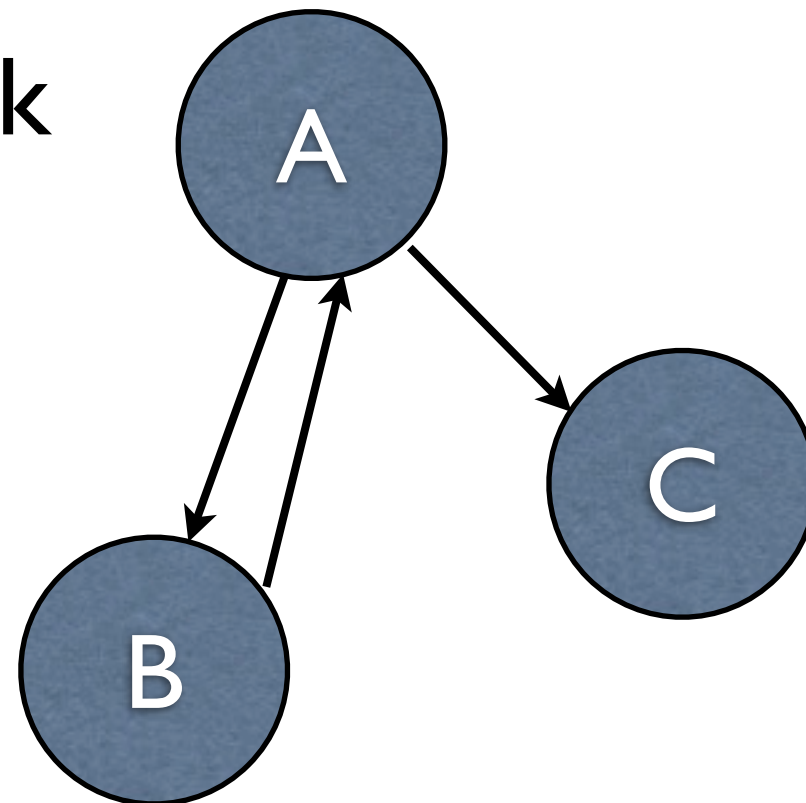


Twitter followers network with Python

S. Stiegemeyer
@Elmos_Buddy

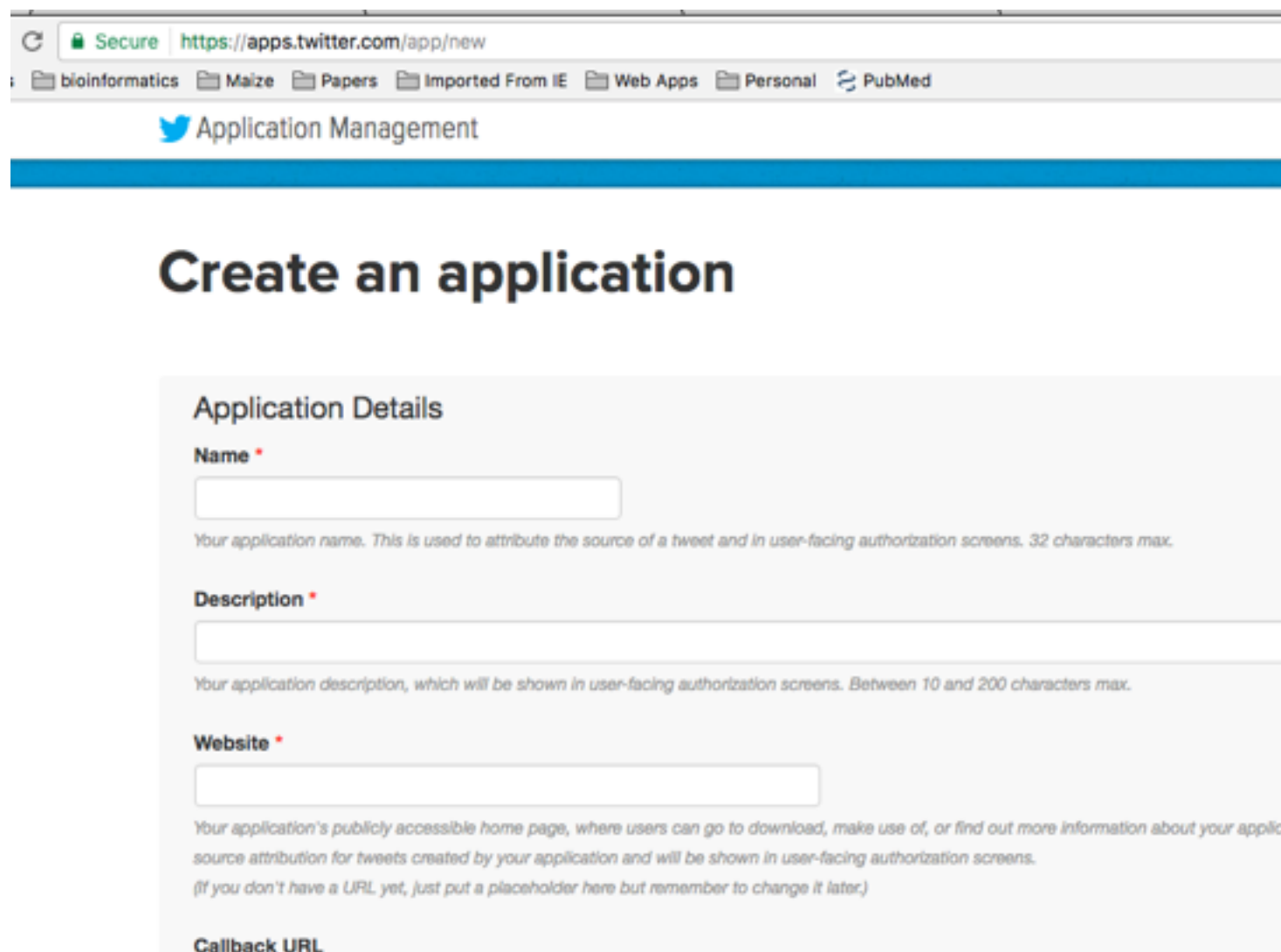
Making a network of people I'm connected to

- I'm going to download my twitter connections
- and make a picture of my connections
- as a directed network



Step 1: Create a Twitter App

- dev.twitter.com
- apps.twitter.com

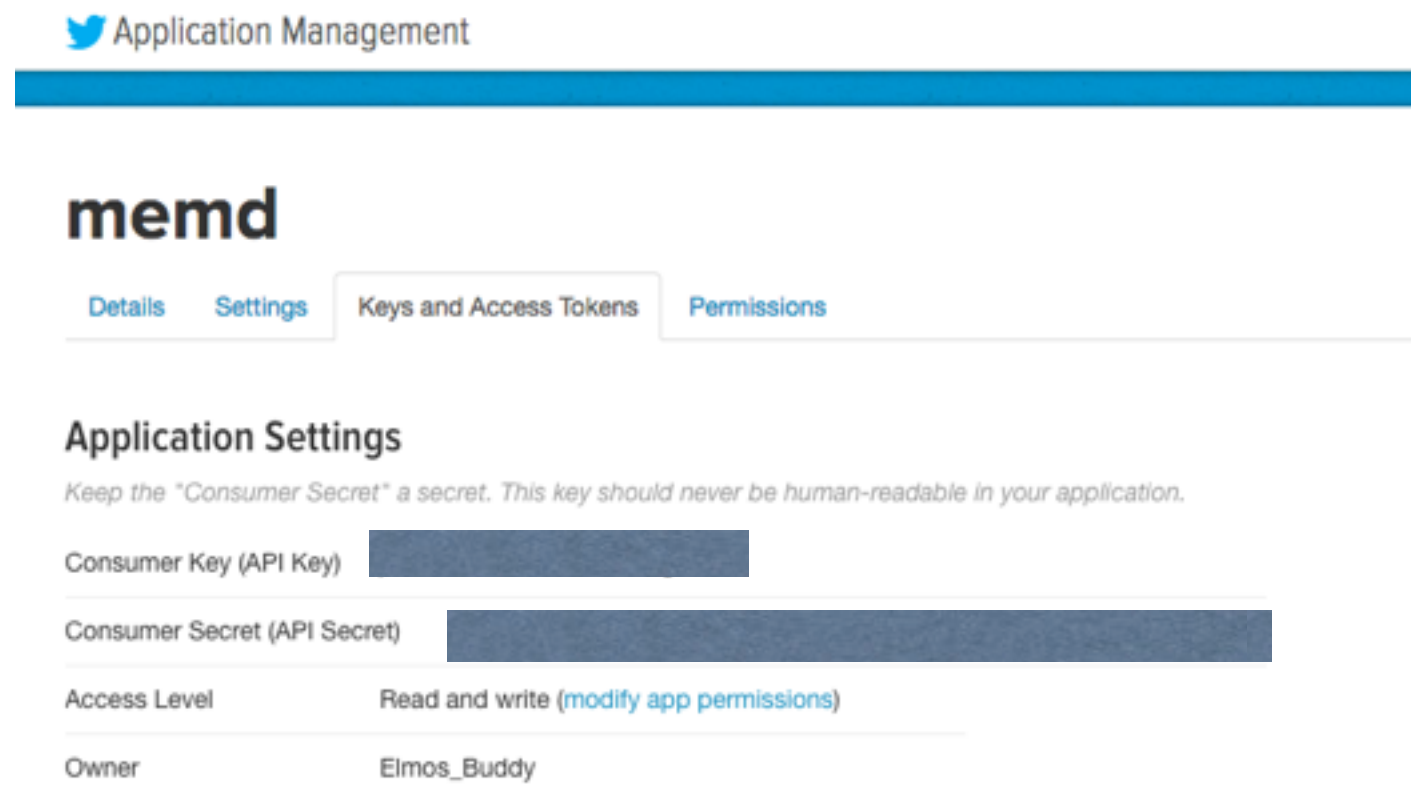


The screenshot shows a web browser window with the address bar displaying <https://apps.twitter.com/app/new>. The browser's address bar also shows a 'Secure' icon and a lock icon. Below the address bar, there is a navigation bar with the Twitter logo and the text 'Application Management'. The main content area is titled 'Create an application' and contains a form with the following fields:

- Application Details**
 - Name ***: A text input field. Below it, a note reads: 'Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.'
 - Description ***: A text input field. Below it, a note reads: 'Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.'
 - Website ***: A text input field. Below it, a note reads: 'Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)'
 - Callback URL**: A text input field.

Step 2: Get keys

- See https://python-twitter.readthedocs.io/en/latest/getting_started.html
- i copied my keys to a file in json format just because :)



The screenshot shows the Twitter Application Management interface. At the top, there's a blue header with the Twitter logo and the text "Application Management". Below this, the application name "memd" is displayed. There are four tabs: "Details", "Settings", "Keys and Access Tokens", and "Permissions". The "Settings" tab is currently selected. Under the "Settings" tab, there's a section titled "Application Settings" with a warning: "Keep the 'Consumer Secret' a secret. This key should never be human-readable in your application." Below this, there are two input fields: "Consumer Key (API Key)" and "Consumer Secret (API Secret)". Both fields are filled with a dark blue color, indicating they have been redacted. Below these fields, there are two rows of information: "Access Level" with the value "Read and write (modify app permissions)" and "Owner" with the value "Elmos_Buddy".

Application Settings	
<i>Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.</i>	
Consumer Key (API Key)	[Redacted]
Consumer Secret (API Secret)	[Redacted]
Access Level	Read and write (modify app permissions)
Owner	Elmos_Buddy

Step 3: Install twitter and networkx

- pip install twitter
 - <https://github.com/bear/python-twitter>
 - <https://python-twitter.readthedocs.io/en/latest/>
- pip install networkx
 - <http://networkx.github.io/>

Step 4: Write some Python

- import libraries and connect to the API

```
import twitter
import networkx
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot
import json

with open("keys.json", 'r') as f:
    mykeys=json.load(f)

api = twitter.Api(consumer_key=mykeys['consumer_key'],
    consumer_secret=mykeys['consumer_secret'],
    access_token_key=mykeys['access_token_key'],
    access_token_secret=mykeys['access_token_secret'])
```

....more python

- Retrieve connections

```
# get who I'm following
friends = api.GetFriends()
fto = [item.screen_name for item in friends]

# get who is following me
followers = api.GetFollowers()
ffrom = [item.screen_name for item in followers]
```

...and some more

- Make network connections. Here I'm creating a network of people I follow who follow me back

```
# make a directed graph
g = networkx.DiGraph()

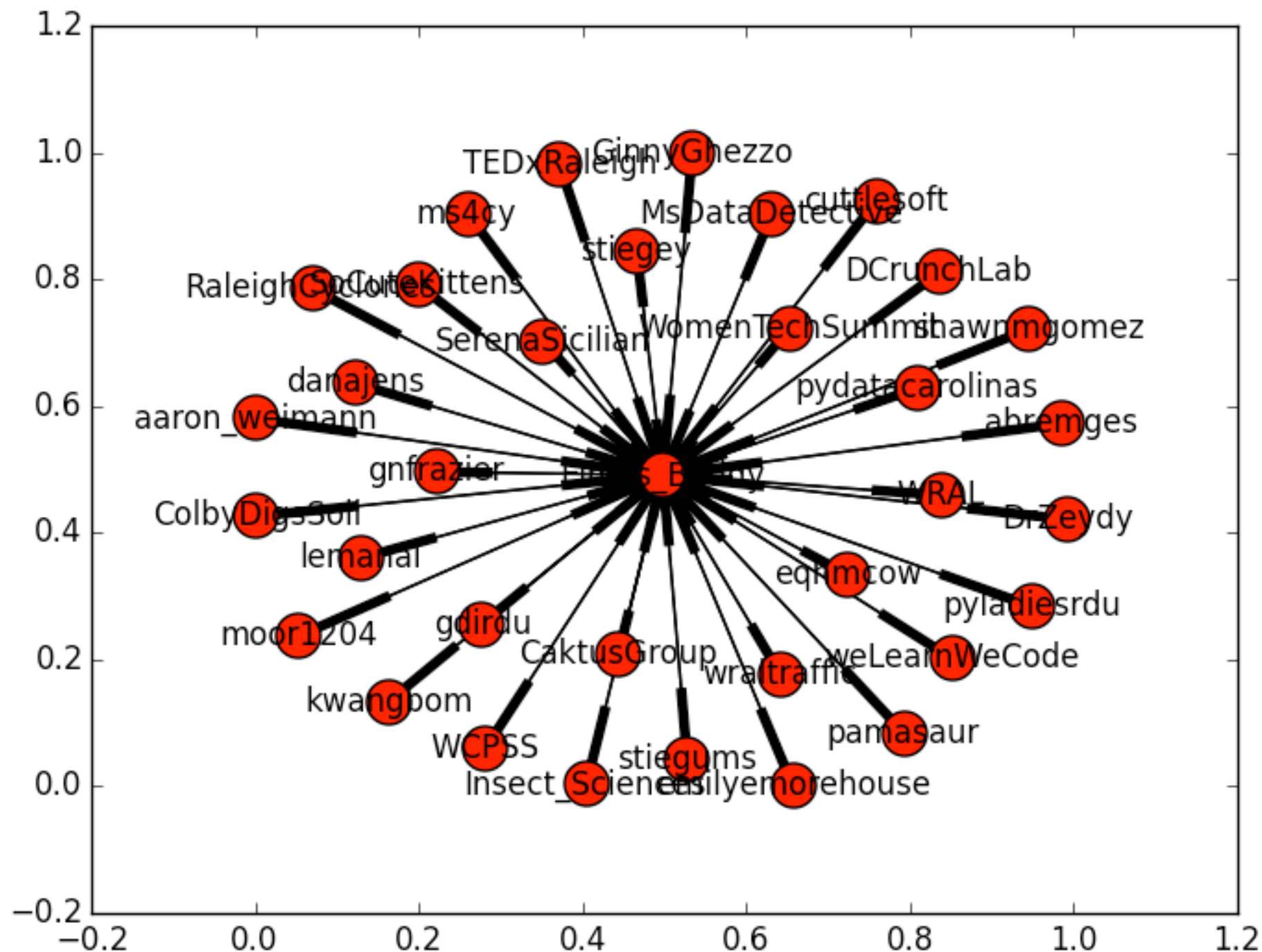
for f in ffrom:
    if f in fto:
        g.add_edge("Elmos_Buddy", f)
        g.add_edge(f, "Elmos_Buddy")
```

- Draw it with matplotlib and/or save it

```
networkx.draw_networkx(g)
matplotlib.pyplot.savefig("blah.png")

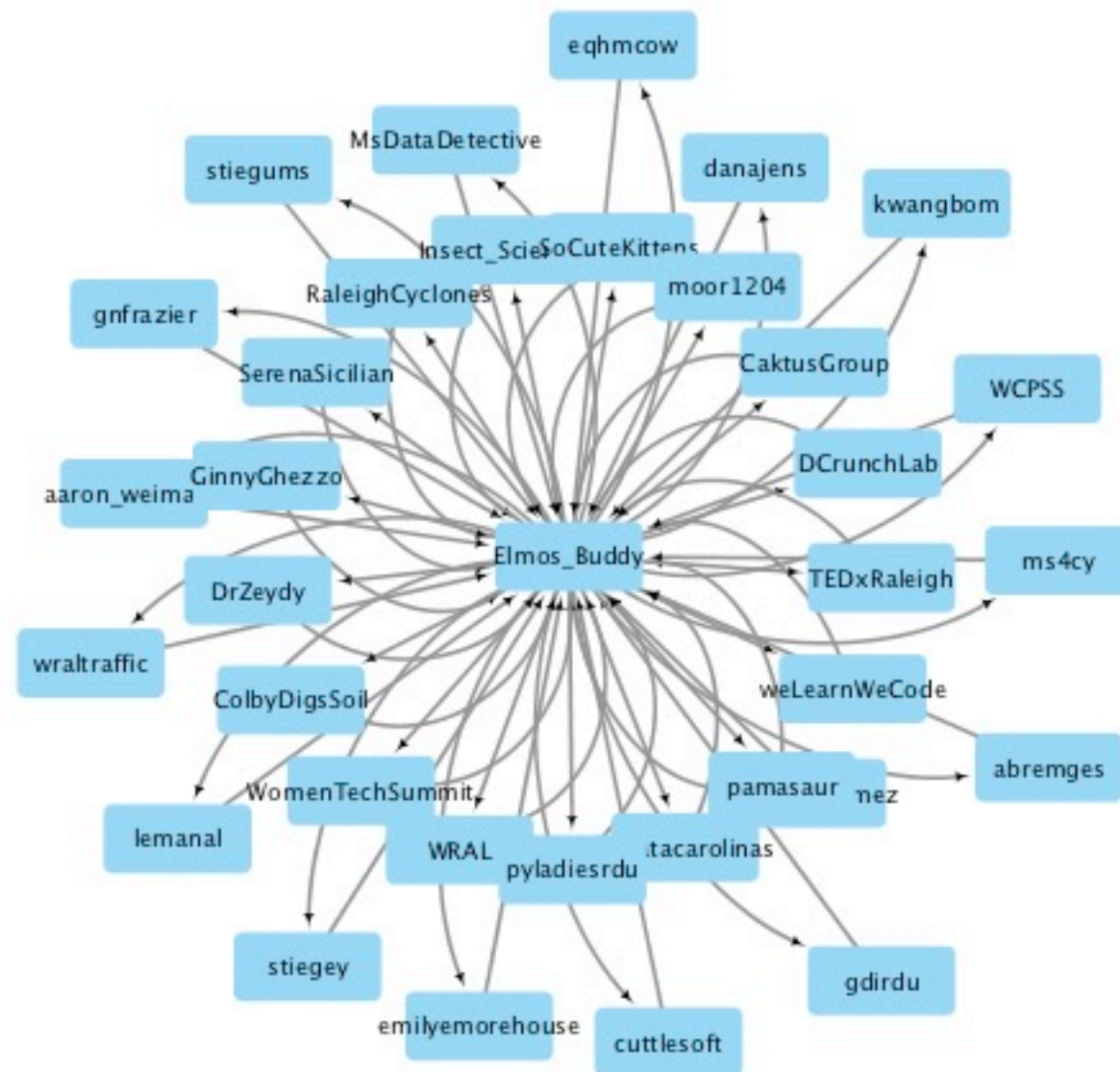
networkx.write_gml(g, "blah.gml")
```


Networkx is not the best for visualizing graphs



Visualize with something else

- Drawn with cytoscape, www.cytoscape.org



What else can you do with networkx?

- Save attributes to nodes and edges
- Many standard graph algorithms
- Find paths between nodes
-
- Active open source project, <http://networkx.github.io/>