

1 Dolphin Network

Paper: *Evidence for Social Role in a Dolphin Social Network*. Lusseau, D. Evolutionary ecology, 2007. Before beginning the programming component of the lab, discuss the following questions.

1. What is the general hypothesis about animal social behavior that this study aims to test?
2. Why are bottleneck dolphins useful to study?
3. There are two behaviors that are noted - what are they and when are they observed?
4. How was the data collected? Are there any potential biases you see about the data collection?

2 Post a Test Graph to GraphSpace

GRAPHSPACE is a webserver that allows researchers to interact with networks. It is located at <http://graphspace.org/>.

2.1 Preliminaries

1. Make an account on GRAPHSPACE.
Username: *This is an email address. Use your Reed email.*
Password: *Use something that you're OK with sharing, in case you hard-code it in your files.*
2. Download the `graphspace-python` Python library for uploading graphs. Instructions for installation are located at <http://manual.graphspace.org/projects/graphspace-python/en/latest/Installing.html>. Remember to use `sudo` if the first attempt fails (this provides admin access to the package can be installed in the Python directory).
3. Test that the packages were properly installed by creating a new `Lab2.py` script and putting these lines at the top of your file:

```
from graphspace_python.api.client import GraphSpace
from graphspace_python.graphs.classes.gsgraph import GSGraph
```

These are two components of the GRAPHSPACE python module that we'll use in this lab.

2.2 Post a Graph to GraphSpace

In this part, you will write a function that will post a test graph to GRAPHSPACE. The documentation for `graphspace-python` library is located at <http://manual.graphspace.org/projects/graphspace-python/en/latest/>.

1. First, create a GRAPHSPACE "session":

```
graphspace = GraphSpace(user_email, user_password)
```

where `user_email` and `user_password` is your username and password. You will pass this `graphspace` object to the `post_test_graph()` function. Now we are ready to make and post a test graph:

```
post_test_graph(): Post a test graph to GRAPHSPACE
```

Inputs: GRAPHSPACE “session” which will allow us to post to GRAPHSPACE

Returns: None

2. Make a small graph with about five nodes and seven edges. Store them as a list of nodes (e.g. ['A', 'B', ...]) and a list of 2-element edges (e.g., [['A', 'B'], ['C', 'A'], ...]).
3. Create a graph object (we'll call it **G**) that will be posted to GRAPHSPACE. Set the title and tags of this graph like so:

```
G = GSGraph()
G.set_name('Test Graph')  ## name this anything you want
G.set_tags(['Lab 2'])  ## tags help you organize your graphs
```

4. Add nodes to your graph **G**. The `add_node()` function takes a required node ID (a string) and an optional node label (also a string). Use a `for` loop to add each node in your node list using the following syntax:

```
G.add_node(node_name, label=node_name)
```

where `node_name` is the name of a node.

5. Add edges to your graph **G**. The `add_edge()` function takes two strings as input: a source and target node IDs; these are simply the node names:

```
G.add_edge(node_name1, node_name2)
```

Use a `for` loop to add all the edges in your edge list.

6. Now, you are ready to post the graph to GRAPHSPACE:

```
graphspace.post_graph(G)
```

7. You will often want to re-upload the same graph with different features and annotations; to do so, you need to write a small helper function to distinguish the case where you are posting a *new* graphspace graph (using the command in #6) and updating an *existing* graphspace graph. Find this small function on the cheat sheet (<https://www.cheatography.com/annaritz/cheat-sheets/graphspace-python-client/>).

8. Add node and edge styles using the options listed on the cheat sheet and the user guide. **Note:** *You must first add nodes/edges before adding node/edge styles.* You can add multiple style types (e.g. red edges, large square nodes, and thick borders); however, make this a single call to `add_node_style()` for each node and a single call to `add_edge_style()` for each edge to avoid some odd behavior.

Note: *HTML colors are allowed, e.g., #73C8F3.* You can select node and edge colors using the HTML color picker at <http://htmlcolorcodes.com/>.

3 Post the Dolphin Network to GraphSpace

The Dolphin network that is described in the Lusseau paper has been parsed in the `dolphin_edgelist.txt` file. This section will post the dolphin network to GRAPHSPACE.

`post_dolphin_network()`: *Post the dolphin network to GRAPHSPACE*
Inputs: GRAPHSPACE “session” which will allow us to post to GRAPHSPACE
Returns: None

1. Parse the dolphin edges and post the graph to GRAPHSPACE. Use the same general steps as you did for the example graph. Refer to the Lab1 References on Moodle for a link to python File input/output help.
2. Annotate the graph with the name and sex of each dolphin (the names are divided into `females.txt`, `males.txt`, and `unknown-sex.txt`). The choice of annotation is up to you (color, size, shape, border, are all options). Add a description of the annotations you selected using the `set_data()` function:

```
G.set_data(data={'description': 'females=squares; males=circles; unkown=stars'})
```

3. Try out different layouts of the graph.
 - (a) Click on the “Change Layout” button to generate the graph using different automatic layouts.
 - (b) Save a manual layout using the Layout Editor. You can also manually lay out the nodes in the main graph page and then save it by selecting “Use Layout Editor” and then “Exit Layout Editor.”

3.1 Challenges (Optional)

1. Further annotate the graph with the “side floppers” (lised in `side-floppers.txt`) and the “upside-down lobtailers” (listed in `upside-down-lobtailers.txt`).
2. Add the degree of each node in the labels.

4 Handin

No code handin is required. Instead, you will share your Dolphin network with the BIOL331F18 Group. On the page that displays the graph, click “share” in the upper right. You should see the group - let Anna know if you cannot share your graph with the group.

Note: *If you have saved manual layouts, make sure they are viewable to all members of the group.*

Suggested:

1. Add comments to your code (this will be useful for posting subsequent graphs).
2. Clean up your GRAPHSPACE graphs by deleting the unused graphs. On the page that displays all graphs, there is an option to remove each graph you have posted.