A live Python coding book on Machine Learning & AI applications to Computational Biology and Bioinformatics

Contents

Author:
Konstantinos Krampis
Associate Professor, Biological Science
Hunter College, City University of New York
Author: Konstantinos Krampis, 2021

Graph Theory and Network analysis in Bioinformatics

- In this lecture we will discuss basic Graph and Network Theory concepts and data structures
- Graphs or otherwise networks are one of the most common ways to represent bioinformatics data
  - Interaction between genes and proteins (genetic regulation) can be represented as graphs
  - Protein structures can also be respensented as graphs denoting the chemical connections between atoms
  - Beyond molecular biology, interacting species for example can be represented as graphs in ecology

Some of the information we cover today can also be found in this wiki article and this online course

---

What is a Graph

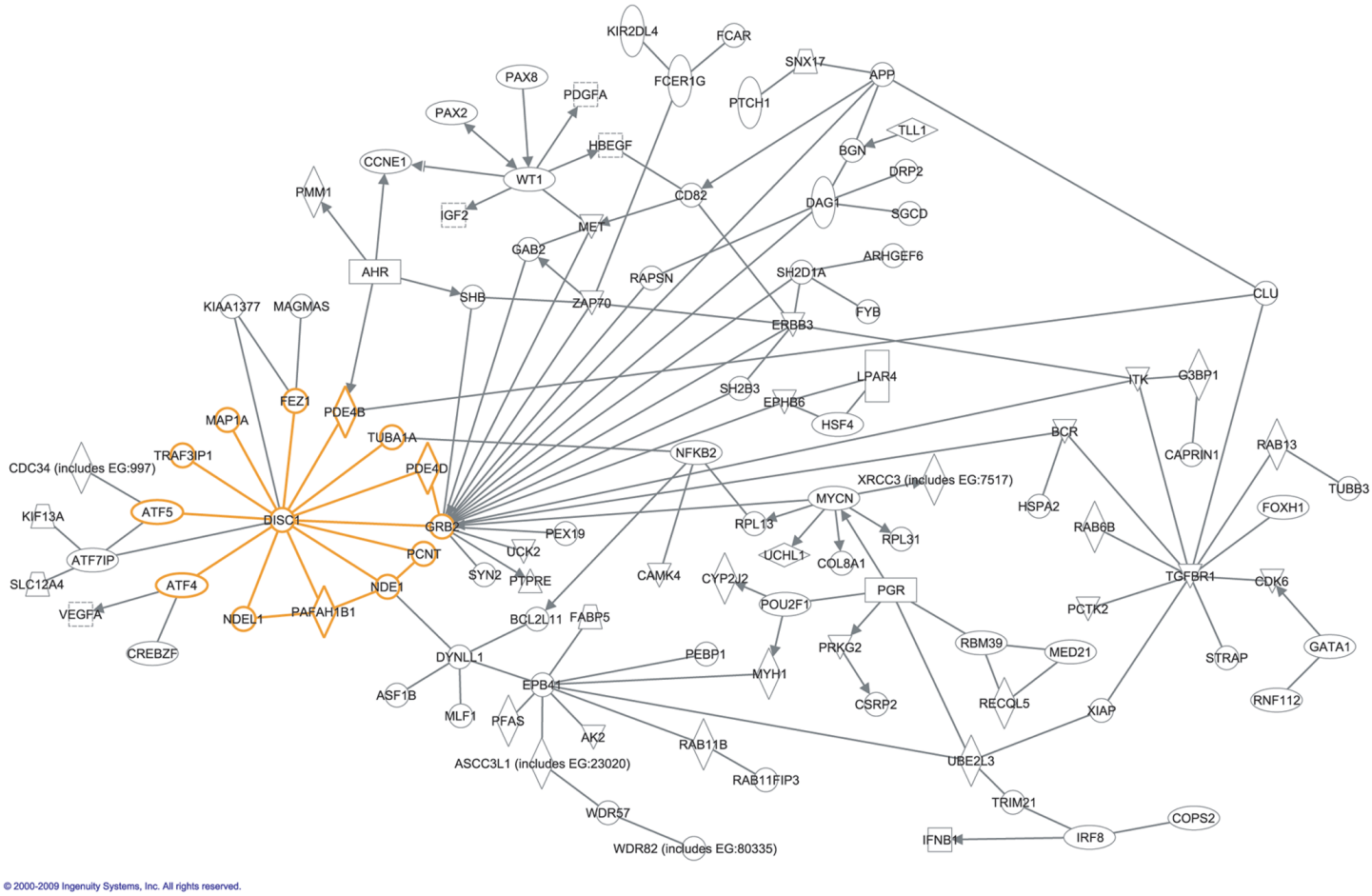
- A mathematical structure capturing binary relationships between objects called *nodes* or *vertices*
- The binary relationships are represented by links or *edges* between the *vertices*
- The *edges* can be of a specific type : *directed or undirected, weighted or not*

- Both *nodes* and *edges* can have attributes attached to them

---

An example of a biological graph

(credit Hennah, Porteus CC BY 2.5 Licence)



The different shapes of the nodes represent different attributes, the edges here are unweighted

---

Protein-protein interaction (PPI) network graphs

- Protein-protein interaction networks (PINs) represent the physical relationship among proteins present in a cell
- Proteins are nodes, and their interactions are undirected edges
- Protein–protein interactions are essential to the cellular processes and well studied networks in biology
- Proteins with high degrees of connectedness (nodes with many links) are more likely to be essential for survival of the cell
- Source of this information and additional reading at this wiki article

---

Let's create a graph in Python

On Google Collab, run first "!pip install matplotlib" and "!pip install networkx" (without the quotes)

```
import networkx as nx
import matplotlib.pyplot

G = nx.Graph()
G.add_node(1)
G.add_nodes_from([2, 3])
```

- We will use a Python library called NetworkX
- First we create an *undirected* graph, meaning it does not matter which direction we travel along the edges

---

We only have nodes (vertices) 1,2,3 let's connect them with edges

```
G.add_edge(1, 2, color='r',weight=2)

e = (2, 3) #another way to add an edge
G.add_edge(*e)

G.add_nodes_from([4, 5]) #add two more vertices
G.add_edges_from([(3, 4), (3, 5)]) #add edges differently

nx.draw(G, with_labels=True, font_weight='bold') #labels used are node numbers
```

Let us add nodes with attributes

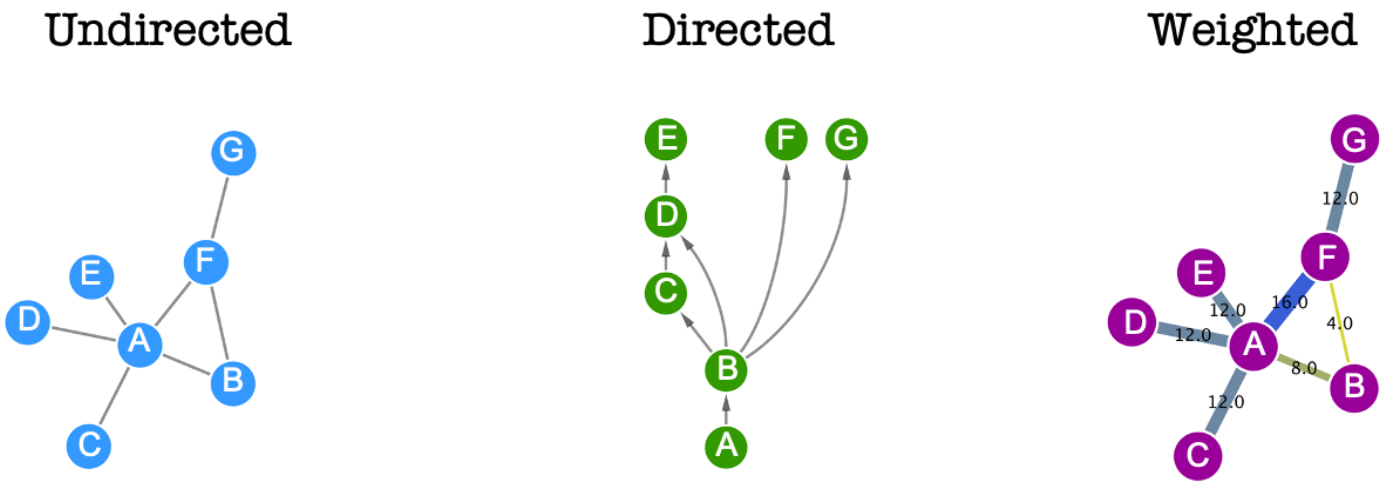
```
G.add_node(6, molecule="protein") #now we also have vertex attribute
G.add_nodes_from([7,8,9], molecule="rna") #add a bunch of RNA vertices
G.nodes[1]["molecule"] = "protein"  #modify attributes of existing vertex
G.add_edges_from([(5,6),(5,7),(7,8),(8,9)])#connect these to the graph

labels = nx.get_node_attributes(G, 'molecule') #using custom labels for nodes
nx.draw(G, labels=labels, font_weight='bold')
```

Refer to these links for the graph coding examples and exercises

Full documentation in NetworkX tutorial and NetworkX reference

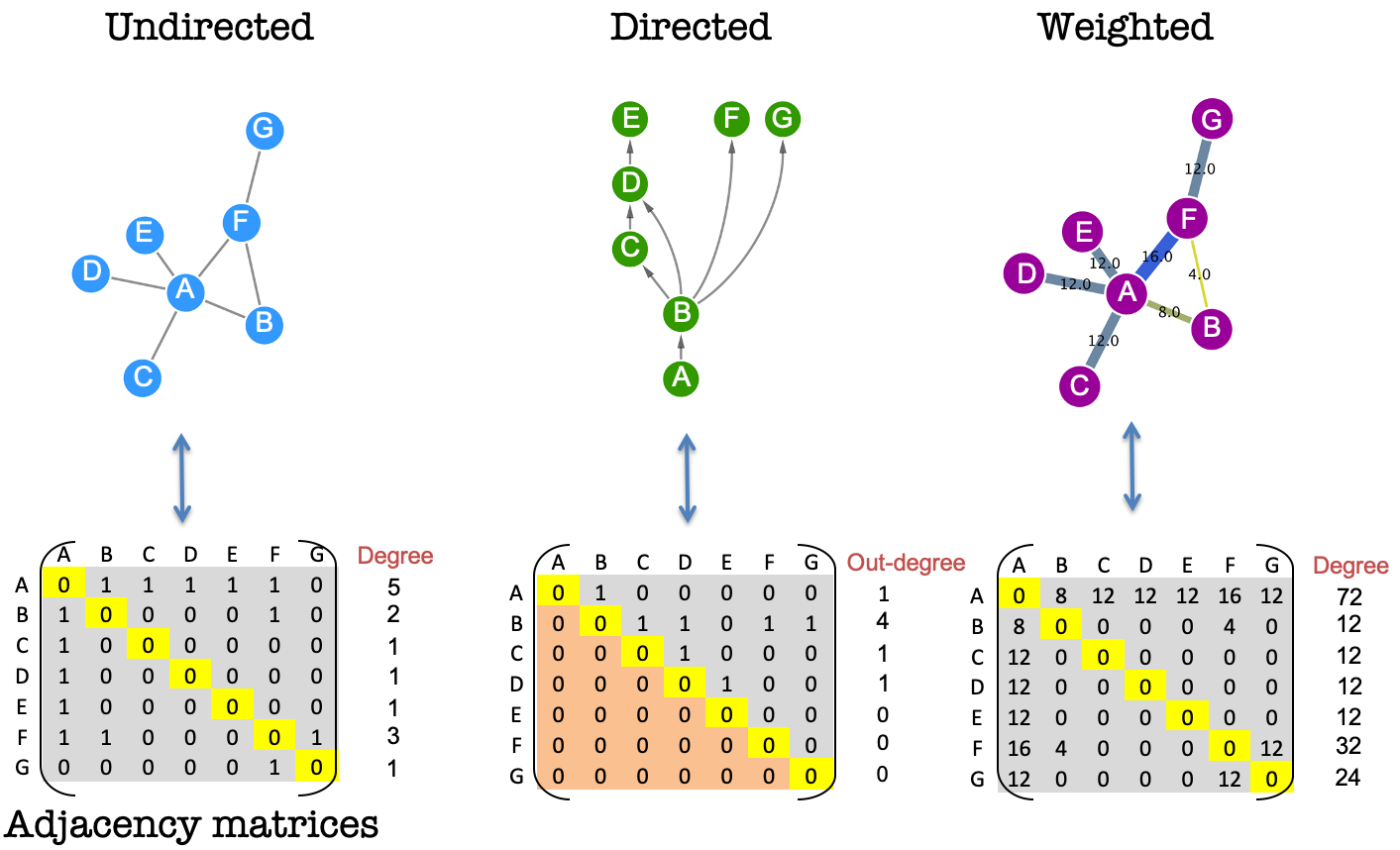
---

Graphs for complex bioinformatics data



(credit European Bioinformatics Institute CC BY 2.5 Licence)

Data structures for graphs data

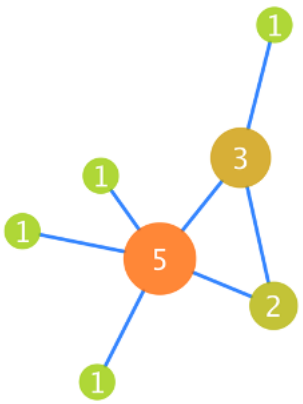


(credit European Bioinformatics Institute CC BY 2.5 Licence)

Let's see this in Python code

- Directed graphs represent directed control, for example biological regulation data
- For example Gene A, regulates Gene B
- Or Protein A controls gene expression of Gene C

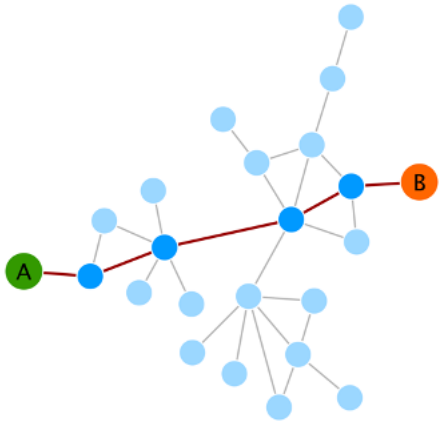
```
G1 = nx.DiGraph() #now we have a directed graph
G1.add_edges_from([(1,2), (2,3), (3, 4), (3, 5)]) #this adds nodes and edges
G1.nodes[1]["name"] = "GeneA"
G1.nodes[2]["name"] = "GeneB"
G1.nodes[3]["name"] = "ProteinA"
G1.nodes[4]["name"] = "GeneC"
labels = nx.get_node_attributes(G1, 'name') #using custom labels for nodes nx.draw(G1, labels=labels, font_weight='bold')
```



(credit European Bioinformatics Institute CC BY 2.5 Licence)

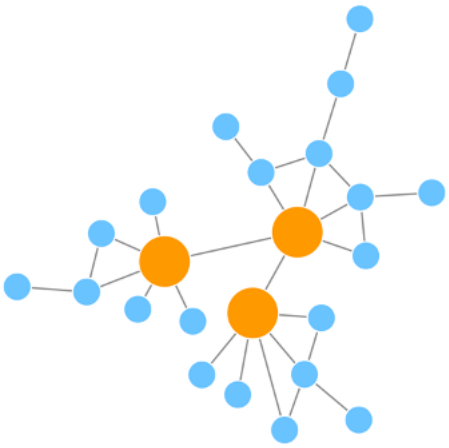
- The degree is the number of edges that connect to a node
- The degree of each node is indicated and reflected in its size and colour
- Directed network nodes have two values : out-degree and in-degree

---



(credit European Bioinformatics Institute CC BY 2.5 Licence)

- Paths between nodes : how many nodes “hops” or steps
- Is used to model how information flows
- Also to model how distant protein / genes control others
- Algorithms to calculate shortest paths on the graph



(credit European Bioinformatics Institute CC BY 2.5 Licence)

- Scale-free networks are those which have high connectivity hubs
- This means (sometimes equally) short paths from one section of the network to the other
- Additional concepts of centrality and transitivity, see here for more details

---

Author: Konstantinos Krampis, 2021