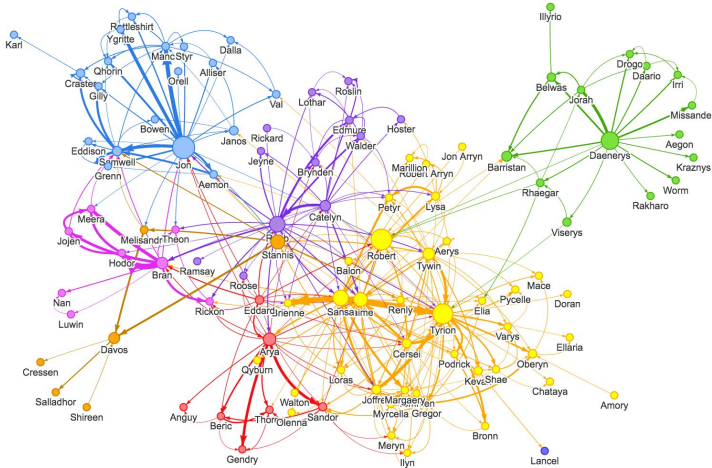
Readme

3 Dependencies

neovis.js

🟢 PASSED npm package 1.2.1

Graph visualizations powered by vis.js with data from Neo4j.



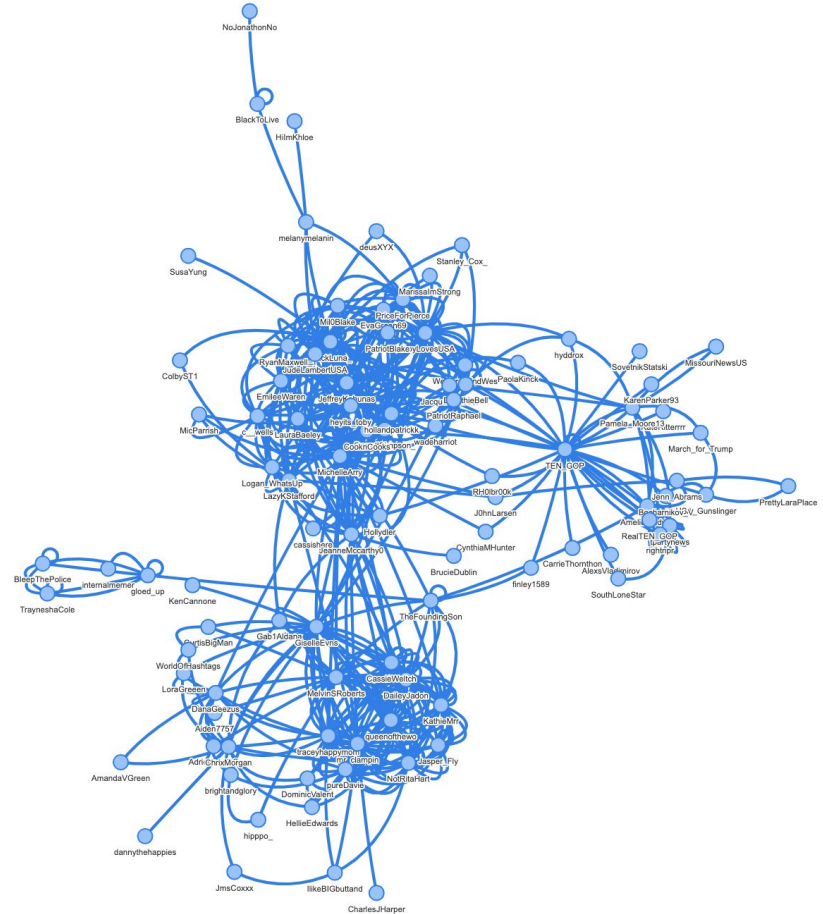
```
var config = {
  container_id: "viz",
  server_url: "bolt://localhost:7687",
  server_user: "neo4j",
  server_password: "sorts-swims-burglaries",
  labels: {
    "Character": {
      "caption": "name",
      "size": "pagerank",
      "community": "community"
    }
  },
  relationships: {
    "INTERACTS": {
      "thickness": "weight",
      "caption": false
    }
  },
  initial_cypher: "MATCH (n)-[r:INTERACTS]->(m) RETURN *"
};
```

```
viz = new NeoVis.default(config);
viz.render();
```

<https://www.npmjs.com/package/neovis.js>

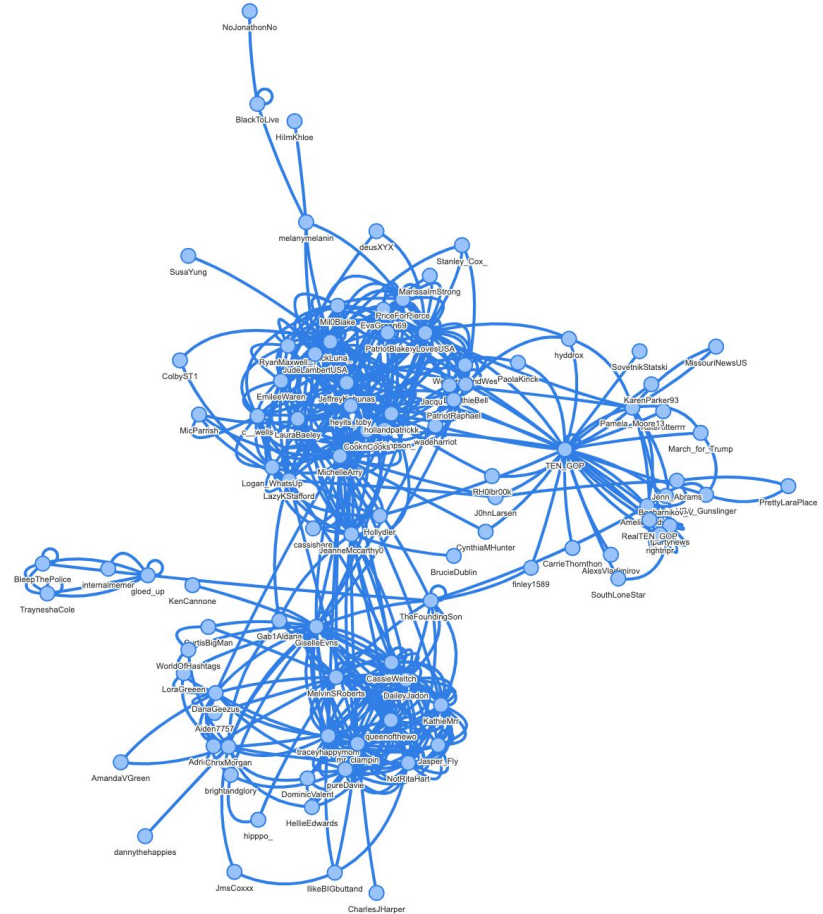
Initial visualization

```
var config = {
  container_id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name"
    }
  },
  relationships: {
    AMPLIFIED: {
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



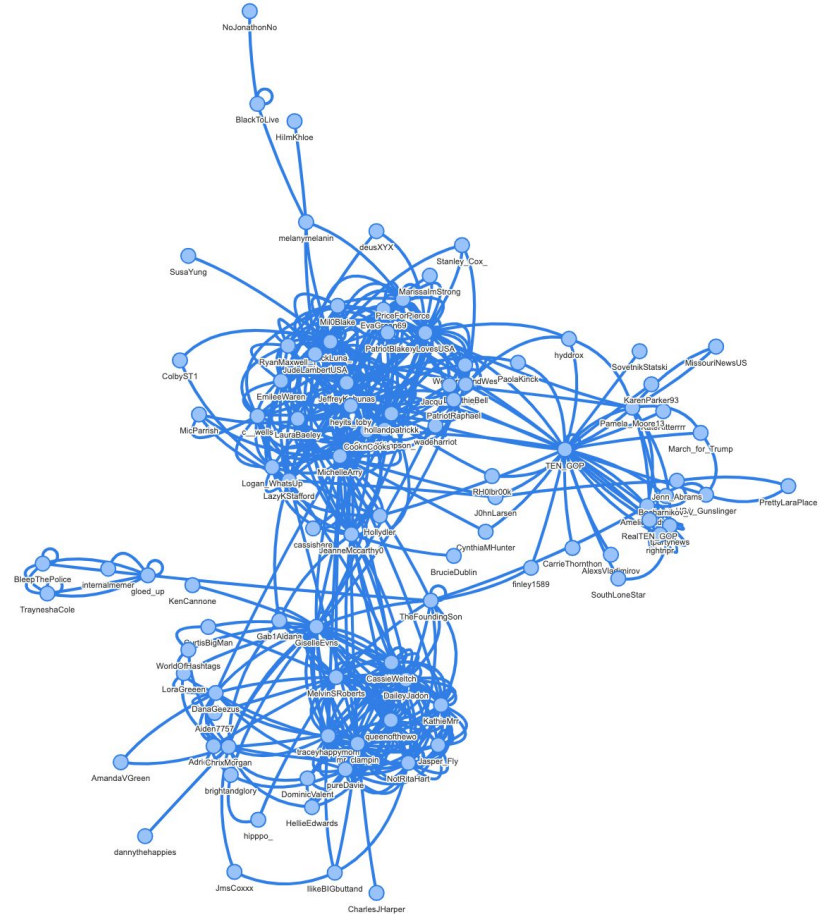
Initial visualization

```
var config = {
  container_id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name"
    }
  },
  relationships: {
    AMPLIFIED: {
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



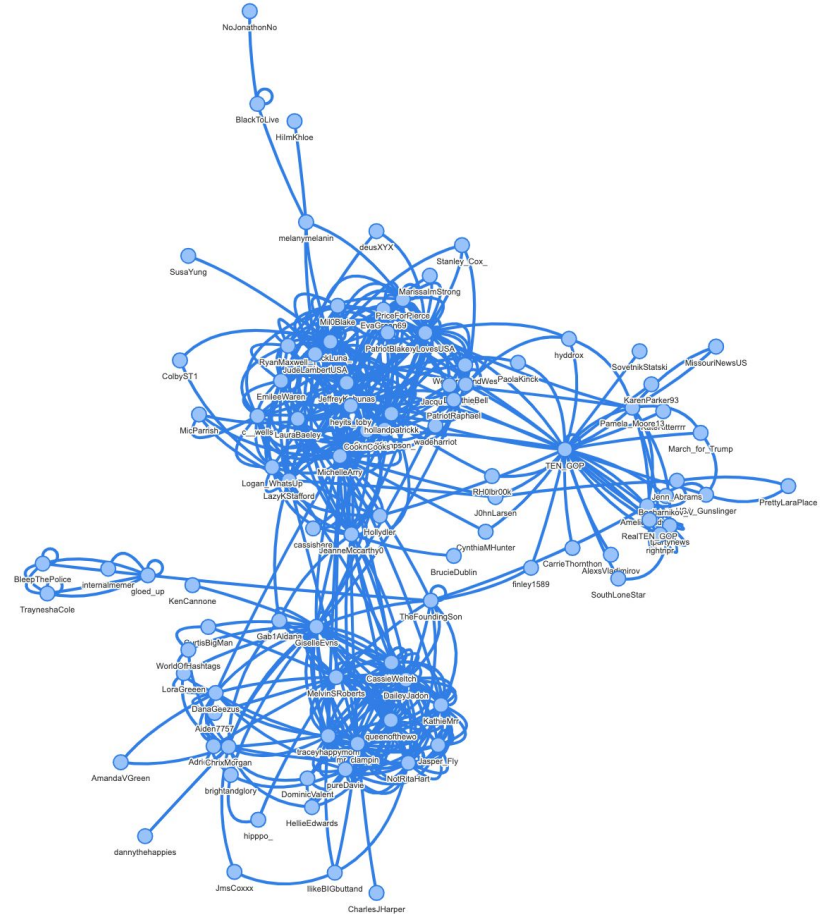
Initial visualization

```
var config = {
  container id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name"
    }
  },
  relationships: {
    AMPLIFIED: {
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



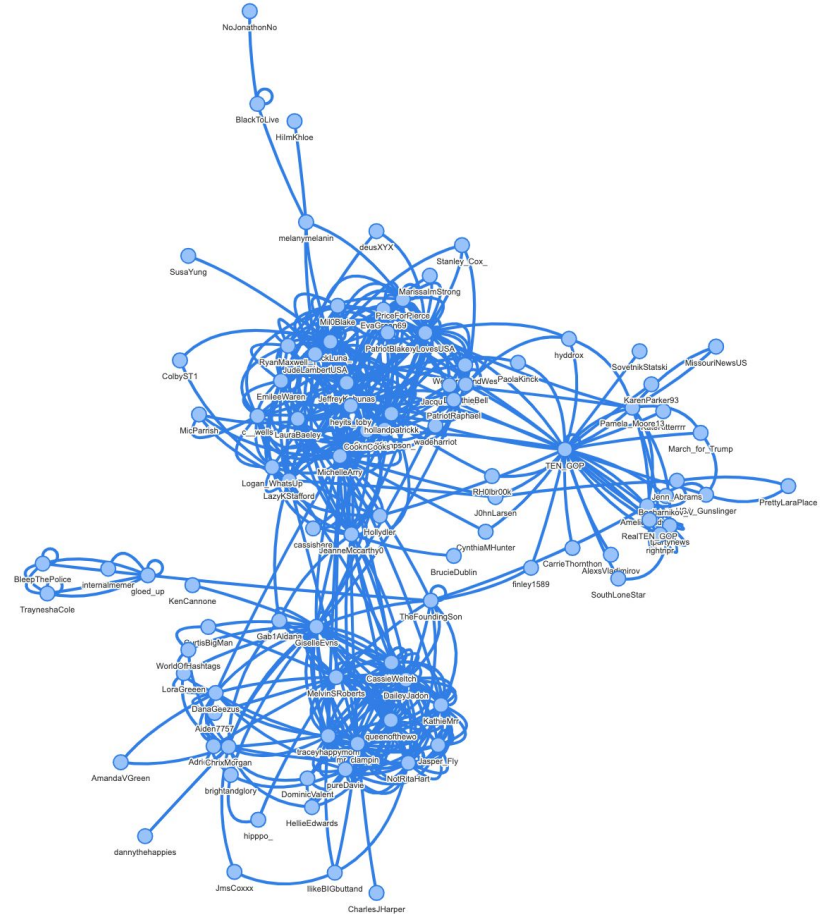
Initial visualization

```
var config = {
  container_id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name"
    }
  },
  relationships: {
    AMPLIFIED: {
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



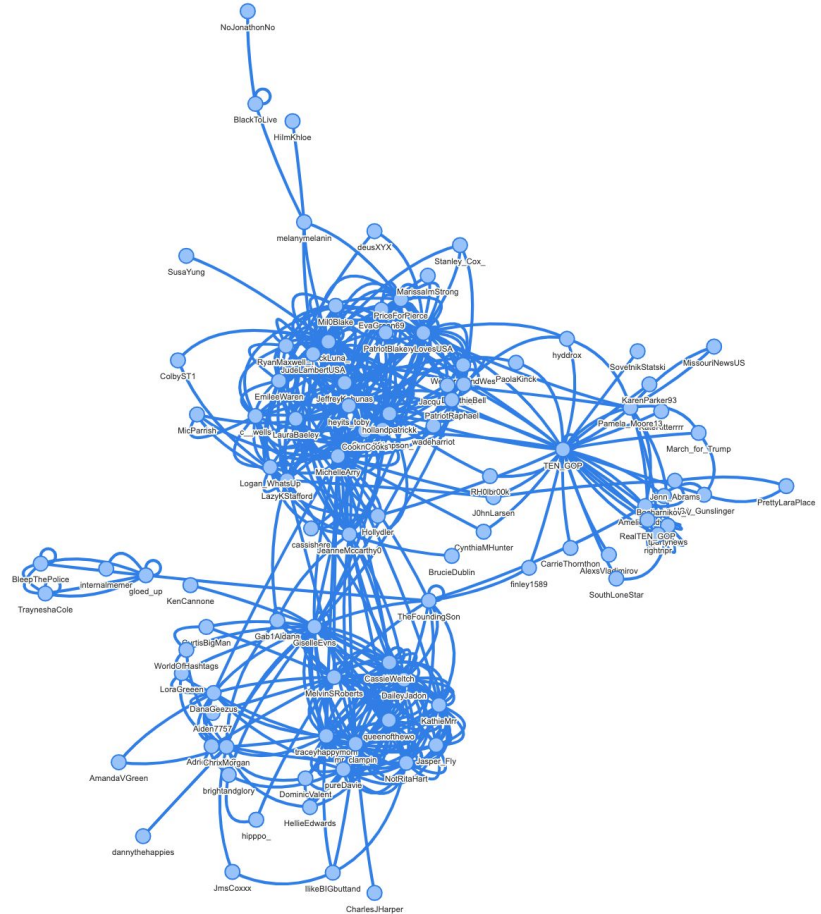
Initial visualization

```
var config = {
  container_id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name"
    }
  },
  relationships: {
    AMPLIFIED: {
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



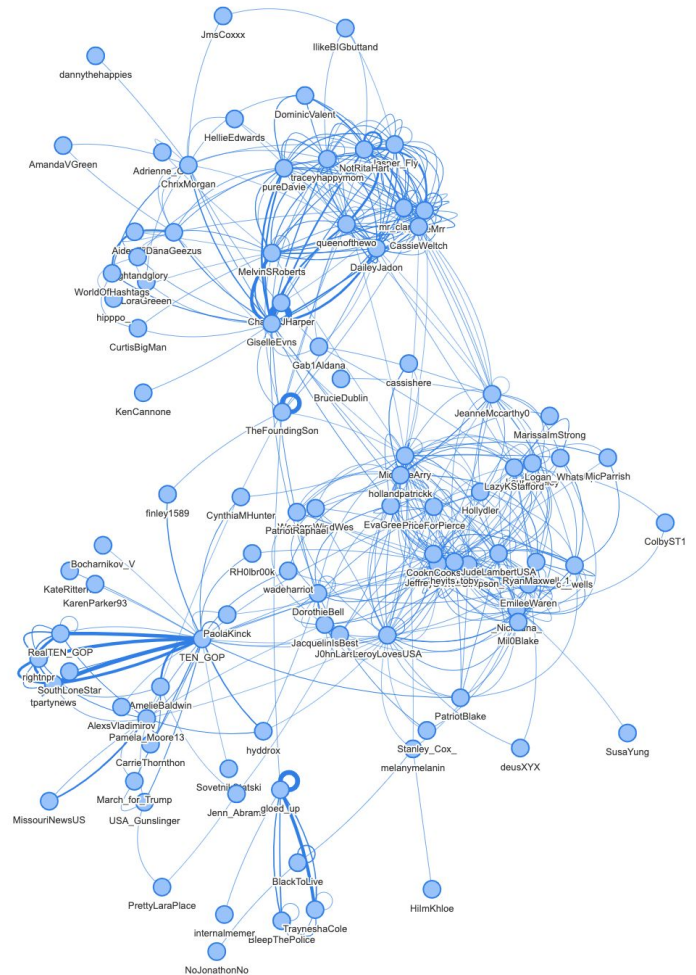
Initial visualization

```
var config = {  
  container_id: "viz",  
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",  
  server_user: "neo4j",  
  server_password: "lapse-narcotics-turnarounds",  
  labels: {  
    Troll: {  
      caption: "screen_name"  
    }  
  },  
  relationships: {  
    AMPLIFIED: {  
      caption: false  
    }  
  },  
  initial_cypher:   
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",  
  encrypted: "ENCRYPTION_ON"  
};
```



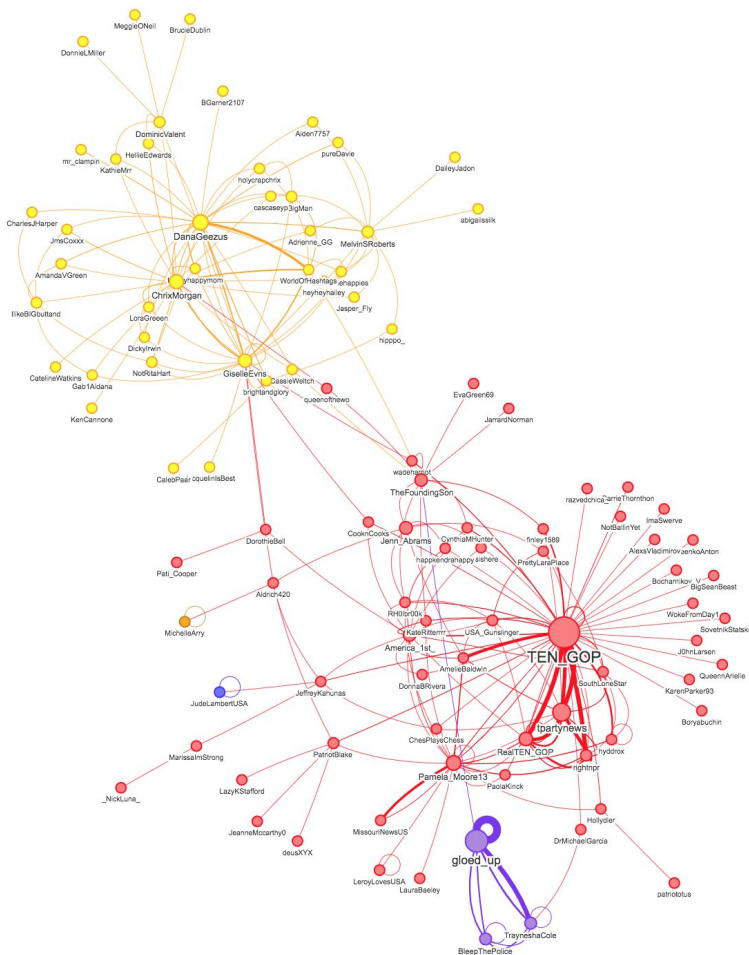
Adding relationship thickness

```
var config = {
  container_id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name"
    }
  },
  relationships: {
    AMPLIFIED: {
      thickness: "count",
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



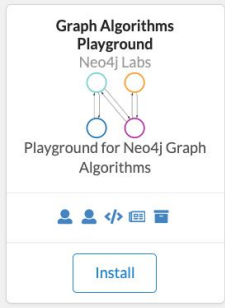
Adding centrality and community detection

```
var config = {
  container_id: "viz",
  server_url: "bolt://ws-10-0-1-230-37028.neo4jsandbox.com:443",
  server_user: "neo4j",
  server_password: "lapse-narcotics-turnarounds",
  labels: {
    Troll: {
      caption: "screen_name",
      size: "pagerank",
      community: "community"
    }
  },
  relationships: {
    AMPLIFIED: {
      thickness: "count",
      caption: false
    }
  },
  initial_cypher:
    "MATCH (n:Troll)-[r:AMPLIFIED]->(m:Troll) RETURN *",
  encrypted: "ENCRYPTION_ON"
};
```



NEuler: Graph Algorithms Playground

install.graphapp.io



Run **Graph Algorithms** with no code in Neo4j Desktop!

Embeds **Neovis.js** for visualizing algorithm results

Louvain
one of the fastest modularity-based algorithms and also reveals a hierarchy of communities at different scales

Label:

Relationship Type:

Direction: Out In Both

Store results:

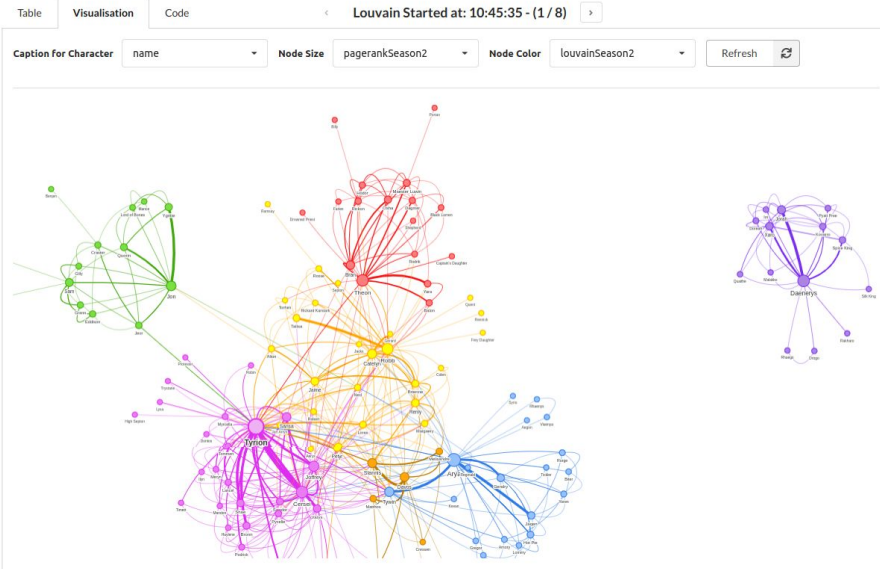
Concurrency:

Weight Property:

Default weight:

Community Property:

Intermediate Communities?



neovis.js

npm install neovis.js

neovis.js

1.2.1 • Public • Published a day ago

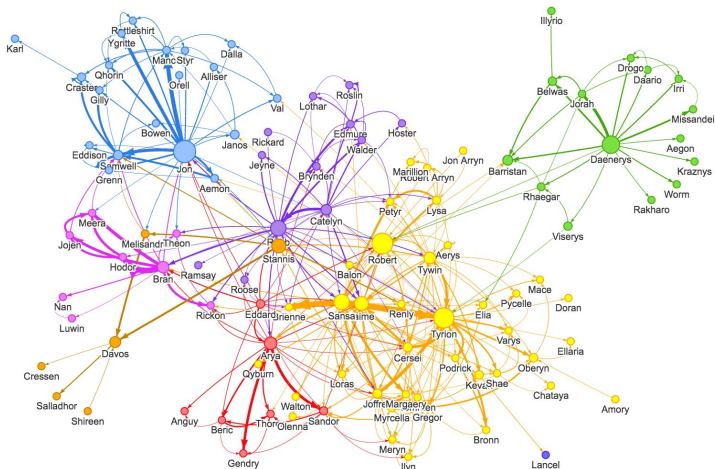
Readme

3 Dependencies

neovis.js

🟢 PASSED npm package 1.2.1

Graph visualizations powered by vis.js with data from Neo4j.



```
var config = {
  container_id: "viz",
  server_url: "bolt://localhost:7687",
  server_user: "neo4j",
  server_password: "sorts-swims-burglaries",
  labels: {
    "Character": {
      "caption": "name",
      "size": "pagerank",
      "community": "community"
    }
  },
  relationships: {
    "INTERACTS": {
      "thickness": "weight",
      "caption": false
    }
  },
  initial_cypher: "MATCH (n)-[r:INTERACTS]->(m) RETURN *";
};

viz = new NeoVis.default(config);
viz.render();
```

<https://www.npmjs.com/package/neovis.js>

Hunger Games Questions for "Leveraging Graph Algorithms In Visualizations With Neovis.js"



1. True or False: results of Graph Algorithms can be used in graph visualizations?
 - a. True
 - b. False
2. What is the command to install neovis.js?
 - a. `sudo apt-get install willviz`
 - b. `npm install neovis.js`
 - c. `wget http://grandstack.io/docs`
3. Which of the following algorithms is used to find communities in a graph?
 - a. PageRank
 - b. Label Propagation
 - c. Eigenvector Centrality

[Answer here: r.neo4j.com/hunger-games](http://r.neo4j.com/hunger-games)