



NILE UNIVERSITY

Biological Networks

A Math211 Project Report

By

Mariam Barakat 202000210

Raghad Hatem 202000571

Salma Megahed 202002800

Sara Onsy 202001706

Submitted to: Dr. Sahar Selim and Eng. Omnia Taha

December 28, 2021

ABSTRACT

The applications of graphs are considered countless due to its powerful underlining truth of connecting entities together. This paper discusses biological networks and their several applications, including niche overlap graphs in ecology and protein interaction graphs. Moreover, we explained the algorithm that is responsible to count connected components in a graph, as this is exceedingly utilized in biological networks. It is concluded that graphs are powerful tools that aid scientists to find connections that are rather impossible to find alone.

Table of Contents

ABSTRACT	2
SECTION I: Introduction	5
SECTION II: Methods	5
1. Directed Graph and Undirected Graph	6
2. Simple and Multigraph	7
3. Bipartite Graph	7
SECTION III: Literature Review	8
Biological Networks	8
1. Niche Overlap Graphs in Ecology	8
2. Protein Interaction Graphs	9
SECTION IV: The Model	11
Input	11
Algorithm	12
Flowchart	13
SECTION V: Results	15
SECTION VI: Conclusion	16
References	17

Tables of Figures

Figure 1 Directed and Undirected. Adapted from [2].....	6
Figure 2 Simple and Multigraph Adapted from [3].....	7
Figure 3 Bipartite Graph. Adapted from [4].	7
Figure 4 Food Web. Adapted from [6].	8
Figure 5 Pollination Networks. Adapted from [7].	9
Figure 6 Protein Interaction Graph. Adapted by [10].	10
Figure 7 Graph Connectivity. Adapted by [11].	11
Figure 8 Adjacency List. Adapted from [12].	12
Figure 9 Example on Connected Components.....	13
Figure 10 Connected Components Flowchart.....	14
Figure 11 Example.....	15
Figure 12 Our Output.....	15

SECTION I: Introduction

Graphs are one of the mystically diverse mathematical concepts that blend seamlessly with a myriad of disciplines, including biology. To illustrate, graphs are tools that represent relationships between different entities while maintaining the connection between each entity. This can be heavily utilized in biology where there are millions of connections between different components, as the graph will summarize and synthesize the connections smoothly. To clarify, any ecosystem considered, contains living and non-living components that affect each other, graphs are used to represent those relationships; thus, scientists can understand and contemplate the connections and characteristics of those components.

This paper aims to provide information regarding the different applications of networks in biology. This includes niche overlap graphs in ecology and protein interaction graphs. The paper commences by explaining important graph terminology, then goes deeper in biological networks and their applications. Proceeding to this, we explained the code that is responsible to count the number of connected components and how it may relate to biological networks.

SECTION II: Methods

In this section, we will focus on important graph categorizations that will help us to describe biological networks effortlessly. To start, a graph consists of nodes and edges. Nodes are the components to explore, and the edges are the relationship between components. To clarify, if we have a miniature graph that describes the relationship between cats and mice, then cats and mice are our nodes and an edge will be between them to represent they are related. However, if the edge means this node eats this node, then we need the edge to be directed from the cat to the mouse. This concept is explained more thoroughly below.

1. Directed Graph and Undirected Graph

Firstly, we will discuss the types of graphs according to their direction of edges. An undirected graph is a graph where the edges have a both forward and backward relationship. On the other hand, a directed graph is a graph where the edges have a specific flow of relation [1].

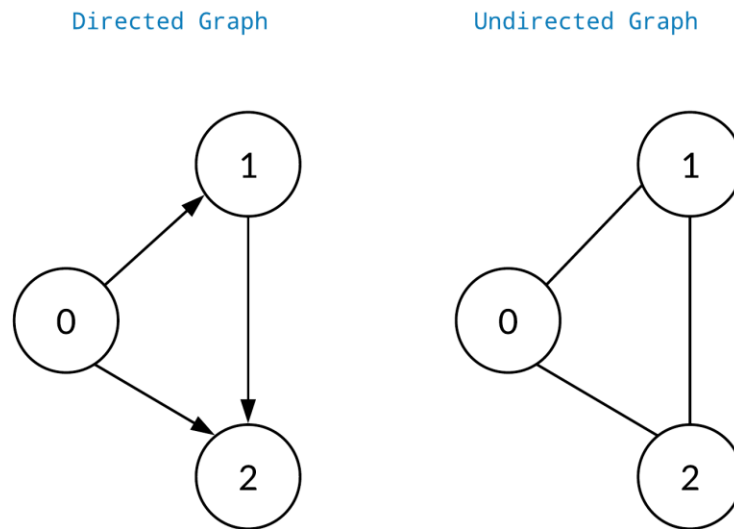


Figure 1 Directed and Undirected. Adapted from [2]

In figure 1 the first graph is directed as node 0 has a directed relation with nodes 1 and 2, but not the other way around. On the other hand, the second graph is undirected as node 0 has a relationship between 1 and 2, and nodes 1 and 2 have a relationship between node 0. For more clarification, directed graphs are used when the edge only has a relationship in a specific direction with the adjacent node. Contrastingly, undirected graphs describe the back-and-forth relationship between the nodes.

2. Simple and Multigraph

Furthermore, graphs can be also categorized according to the number of edges between nodes. To clarify, a graph that has only one link between its nodes is called a simple graph. Moreover, a multigraph is when multiple edges connect the same two vertices. Figure 2 illustrates their differences [1].

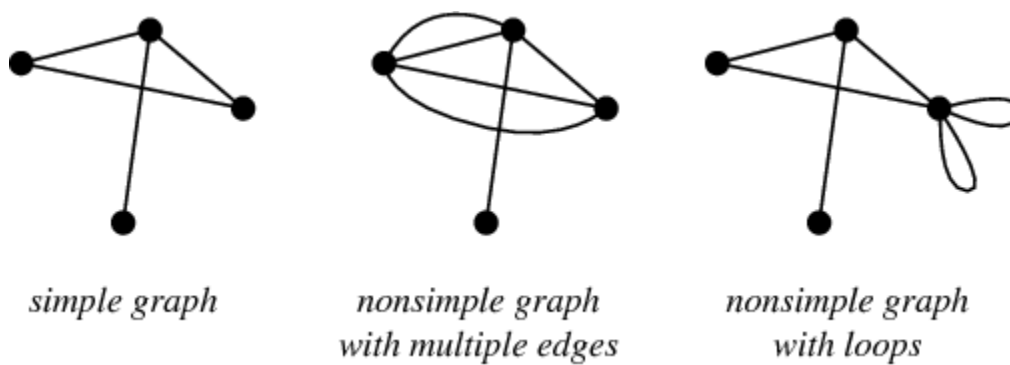


Figure 2 Simple and Multigraph Adapted from [3]

3. Bipartite Graph

A bipartite graph is when nodes can be divided into two disjoint sets and each edge connects to a vertex of the other set, this property gives the graph the ability to be colored in two colors where each node only points to the other color, this is described in figure 3 [1]

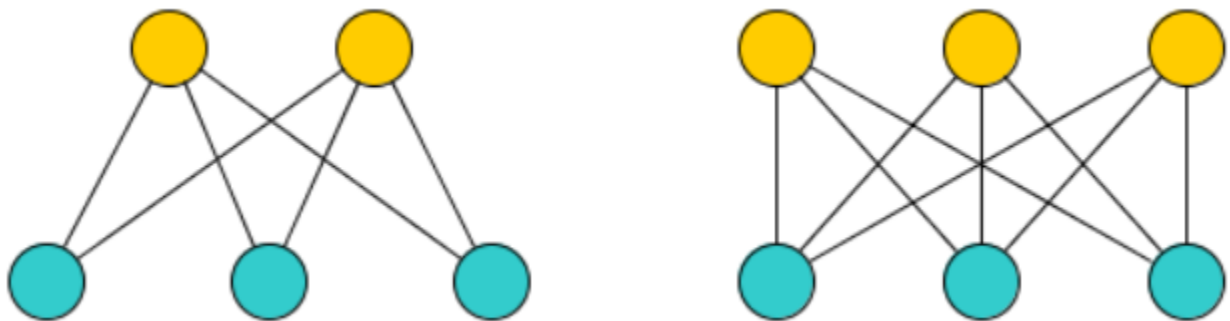


Figure 3 Bipartite Graph. Adapted from [4].

SECTION III: Literature Review

Biological Networks

Biology is fundamentally how organisms react to the environment. This hints that some entity will affect another, meaning it can be modeled using graphs. Numerous biological networks are used by scientists, this includes, protein-protein interaction networks, metabolic networks, genetic interaction networks, transcriptional regulatory networks, cell signaling networks, and ecological networks. This paper will discuss ecological networks and protein interaction networks.

1. Niche Overlap Graphs in Ecology

In biology niche overlap is when two organisms are competing for limited natural resources, therefore they start to evolve into similar characteristics [5]. Graphs are heavily used in representing ecological networks, which includes the relationship between species, energy flow, etc., those interactions create complex relationships which can be represented using graphs [5].

There are various types of ecological networks. Firstly, food webs which are interconnections of food chains, it displays the connection of what organisms eat. This is represented in a directed multigraph, where the edges tell what the organism eats [5].

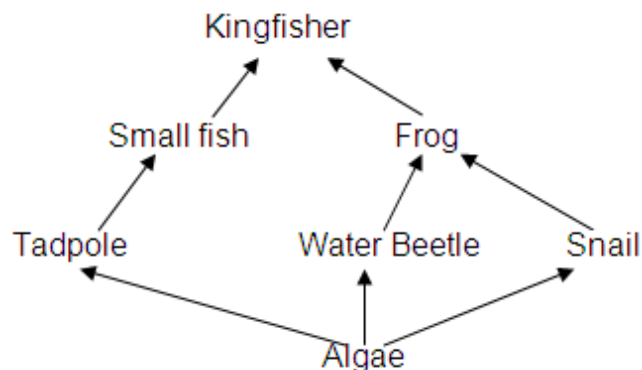


Figure 4 Food Web. Adapted from [6].

Figure 4 represents a simple food web, where the edges display the flow of energy. For example, kingfishers eat frogs, and frogs eat Algae. Notice that arrows are only one way and cannot work the other way around.

Additionally, another ecological network is pollination networks. This network is represented in a bipartite graph.

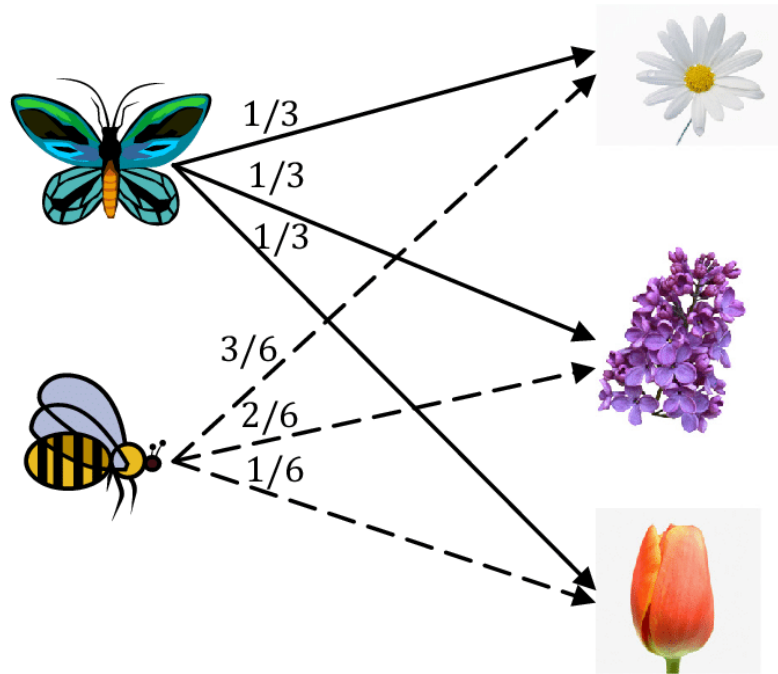


Figure 5 Pollination Networks. Adapted from [7].

In figure 5, we can see that the set of pollinators is connected to the set of plants, but the set of plants is not connected, nor the set of pollinators. This is a simple and directed graph, and the edges represent which pollinator goes to which plant.

2. Protein Interaction Graphs

Protein-protein interactions (PPIs) is how the different proteins are related, and it helps scientists comprehend the function of proteins and how they affect cellular biology [8]. This provides valuable information for drug development and disease diagnosis. Therefore, graphs help

us visualize such complex interactions [8]. For example, STRING database contains 3,123,056,667 total protein interactions; surprisingly, those interactions are between 20,000,000 proteins in over 5,000 organisms [9]. Imagine the number of underlining connections ready to be discovered which will account for unlocking biology's deepest secrets.

Figure 6 Protein Interaction Graph. Adapted by [10].

Figure 6 represents the protein interaction graph, this is an undirected and multigraph, which displays the interactions between different proteins. Intriguingly, you can see groups of nodes clustering together, meaning that those groups of proteins are closely related and depend on each other. It is fascinating how graphs can say so much in just pictorial representations.

SECTION IV: The Model

Biologists utilizing biological networks need to understand their graphs more, this is done by running computations that calculate the number of nodes and edges, whether it is cyclic or acyclic or even to count the number of components. We will focus on the latter; biologists use this concept to understand the connectivity of their graph and how many systems are related together.

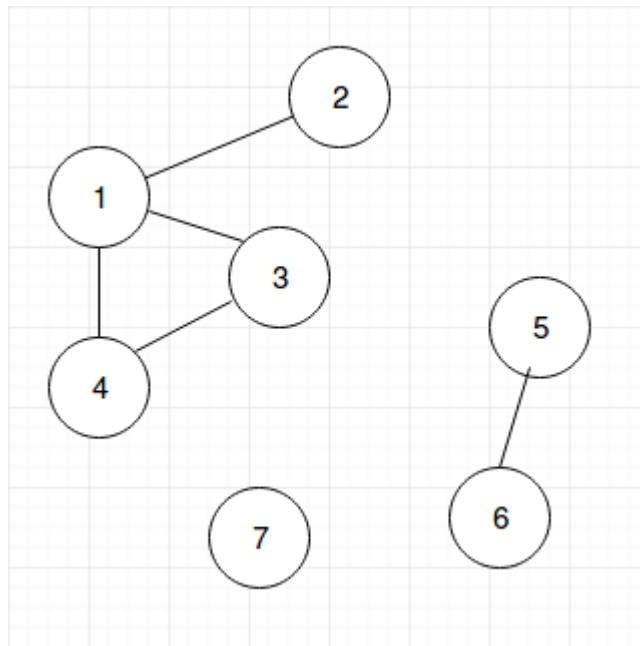


Figure 7 Graph Connectivity. Adapted by [11].

In figure 7, this graph has three connected components. If interpreted scientifically, we could say that $\{1,2,3,4\}$ are related together as they are all connected. However, the latter set is not connected to $\{5,6\}$ nor $\{7\}$, as there is no connecting edge.

Input

The most suitable format to input our graph is using an adjacency list, as it will allow us to smoothly move from one node to another while exploring its adjacent nodes. This is represented in the form of a list holding the nodes, and another list related to each node containing the adjacent nodes [1].

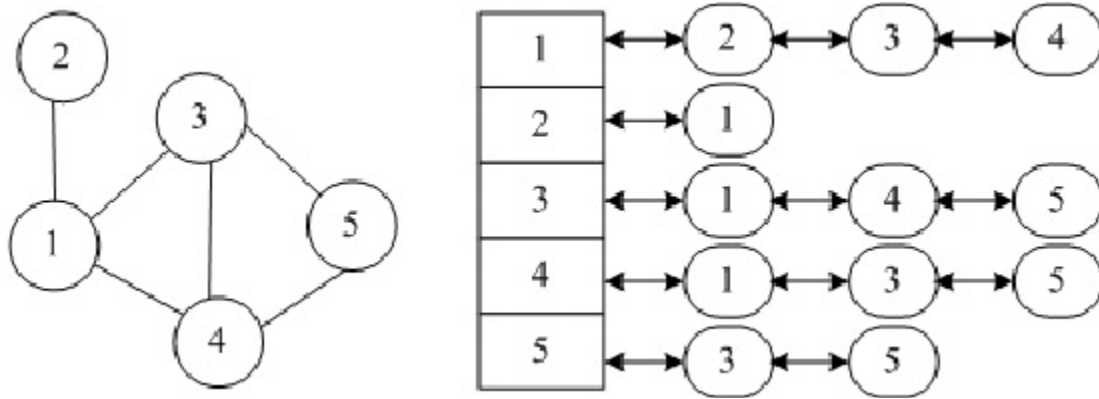


Figure 8 Adjacency List. Adapted from [12].

In figure 8, we can see that the graph is represented linearly. For example, the first node {1}, has adjacent nodes of {2,3,4}. Unfortunately, there is a minor error in figure 8, as researchers creating this graph wrote that 5 has an adjacent list of {3,5}, even though it is {3,4}.

Algorithm

To count the number of connected components, we first need to traverse the entire graph as this will allow us to check the connectivity of each node. We have chosen Depth First Search (DFS) as our way of traversal. This traversal starts from the source node and explores all the adjacent nodes recursively. A DFS traversal for figure 8 may result in {1, 2, 3, 5, 4}. Moreover, we have a Boolean array that stores whether we have visited the node of interest or not. This will not only help in DFS but will also aid in calculating the number of connected components. In other words, The DFS will end when all connected components are visited, thus, to count the next connected component we will re-do DFS but on the next unvisited node, this will be repeated until all the nodes are visited.

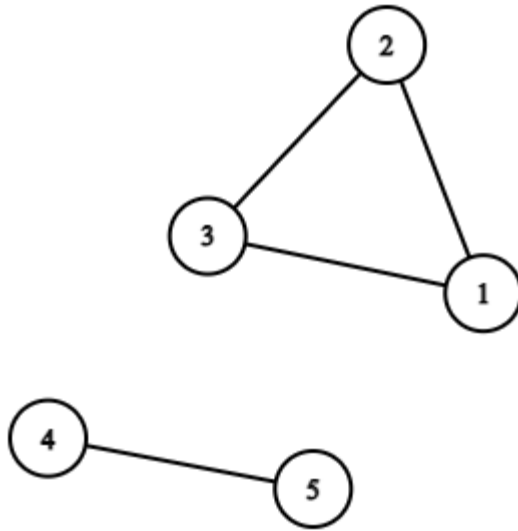


Figure 9 Example on Connected Components

In figure 9, when we run the connected components algorithm, the expected output will be 2. As {1,2,3} are one connected entity and {4,5} are one connected entity.

Flowchart

The flowchart in figure 10 explains the algorithm responsible to count the number of connected components. The program starts with taking the number of nodes and edges, then the user inputs the node and its adjacent node until all edges are consumed. Then, the program starts to traverse using DFS and increments the counter whenever a DFS call is done, when a DFS call is done it means that all the *already* connected components have been visited. Consequently, the program will check if *all* the nodes have been visited, if not all nodes were visited the program will redo DFS on an unvisited node and traverse on all its connected components, else will print the number of connected components and terminate.

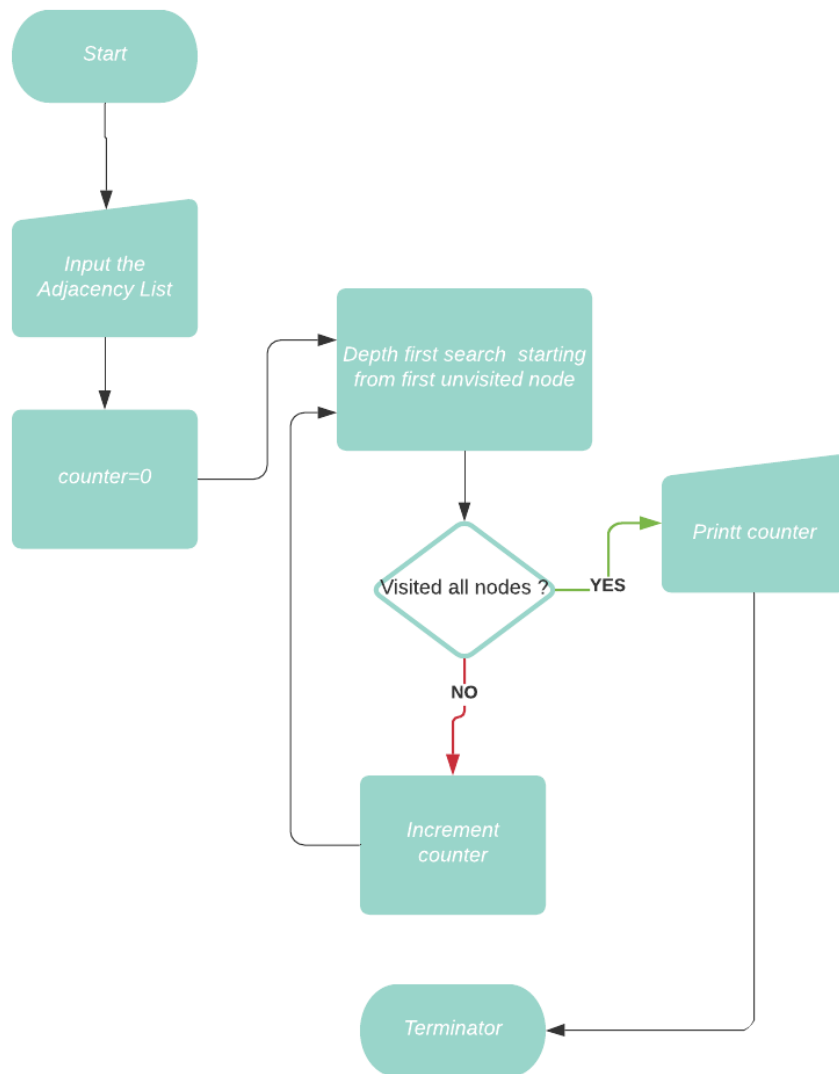


Figure 10 Connected Components Flowchart

SECTION V: Results

To illustrate the interface, we will demonstrate an example below. Figure 10 below represents our input graph pictorially. It is obvious we have 2 connected components.

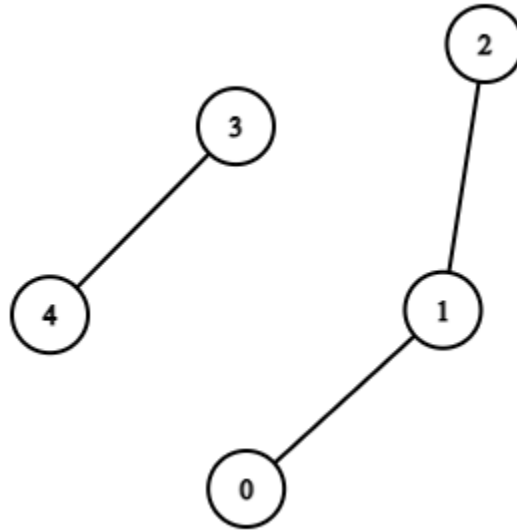


Figure 11 Example

```
=====Connected Components Algorithm=====
Please input the number of nodes and edges respectively:      5 3
Node : 0
Adjacent node : 1
Node : 1
Adjacent node : 2
Node : 3
Adjacent node : 4
This graph is not connected
Number of connected components  2

Process returned 0 (0x0)   execution time : 22.928 s
Press any key to continue.
```

Figure 12 Our Output

Figure 12 represents our interface; the user entered 5 nodes and 3 edges. Therefore, the program will ask for a node and its adjacent node three times to create the graph in memory. After the DFS is run many times, it outputted that our graph is not connected and that there are 2 connected components, which was verified by figure 11.

SECTION VI: Conclusion

In conclusion, in this paper, we discussed some graph terminology and the importance of biological networks using examples of ecological networks and protein interaction graphs. Those examples illustrated the usage of graphs in biology, and how graphs are powerful tools in representing relationships between different entities. Consequently, we got deep into the algorithm that executes the number of connected components, and how it may be used in biological networks.

References

- [1] K.H. Rosen, Discrete mathematics and its applications, McGraw-Hill, New York, NY, 2019.
- [2] Directed & undirected graphs, Graphs | CodePath Android Cliffnotes. (n.d.).
<https://guides.codepath.com/compsci/Graphs> (accessed December 26, 2021).
- [3] A.G. Evans, What is the gist of "simple graphs," "multigraphs," and pseudographs" (graph theory)?, Ruficalix. (2020). <http://ruficalix.com/gist/what-is-the-gist-of-simple-graphs-multigraphs-and-pseudographs-graph-theory/> (accessed December 26, 2021).
- [4] A.M. Joan and W.J. Robin, “*Graphs and Applications- An Introductory Approach*”, Springer, 2000. [Online]. Available: <https://link.springer.com/book/9781852332594> (accessed December 26, 2021).
- [5] H. Cornell, “Niche Overlap,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/273764925_Niche_Overlap. [Accessed: 18-Dec-2021].
- [6] “Food chains, food webs, biomass pyramids, and cycles,” *Qld Science Teachers*. [Online]. Available: <https://www.qldscienceteachers.com/junior-science/biology/food-chains-webs-biomass-pyramid-cycles>. [Accessed: 26-Dec-2021].
- [7] R. Rader, “Pollination Network.” *ResearchGate*, 2015.
https://www.researchgate.net/figure/Pollination-network-with-two-pollinator-and-three-plant-species-Interaction_fig1_281672438. [Accessed: 26-Dec-2021].

- [8] F. Yang, K. Fan, D. Song, and H. Lin, “Graph-based prediction of protein-protein interactions with attributed signed graph embedding,” *BMC Bioinformatics*, vol. 21, no. 1, 2020.
- [9] F. Combs, “Visualizing Protein Networks in python,” *Medium*, 11-Dec-2020. [Online]. Available: <https://towardsdatascience.com/visualizing-protein-networks-in-python-58a9b51be9d5>. [Accessed: 18-Dec-2021].
- [10] L. P. Julianna, “Protein-Protein Interactions Get a New Groove On: Adding a modern twist to a traditional assay-the yeast two-hybrid system-A-Alpha Bio measures millions of PPIs at a time,” *Genet. eng. biotechnol. news*, vol. 39, no. 11, pp. 30–32, 2019.
- [11] “Codaholic.cpp,” *Blogspot.com*. [Online]. Available: <https://sleepincode.blogspot.com/2017/07/finding-connected-components-using-dfs.html>. [Accessed: 26-Dec-2021].
- [12] “Fast algorithm for searching adjacent communities and its application in hierarchical community discovery,” *Researchgate.net*. [Online]. Available: https://www.researchgate.net/publication/228563151_Fast_algorithm_for_searching_adjacent_communities_and_its_application_in_hierarchical_community_discovery. [Accessed: 26-Dec-2021].