Practical 4: NCBI Taxonomy

Ximena Moure Eliya Tiram

10/10/2023, submission deadline 16/10/2023

1. **Given a file nodes.dmp for the NCBI taxonomy, write a Python script to store it in a rooted tree. Give the code of your Python script as your answer to this question**
   The rooted tree we have stored the data in is in the following python code. Since we build a rooted tree, as requested in the question, we ignore self loops, for example 1—1 in the data.

```python
import networkx as nx

# Define a function to process the data and construct the graph
def construct_tree_from_file(file_path):
    # Create a directed graph
    graph = nx.DiGraph()

    with open(file_path, ’r’) as file:
        for line in file:
            columns = line.strip().split(’|’)
            if len(columns) >= 2:
                parent, child = map(int, columns[:2])

                # Skip self-loops (edges where parent and child are the same)
                if parent != child:
                    # Add nodes to the graph
                    graph.add_node(parent)
                    graph.add_node(child)

                    # Add an edge to represent the parent-child relationship
                    graph.add_edge(parent, child)

    return graph

# Path to the input file
file_path = "nodes.dmp"

# Construct the tree from the data in the file
tree = construct_tree_from_file(file_path)

```

Listing 1: Python code ex1

2. **Is the NCBI taxonomy a rooted tree, a directed acyclic graph, or a directed graph with cycles?**
   The NCBI taxonomy is a directed graph with cycles. In this section we do not ignore self loops so we will know the NCBI taxonomy type.

```python
import networkx as nx

# Define a function to process the data and construct the graph
def construct_tree_from_file(file_path):
    # Create a directed graph
    graph = nx.DiGraph()

    with open(file_path, ’r’) as file:
        for line in file:
            columns = line.strip().split(’|’)
            if len(columns) >= 2:
                parent, child = map(int, columns[:2])
                # Add nodes to the graph
                graph.add_node(parent)
                graph.add_node(child)
                # Add an edge to represent the parent-child relationship
                graph.add_edge(parent, child)

    return graph

# Path to the input file
file_path = "nodes.dmp"

# Construct the tree from the data in the file
tree = construct_tree_from_file(file_path)

# Check if it's a rooted tree, DAG, or graph with cycles
if nx.is_tree(tree):
    print("The NCBI taxonomy is a rooted tree.")
elif nx.is_directed_acyclic_graph(tree):
    print("The NCBI taxonomy is a directed acyclic graph (DAG).")
else:
    print("The NCBI taxonomy is a directed graph with cycles.")

```

Listing 2: Python code ex2

3. **How many nodes are there in the NCBI taxonomy?**
   In this section we ignore self loops again and we get the number of nodes in the NCBI taxonomy is 2442791 and the number of edges in the NCBI taxonomy is 2442790.

```python
# Number of nodes and edges in the NCBI taxonomy
num_nodes = len(tree.nodes())
num_edges = tree.number_of_edges()
print(f"The number of nodes in the NCBI taxonomy is {num_nodes}.")
print(f"The number of edges in the NCBI taxonomy is {num_edges}.")

```

Listing 3: Python code ex3

4. **Write a Python script to restrict the rooted tree to the seven standard taxonomic ranks: kingdom, phylum, class, order, family, genus, species.**

```python
import networkx as nx

# Define the valid ranks
valid_ranks
    = ["kingdom", "phylum", "class", "order", "family", "genus", "species"]


def construct_tree_from_file(file_path):
    # Create a directed graph
    graph = nx.DiGraph()

    # Read the file and build the entire graph
    with open(file_path, 'r') as file:
        for line in file:
            columns = line.strip().split('\t|\t')
            if len(columns) >= 3:
                node_id, parent_id, rank = map(str.strip, columns[:3])
                node_id, parent_id = map(int, [node_id, parent_id])
                # Add nodes to the graph with rank attribute
                graph.add_node(node_id, rank=rank)
                graph.add_node(parent_id)
                # Add an edge to represent the parent-child relationship
                graph.add_edge(parent_id, node_id)

    # Identify nodes to be removed and nodes to be retained
    not_valid_nodes = []
    for node, attrs in graph.nodes(data=True):
        # Check if the rank of the current node is not in the valid ranks
        if attrs['rank'] not in valid_ranks:
            # If the rank is not valid, add the node to the not_nodes list
            not_valid_nodes.append(node)

    restricted_nodes = [node for node, attrs in graph.nodes(data=True) if attrs['rank'] in valid_ranks]

    # Create a copy of the original graph for modification
    rest_tax = graph.copy()

    # Lists of successors and predecessors for the nodes which will be removed
    successors = [list(graph.successors(i)) for i in not_valid_nodes]
    predecessors = [list(graph.predecessors(i)) for i in not_valid_nodes]

    # Add edges from the predecessor to the successors of nodes to be removed
    rest_tax.add_edges_from(
        [(predecessors[i][0], successors[i][j]) for i in range(len(successors)) for j in range(len(successors[i]))])

    # Restrict the graph to nodes of standard ranks
    rest_tax = nx.subgraph(rest_tax, restricted_nodes)

    return rest_tax  # Return the restricted graph and the root node


# Path to the input file
file_path = "nodes.dmp"
restricted_taxonomy = construct_tree_from_file(file_path)
num_nodes = restricted_taxonomy.number_of_nodes()
num_edges = restricted_taxonomy.number_of_edges()
print(f"The number of nodes in the restricted taxonomy is {num_nodes}.")
print(f"The number of edges in the restricted taxonomy is {num_edges}.")

```

Listing 4: Python code ex4

5. **What is the name of the kingdom taxonomic rank in the NCBI taxonomy?**
   Superkingdom

6. **How many nodes are there in the NCBI taxonomy, once restricted to the seven standard taxonomic ranks?**
   There are 2112007 nodes.