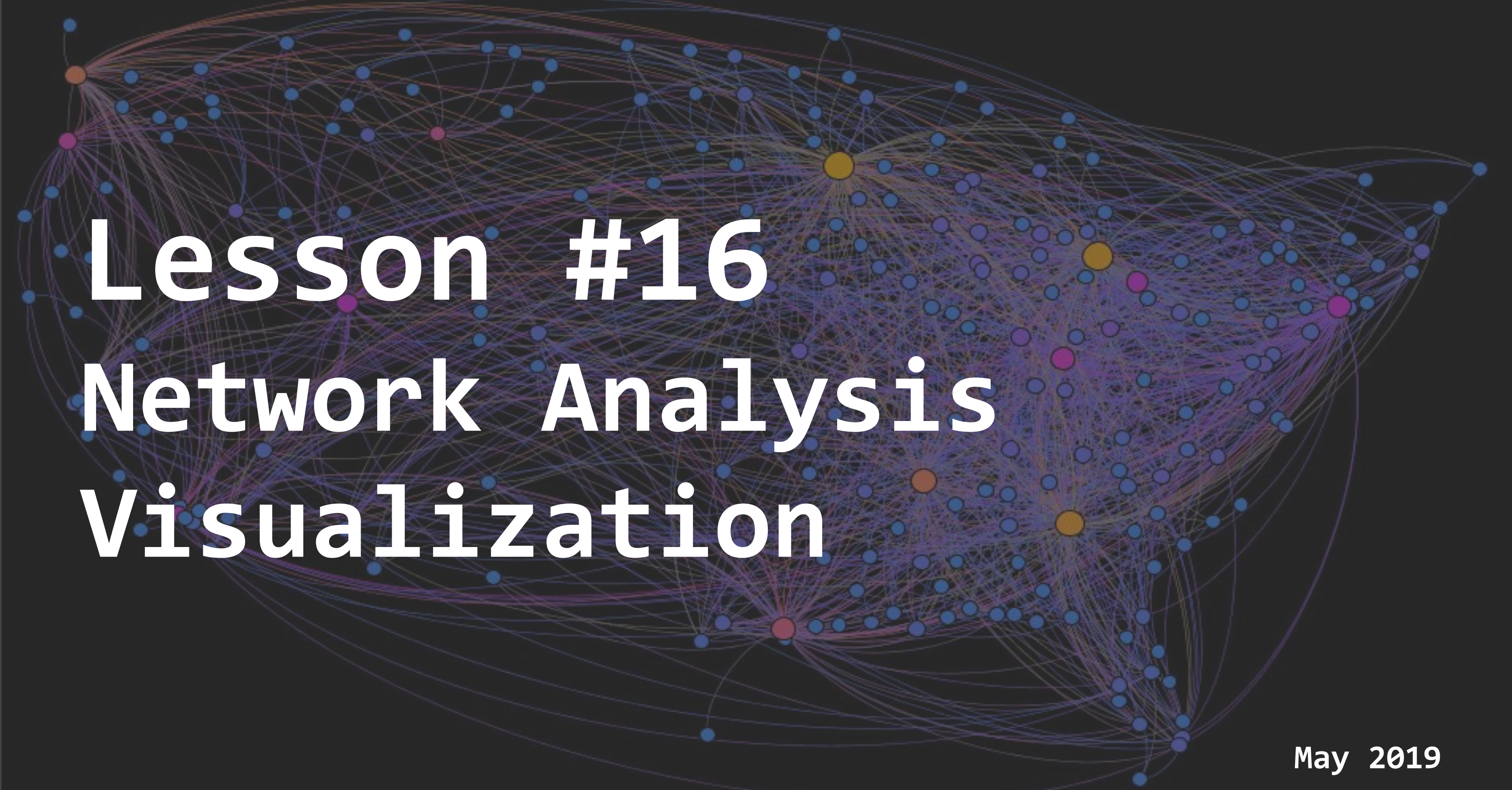
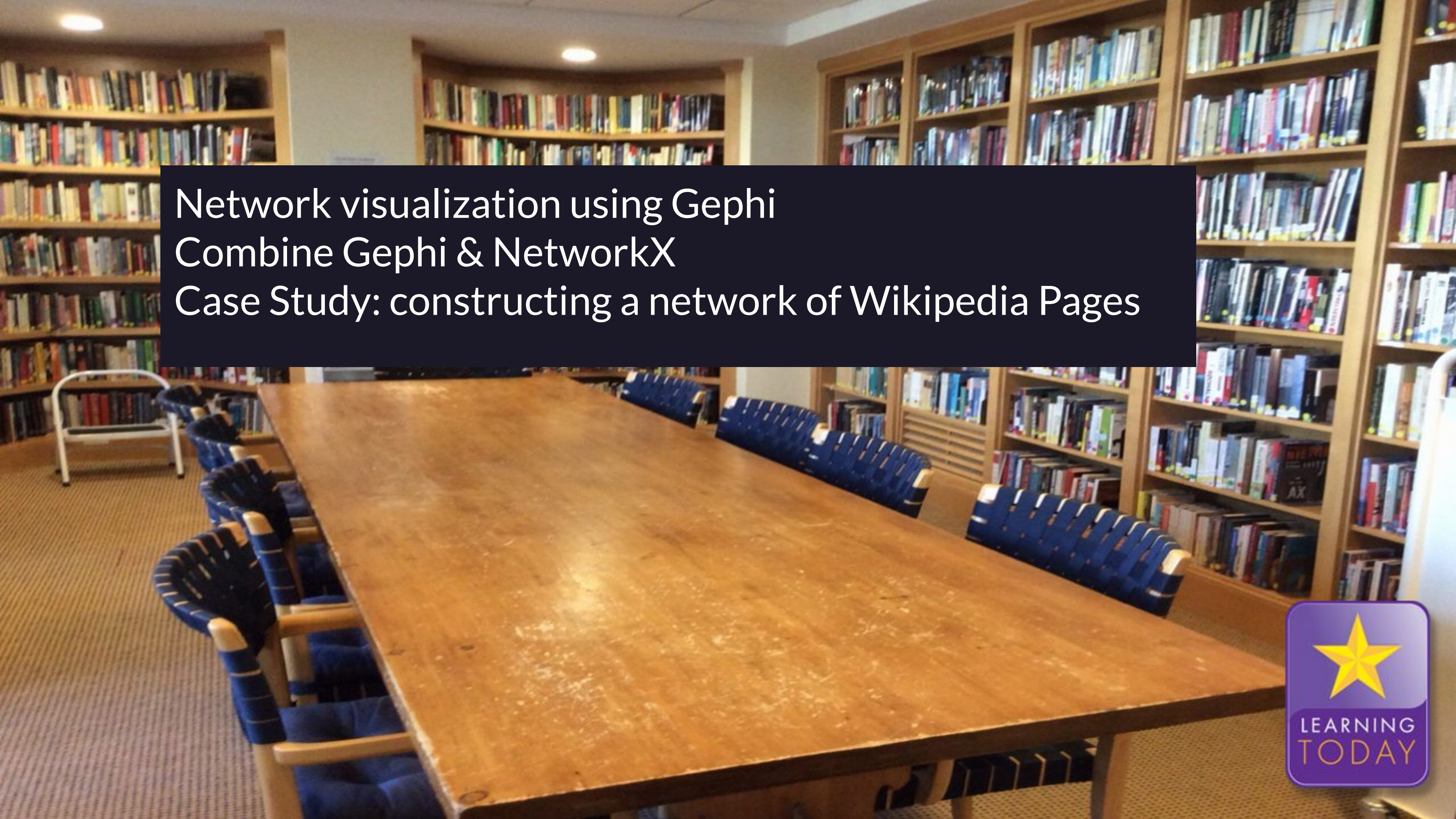


Lesson #16
Network Analysis Visualization

May 2019



Network visualization using Gephi
Combine Gephi & NetworkX
Case Study: constructing a network of Wikipedia Pages

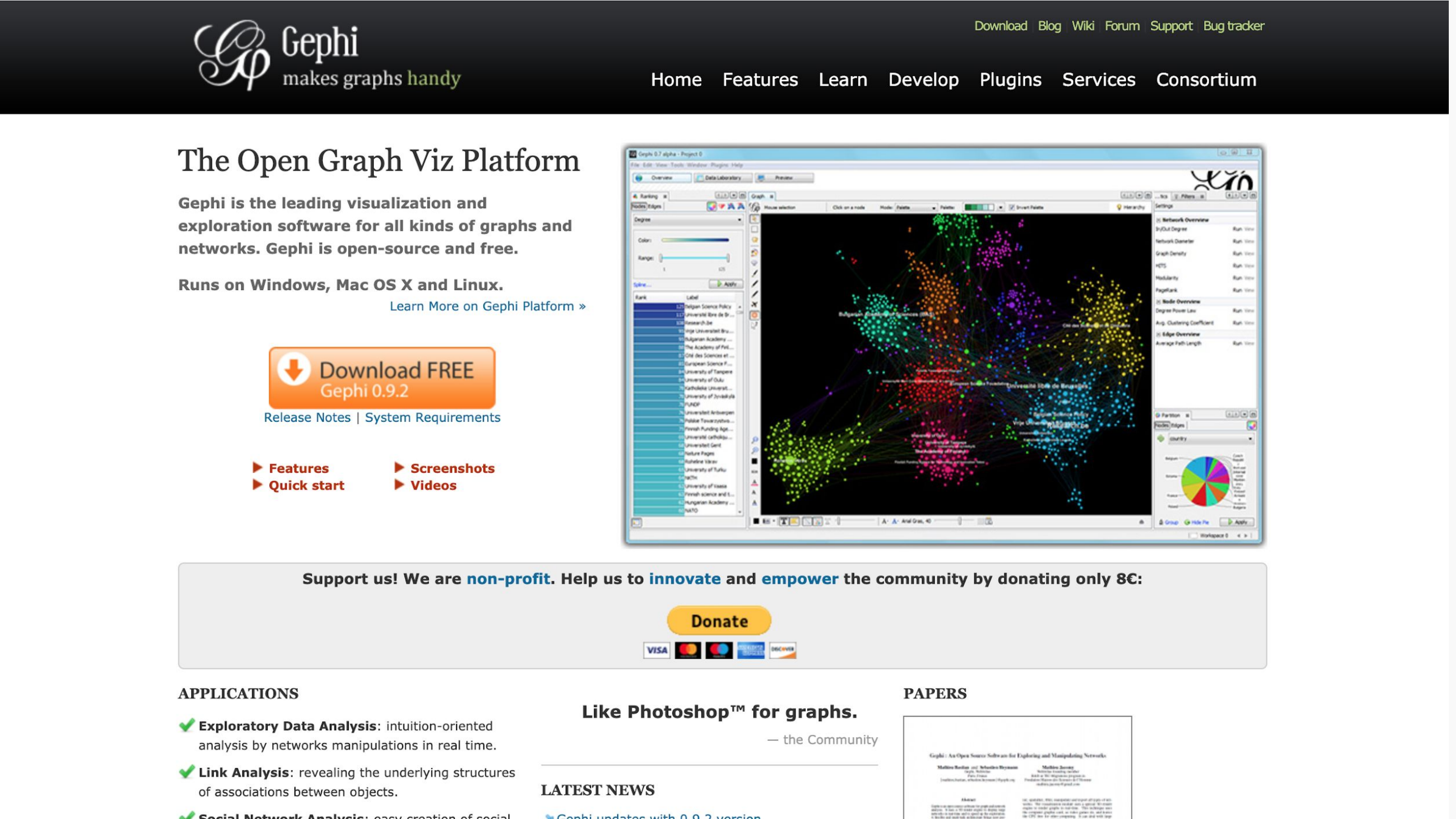
LEARNING TODAY

Update from repository

```
git clone https://github.com/ivanovitchm/datascience_one_2019_1
```

Or ….

```
git pull
```



Download | Blog | Wiki | Forum | Support | Bug tracker

Gephi
makes graphs handy

Home Features Learn Develop Plugins Services Consortium

The Open Graph Viz Platform

Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

Runs on Windows, Mac OS X and Linux.

Learn More on Gephi Platform »

Download FREE
Gephi 0.9.2

Release Notes | System Requirements

- **Features**
- **Quick start**
- **Screenshots**
- **Videos**

Support us! We are non-profit. Help us to innovate and empower the community by donating only 8€:

Donate

APPLICATIONS

- ✓ **Exploratory Data Analysis**: intuition-oriented analysis by networks manipulations in real time.
- ✓ **Link Analysis**: revealing the underlying structures of associations between objects.
- ✓ **Social Network Analysis**: easy creation of social

Like Photoshop™ for graphs.

— the Community

LATEST NEWS

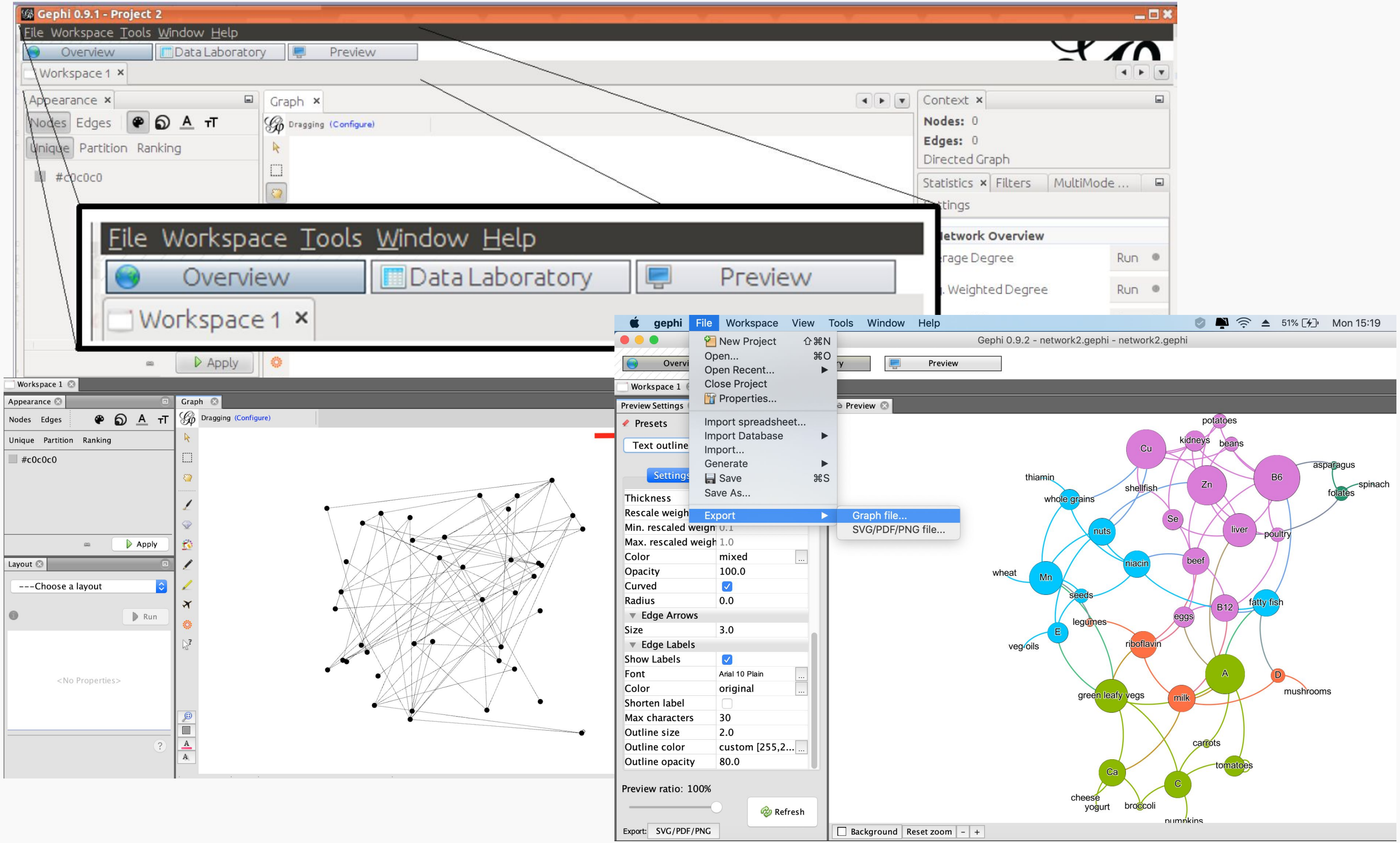
Gephi updates with 0.9.2 version

PAPERS

Gephi : An Open Source Software for Exploring and Manipulating Networks

Mathieu Bastian and Sebastien Heymann
Mathieu Jacomy

Abstract



Nodes: 0
Edges: 0
Directed Graph

Thickness
Rescale weigh
Min. rescaled weigh 0.1
Max. rescaled weigh 1.0
Color mixed
Opacity 100.0
Curved ☑
Radius 0.0
Edge Arrows
Size 3.0
Edge Labels
Show Labels ☑
Font Arial 10 Plain
Color original
Shorten label □
Max characters 30
Outline size 2.0
Outline color custom [255,2...
Outline opacity 80.0

Preview ratio: 100%

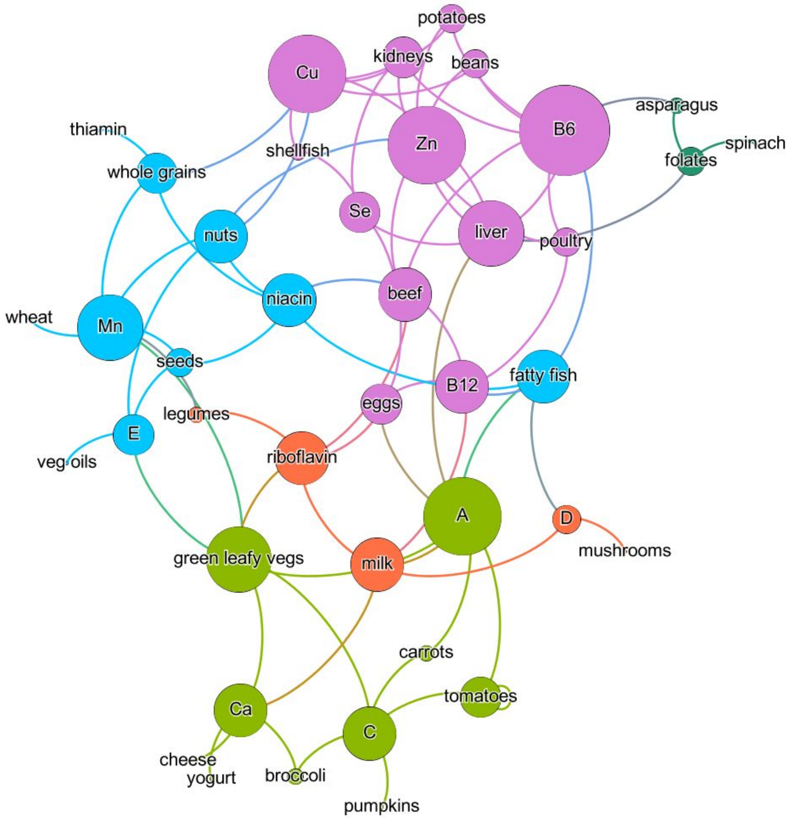
Export: SVG/PDF/PNG

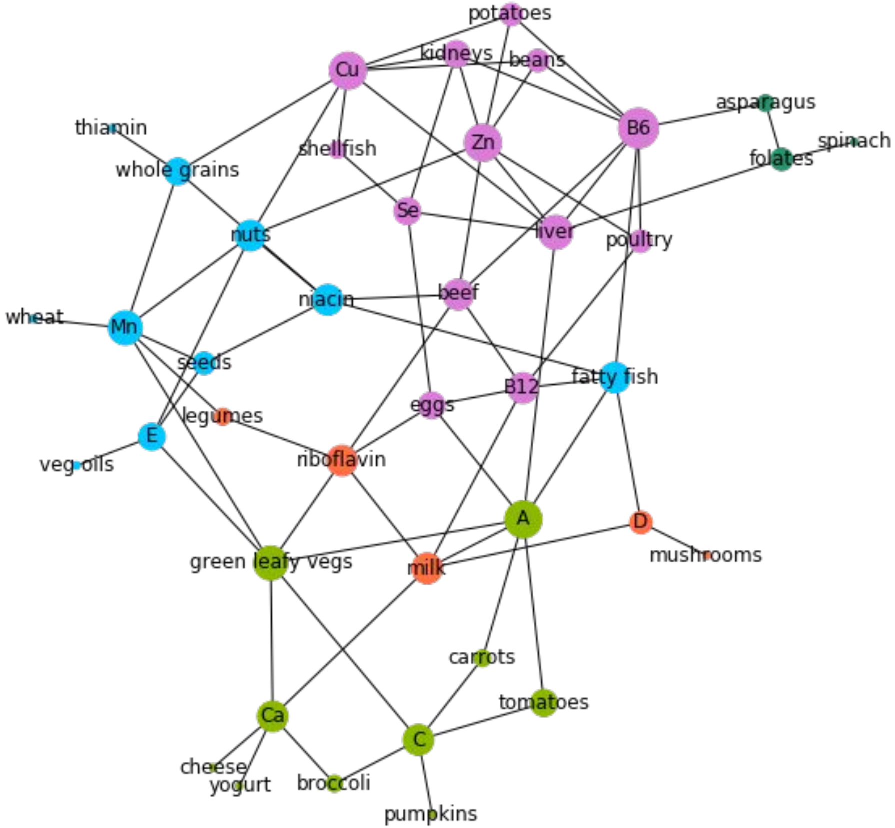
Tutorial Hands on Gephi

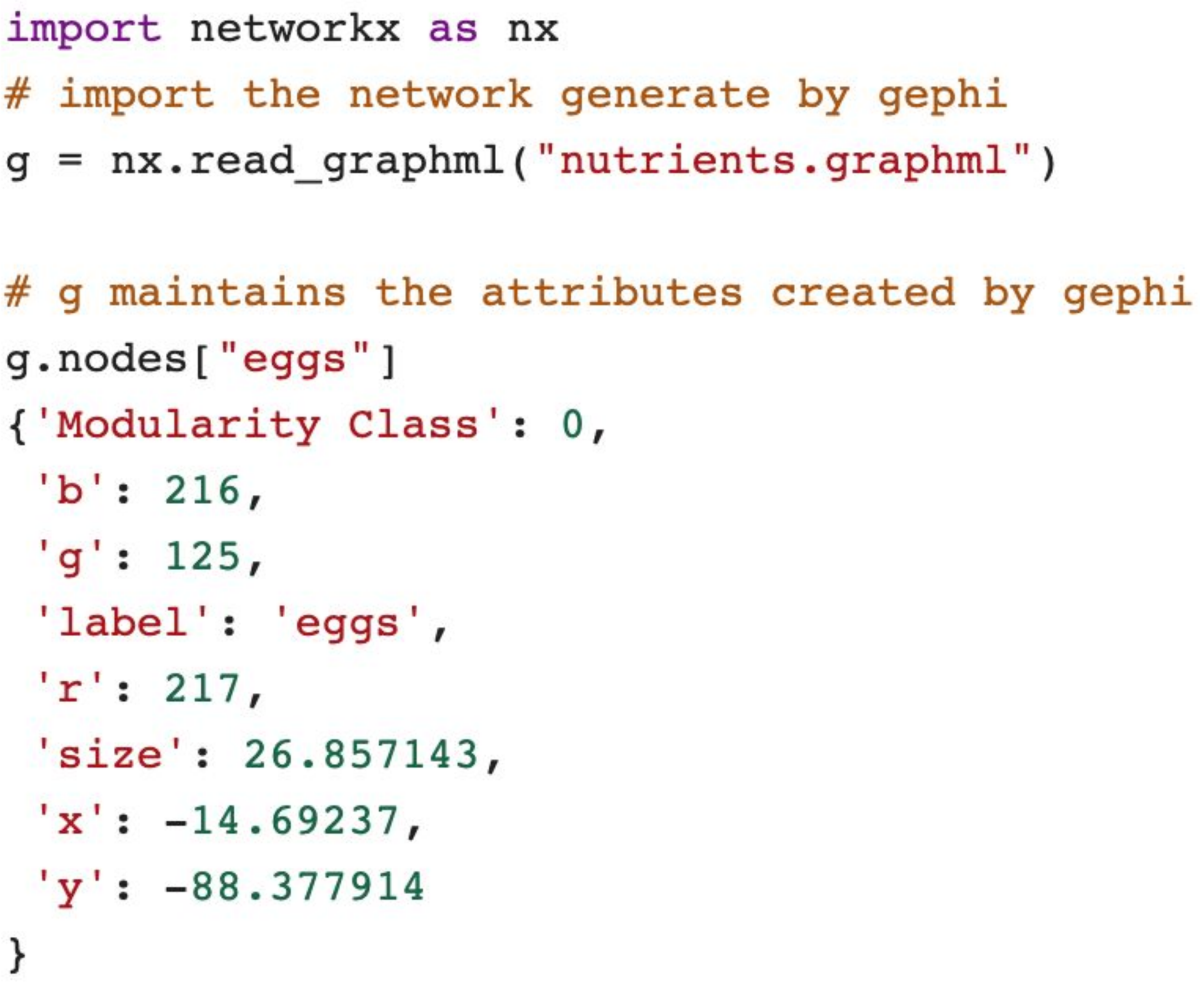
1. Install gephi
2. Import nutrients.csv
3. Modify a simple network
4. Explore the network
5. Sketch the network
6. Prepare a presentation-quality image
7. Export the network

KEEP
CALM
AND
LETS DO IT
TOGETHER

Combine Gephi & NetworkX

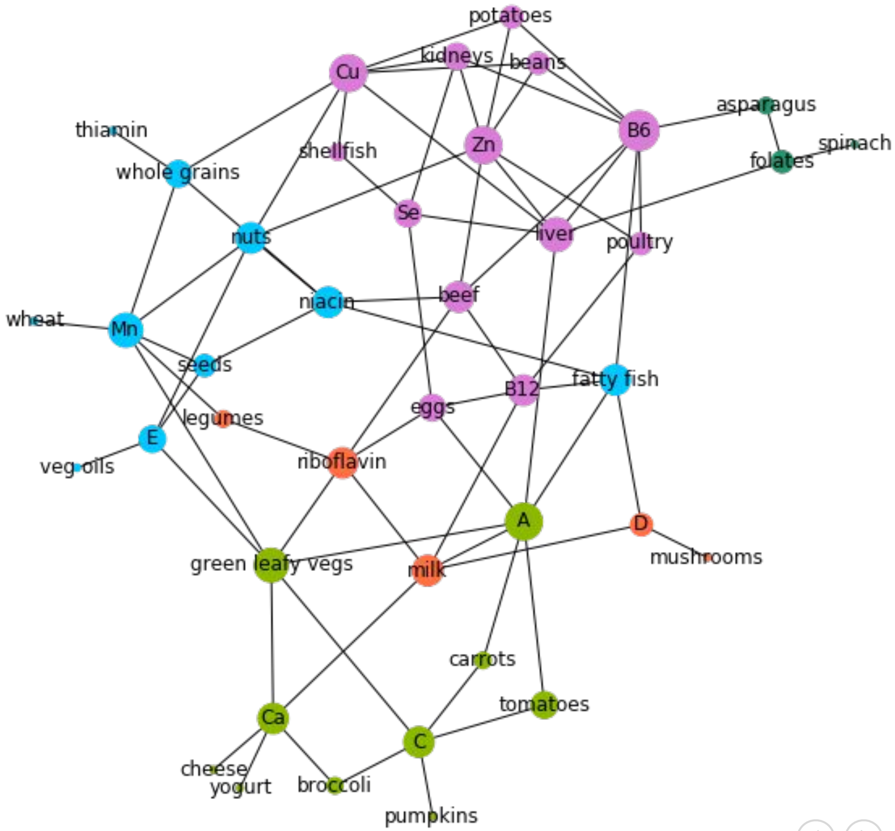






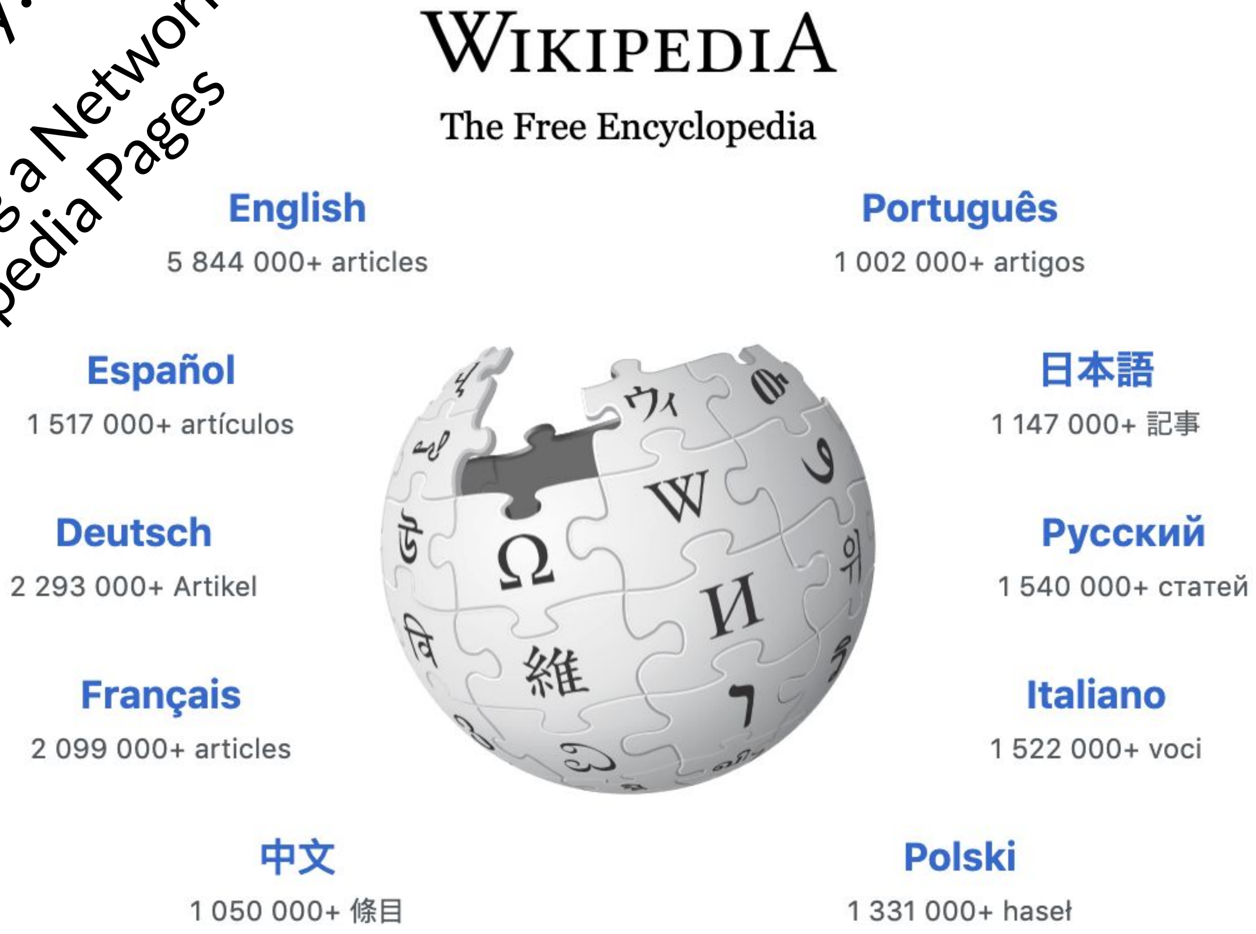
```
import networkx as nx
# import the network generate by gephi
g = nx.read_graphml("nutrients.graphml")

# g maintains the attributes created by gephi
g.nodes["eggs"]
{'Modularity Class': 0,
 'b': 216,
 'g': 125,
 'label': 'eggs',
 'r': 217,
 'size': 26.857143,
 'x': -14.69237,
 'y': -88.377914
}
```

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(10,10))
nx.draw_networkx(g, pos=pos,
                 labels=labels,
                 node_size=node_size,
                 ax=ax,
                 node_color=node_color)


plt.axis("off")
plt.show()
```



Let's
CODE It.

Case Study: Constructing a Network of Wikipedia Pages



Wikipedia

The Free Encyclopedia

English
5 844 000+ articles

Português
1 002 000+ artigos

Español
1 517 000+ artículos

日本語
1 147 000+ 記事

Deutsch
2 293 000+ Artikel

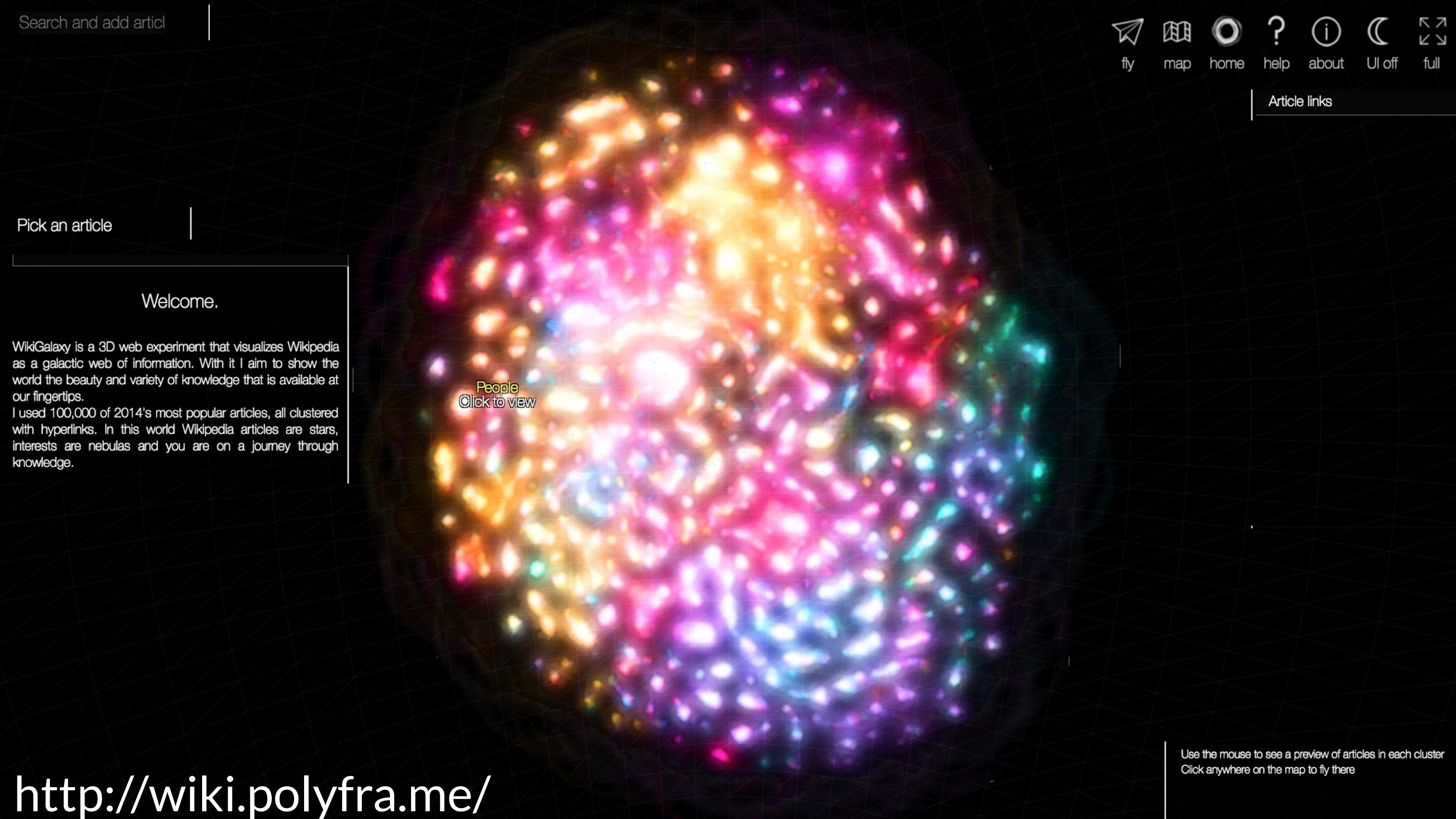
Русский
1 540 000+ статей

Français
2 099 000+ articles

Italiano
1 522 000+ voci

中文
1 050 000+ 條目

Polski
1 331 000+ haseł



Welcome.

WikiGalaxy is a 3D web experiment that visualizes Wikipedia as a galactic web of information. With it I aim to show the world the beauty and variety of knowledge that is available at our fingertips.

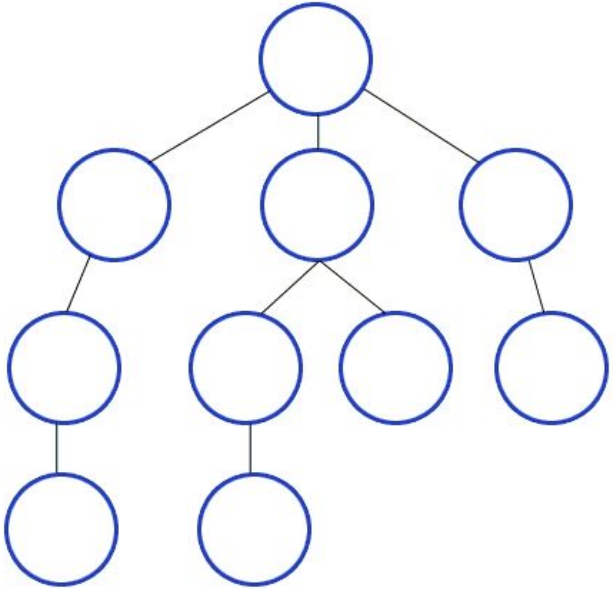
I used 100,000 of 2014's most popular articles, all clustered with hyperlinks. In this world Wikipedia articles are stars, interests are nebulas and you are on a journey through knowledge.

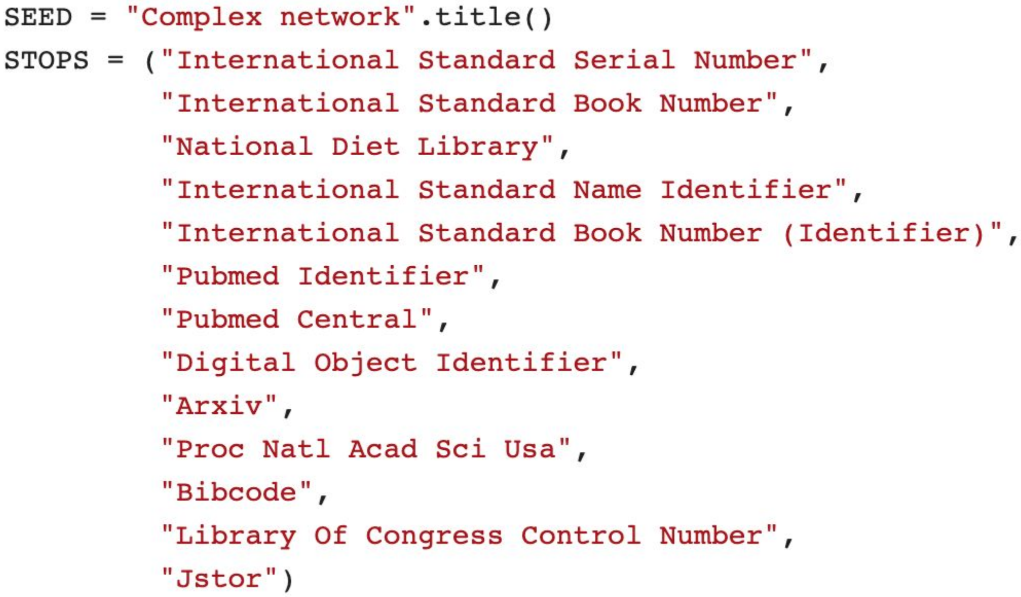
Use the mouse to see a preview of articles in each cluster
Click anywhere on the map to fly there

http://wiki.polyfra.me/

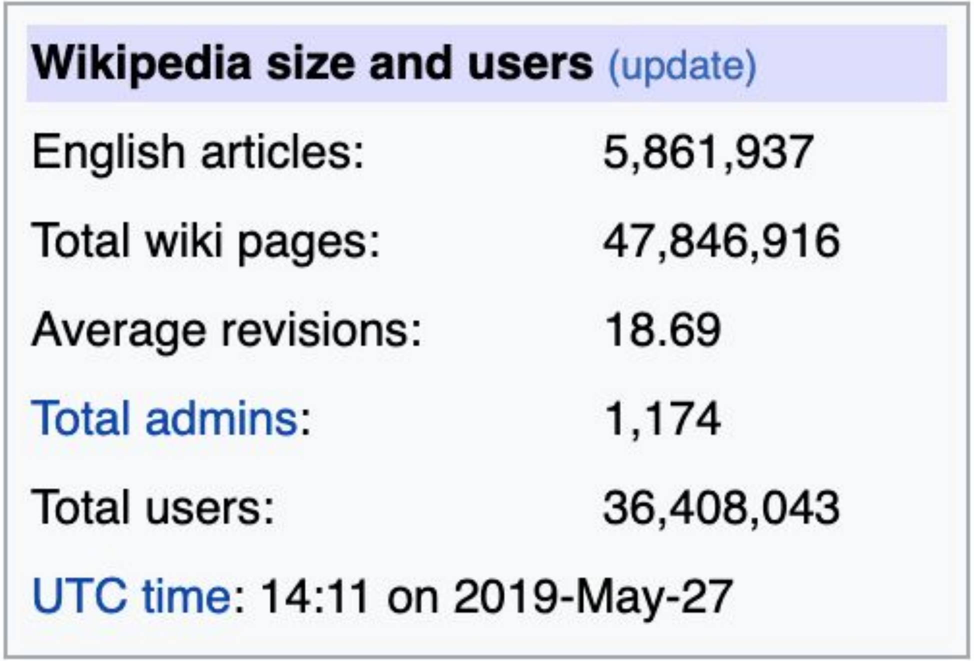
Get the data, build the network

Snowballing process (breadth-first search - BFS)





```
SEED = "Complex network".title()
STOPS = ("International Standard Serial Number",
         "International Standard Book Number",
         "National Diet Library",
         "International Standard Name Identifier",
         "International Standard Book Number (Identifier)",
         "Pubmed Identifier",
         "Pubmed Central",
         "Digital Object Identifier",
         "Arxiv",
         "Proc Natl Acad Sci Usa",
         "Bibcode",
         "Library Of Congress Control Number",
         "Jstor")
```

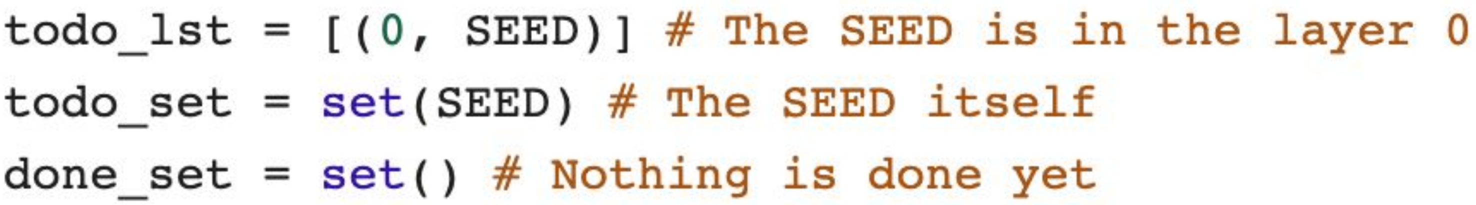


Wikipedia size and users (update)

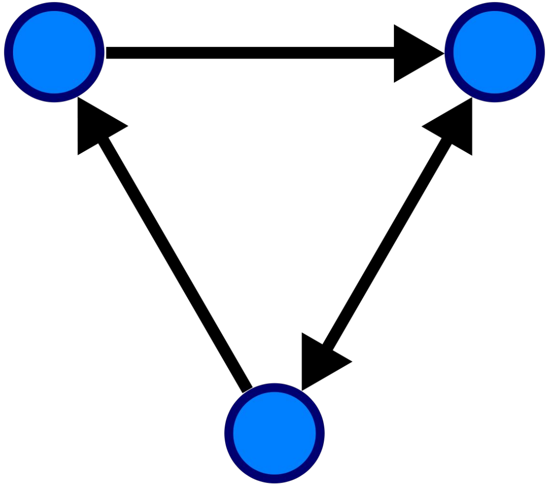
English articles:	5,861,937
Total wiki pages:	47,846,916
Average revisions:	18.69
Total admins:	1,174
Total users:	36,408,043

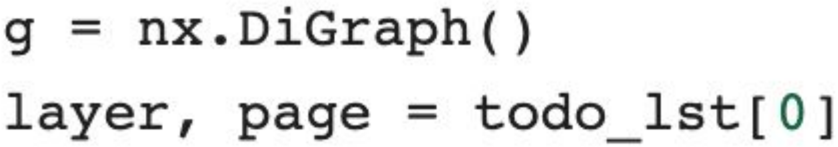
UTC time: 14:11 on 2019-May-27

Layer 0: 1
Layer 1: N
Layer 2: N + N*N
....



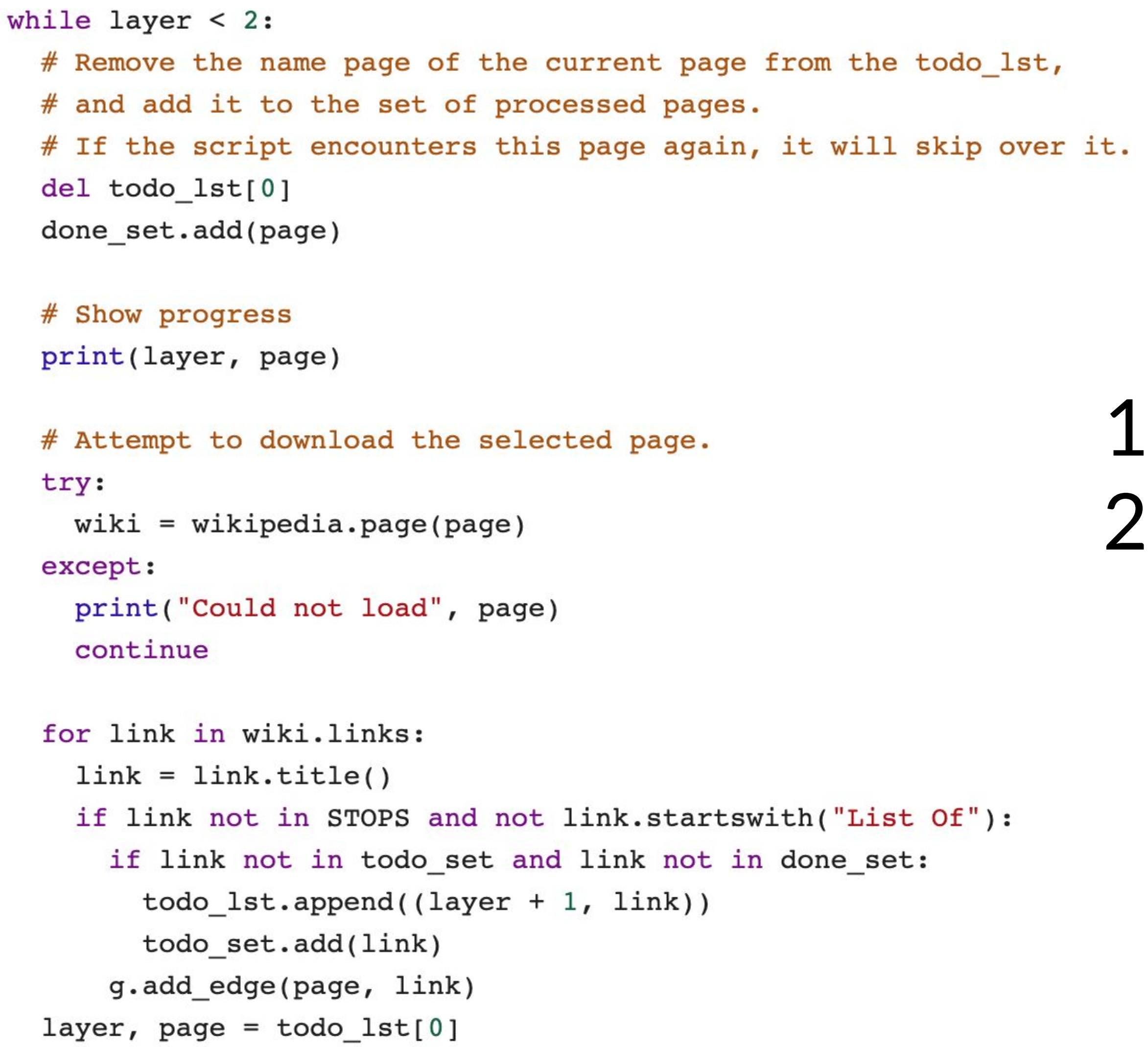
```
todo_lst = [(0, SEED)] # The SEED is in the layer 0
todo_set = set(SEED) # The SEED itself
done_set = set() # Nothing is done yet
```





```
g = nx.DiGraph()
layer, page = todo_lst[0]
```

We choose a directed graph because the edges that represent HTML links are naturally directed: a link from page A to page B does not imply a reciprocal link.

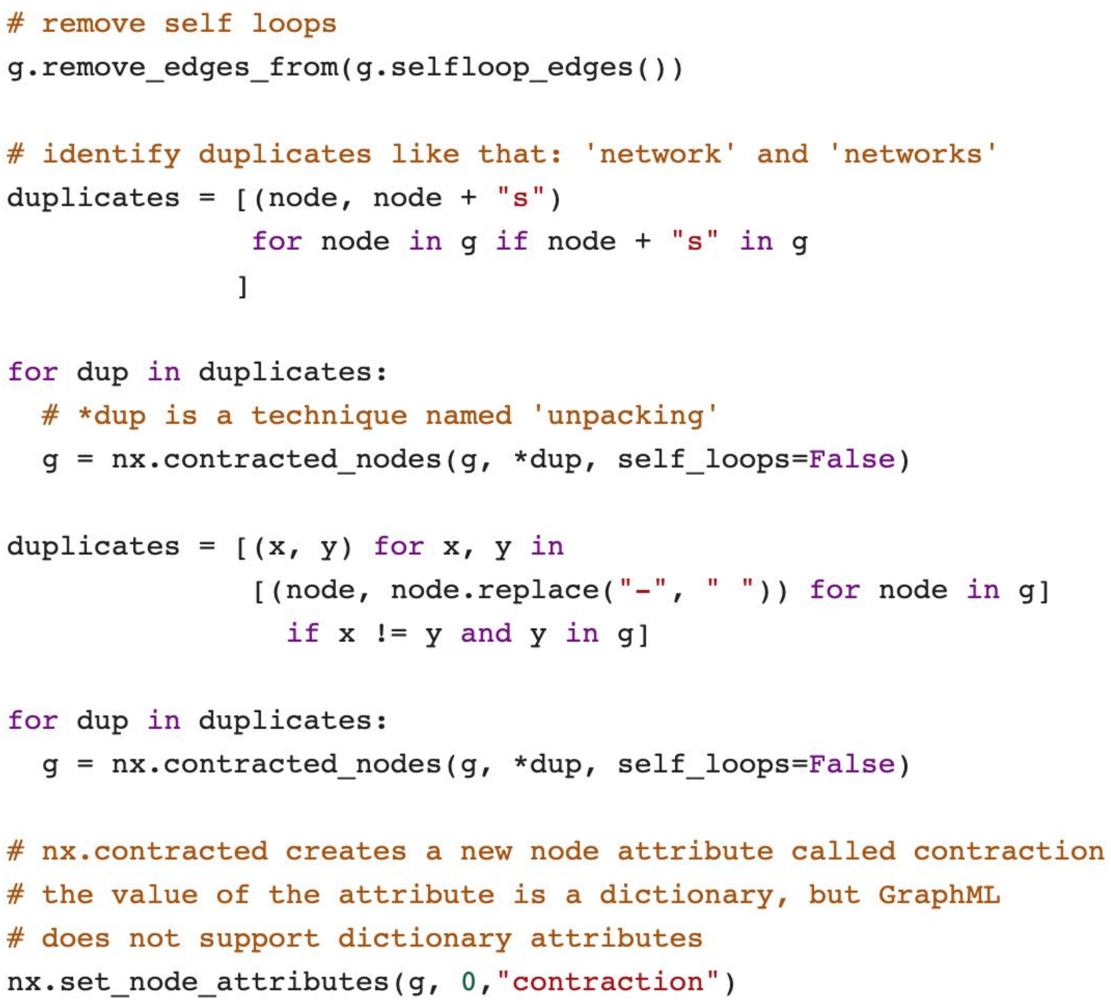


```python
while layer < 2:
  # Remove the name page of the current page from the todo_lst,
  # and add it to the set of processed pages.
  # If the script encounters this page again, it will skip over it.
  del todo_lst[0]
  done_set.add(page)

  # Show progress
  print(layer, page)

  # Attempt to download the selected page.
  try:
    wiki = wikipedia.page(page)
  except:
    print("Could not load", page)
    continue

  for link in wiki.links:
    link = link.title()
    if link not in STOPS and not link.startswith("List Of"):
      if link not in todo_set and link not in done_set:
        todo_lst.append((layer + 1, link))
        todo_set.add(link)
      g.add_edge(page, link)
  layer, page = todo_lst[0]
```

11738 nodes

22716 edges

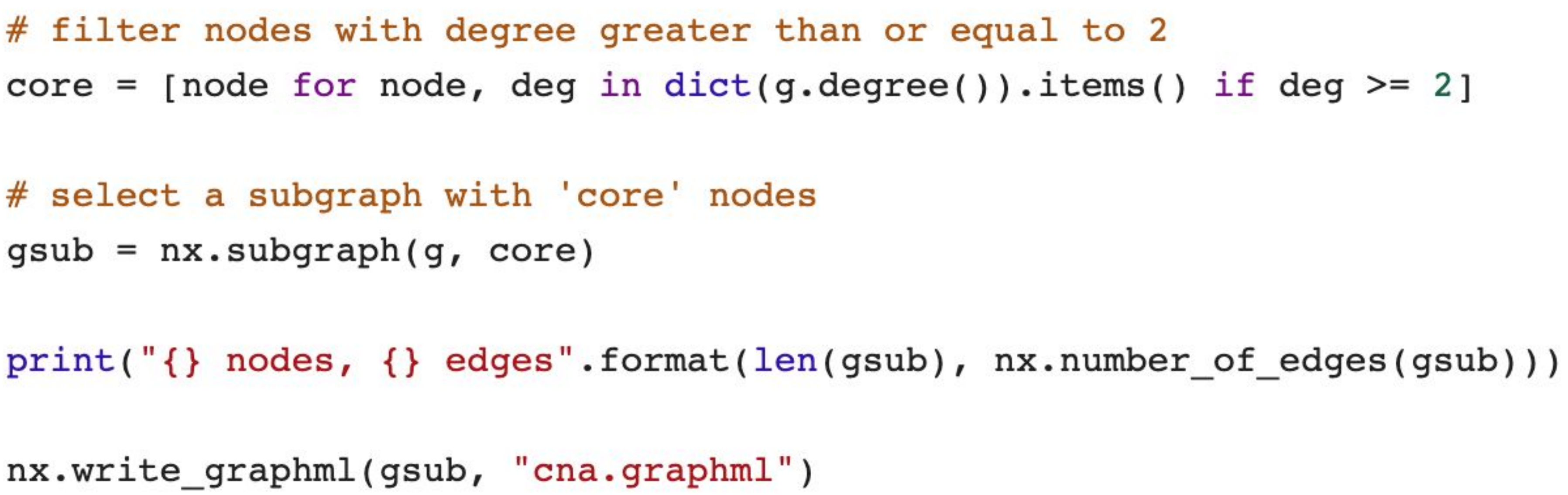
Eliminate Duplicates



```
# remove self loops
g.remove_edges_from(g.selfloop_edges())

# identify duplicates like that: 'network' and 'networks'
duplicates = [(node, node + "s")
              for node in g if node + "s" in g
             ]

for dup in duplicates:
  # *dup is a technique named 'unpacking'
  g = nx.contracted_nodes(g, *dup, self_loops=False)

duplicates = [(x, y) for x, y in
              [(node, node.replace("-", " ")) for node in g]
                if x != y and y in g]

for dup in duplicates:
  g = nx.contracted_nodes(g, *dup, self_loops=False)

# nx.contracted creates a new node attribute called contraction
# the value of the attribute is a dictionary, but GraphML
# does not support dictionary attributes
nx.set_node_attributes(g, 0,"contraction")
```

Before:
11738 nodes
22716 edges

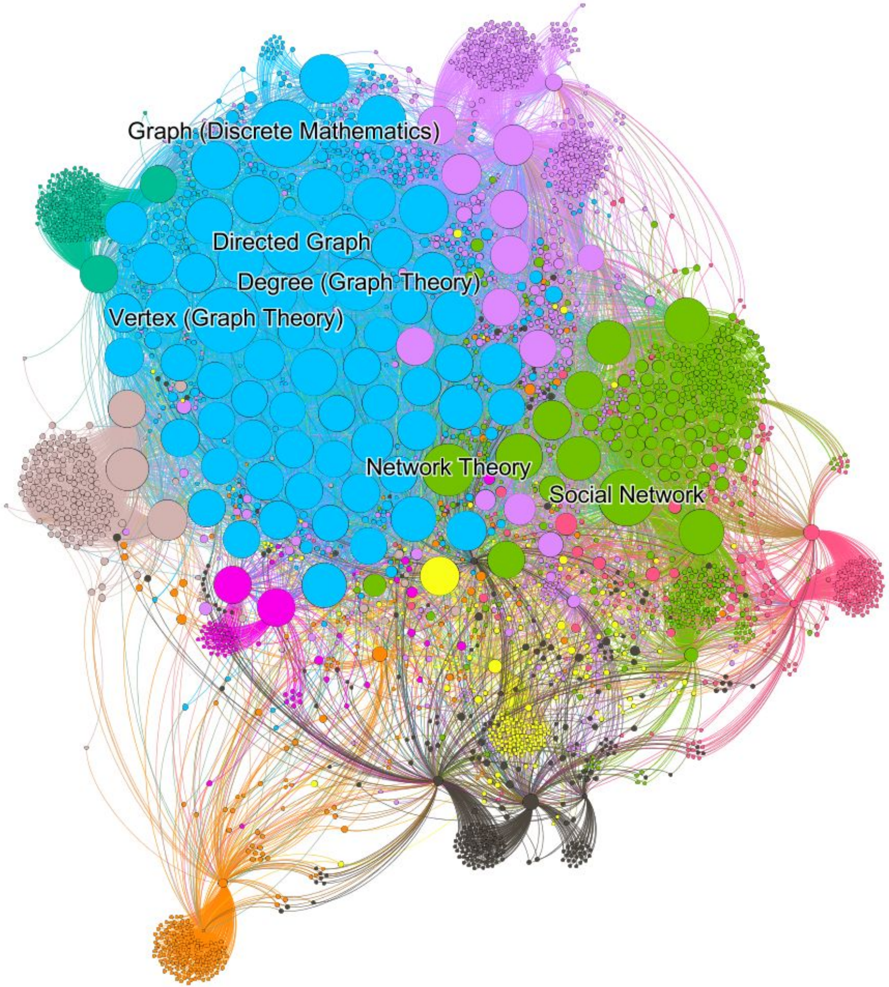
After:
11613 nodes
21574 edges



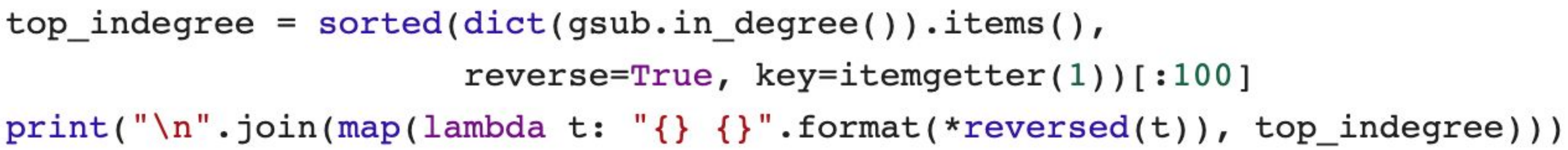
```
# filter nodes with degree greater than or equal to 2
core = [node for node, deg in dict(g.degree()).items() if deg >= 2]

# select a subgraph with 'core' nodes
gsub = nx.subgraph(g, core)

print("{} nodes, {} edges".format(len(gsub), nx.number_of_edges(gsub)))

nx.write_graphml(gsub, "cna.graphml")
```

Truncate the Network (4.11 edges per node)

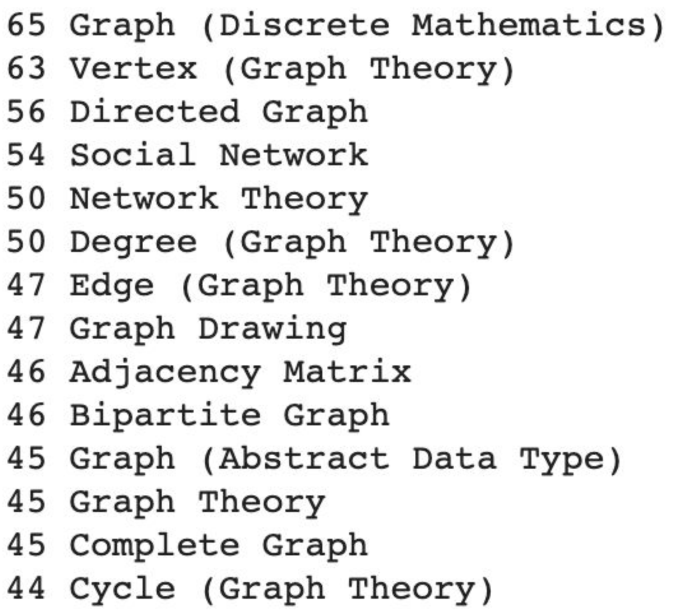
Before:	After:
11613 nodes	3199 nodes
21574 edges	13160 edges



Explore the Network in Gephi



```python
top_indegree = sorted(dict(gsub.in_degree()).items(),
                      reverse=True, key=itemgetter(1))[:100]
print("\n".join(map(lambda t: "{} {}".format(*reversed(t)), top_indegree)))
```



```
65 Graph (Discrete Mathematics)
63 Vertex (Graph Theory)
56 Directed Graph
54 Social Network
50 Network Theory
50 Degree (Graph Theory)
47 Edge (Graph Theory)
47 Graph Drawing
46 Adjacency Matrix
46 Bipartite Graph
45 Graph (Abstract Data Type)
45 Graph Theory
45 Complete Graph
44 Cycle (Graph Theory)
```

Now is your turn!!!