

Assignment: Practicing Graph Analytics in Neo4j With Cypher

Practicing Graph Analytics in Neo4j With Cypher

In order to truly understand how to apply analytics techniques to graph networks, you need to go through the necessary steps to create a query using Cypher, submit that query to Neo4j, and interpret the results. You've seen this -- and maybe even followed along yourself -- in videos. This assignment is a chance to solidify your knowledge by practicing the same tasks on a new data set!

Learning Outcomes

After completing this assignment you will be able to:

- 1. Create your own Cypher queries by adapting existing queries.
- 2. Submit queries to Neo4j and interpret the results.
- 3. Describe the advantages and disadvantages of methods for analyzing graphs using different approaches including path analytics and connectivity analytics.

Introduction

The best way to understand graph analytics and the Cypher query language is to begin with existing query examples and modify them to apply to different datasets. In this assignment you will take existing query examples and adapt them to apply to the example dataset we provided containing the first 50,000 rows of the larger gene-gene association dataset.

Resources

You will need to have Neo4j container running in your browser.

You will also need the *big-data-5/gene_gene_associations_50k.csv* file.

What you will do

1. Explore the csv file. Be sure that the *gene_gene_associations_50k.csv* dataset contains 50,000 rows and three columns of data. This file is the same which will be located inside *datasets* in your Docker container.

2. Modify an existing script to import the dataset into Neo4j.

In [Importing Data Into Neo4j](#), you were shown how to write a Cypher script to import a dataset of a simple road network graph. We include that script here for your convenience:

```
1  LOAD CSV WITH HEADERS FROM "file:///datasets/test.csv" AS line
2  MERGE (n:MyNode {Name: line.Source})
3  MERGE (m:MyNode {Name: line.Target})
4  MERGE (n) -[:TO {dist: toInteger(line.distance)}]-> (m)
```

Here are commands to clear your Neo4j database:

```
1  match (n)-[r]-() delete n, r;
2  match (n) delete n
```

The first row deletes all edges and their corresponding nodes, the second command deletes all nodes with no edges.

Modify the above script according to the following:

- Load the *gene_gene_associations_50k.csv* (instead of the *text.csv*),
- Define the node type to be *TrialGene*,
- Add a *Name* property to the source node and assign the *OFFICIAL_SYMBOL_A* column values to it,
- Add a *Name* property to the target node and assign the *OFFICIAL_SYMBOL_B* column values to it,
- Define the edge type to be *AssociationType*,
- Give each edge a property named *AssociatedWith* and assign the content of the column in the dataset with the heading *EXPERIMENTAL_SYSTEM*.

This is a large file, so it may take a few minutes to load, so please be patient. We also recommend that you do not view the full graph, as these visualizations may slow down your system.

Need help? Review the following reading:

[Hands-On: Importing data into Neo4j](#)

3. Perform the following analyses and document your results in order to answer the questions in the accompanying quiz (e.g. You will answer a question for each numbered item below. We suggest you write down that answer as you go along, then enter it in the quiz when you are done):

- 1. Calculate number of nodes in the graph.
- 2. Calculate the number of edges in the graph.
- 3. Calculate the number of loops in the graph.
- 4. Submit the following query and report the results.

```
1  match (n)-[r]->(m)
2  where m <> n
3  return distinct n, m, count(r)
```

5. Interpret the results of the query in Step 4 above.

6. Submit the following query and report the results:

```
1  match (n)-[r]->(m)
2  where m <> n
3  return distinct n, m, count(r) as myCount order by myCount desc limit 1
```

7. Run the following query and interpret the results:

```
1  match p=(n {Name: 'BRCA1'})-[:AssociationType*..2]->(m)
2  return p
```

[Need help with Questions 1-7? Review the [Basic Queries](#)]

8. Count how many shortest paths there are between the node named 'BRCA1' and the node named 'NBRI'. [Need help? Review [Path Analytics](#)]

9. Find the top 2 nodes with the highest outdegree. [Need help? Review [Connectivity Analytics](#)]

10. Modify one of the Cypher queries we provided and create the degree histogram for the network, then calculate how many nodes are in the graph having a degree of 3. [Need help? Review the [Connectivity Analytics](#)]

Go to next item

✓ Completed