Cabeçalho:

- Bernardo Bertante Martins
- 202220172
- GRAFOS
- GCC218
- VINICIUS VITOR DOS SANTOS DIAS

Definição e apresentação da base de dados:

A base de dados utilizada nesse projeto consiste de PROTEÍNAS como vértices e as LIGAÇÕES PROTÉICAS como as arestas que compõe o Grafo. O intuito principal do Grafo é primeiramente mostrar que, em termos de importância, proteínas com um alto grau na rede tendem a ser consideradas mais essenciais. Isso porque estão envolvidas em várias interações, desempenhando papéis cruciais na conectividade e funcionamento da rede de interações proteicas.

Em segundo lugar, o principal intuito é a analise de proteínas/vértices que funcionam como articulação, ou seja, é um vértice cuja remoção do grafo aumenta o número de componentes conectados. Em outras palavras, se você retirar esse vértice do grafo, isso resultará em dois ou mais componentes conexos independentes onde antes existia apenas um único componente. Essas proteínas de “articulação" são importantes para a comunicação entre grupos de proteínas. Elas facilitam a transmissão de sinais ou substâncias entre diferentes regiões da rede proteica, desempenhando um papel fundamental na integração e no funcionamento coordenado das diferentes partes do sistema biológico.

Logo, de forma resumida, estamos analisando nesse grafo quais são as proteínas críticas que compõe o todo, as quais em sua remoção, teríamos impactos críticos no funcionamento metabólico da vida.

Se trata de um Grafo bem esparso, ou de pouca densidade (0.000758), isso pois o máximo de arestas possíveis com 7393 vértices seria de 27324528 arestas, porém o que de fato temos é 25569. Além disso, notamos que há ciclos no código, logo se trata de um grafo cíclico em sua totalidade.

Se trata também de um grafo não direcionado, ou seja, não há arestas que apontam unicamente para outro vértice, mas conectam o par de vértice de forma igual.

Com a afirmação anterior, sabemos com certeza que não possuiremos arestas paralelas, e com a função find_loops, podemos confirmar que também não encontramos laços, ou seja, se trata de um Grafo Simples.

Sim, possui o atributo conexões, onde nos é informado, além do número do vértice, quantas conexões o mesmo possui, ou seja, seu grau.

O grafo possui exatamente 7393 vértices, que representam as proteínas e exatos 25569 vértices que simbolizam as Ligações Protéicas. E por meio da função avgDegree, o mesmo retorna aproximadamente 3.5!

Limitação do escopo e definição do problema:

O intuito do projeto é retornar informações principais valiosas para a analise de importância, e retornos secundários que agregam valor de conhecimento sobre mais informações sobre o Grafo.
O principal objetivo se baseia em achar quais as 10 principais proteínas presentes no metabolismo humano que possuem maior importância, pois constituem as mais diversas ligações, ou seja, o funcionamento metabólico proteico depende principalmente dessas proteínas. Outro fator principal é a seleção de todas as proteínas criticas, que sustentam conjuntos de ligações que dependem entre si para o funcionamento do corpo, e que São sustentadas por apenas uma parcela de proteínas, por isso chamadas de críticas. Com isso, o intuito se encontra em saber quais as principais proteínas, pois informações como essas podem indicar a profissionais da saúde quais as principais necessidades de reposição dessas proteínas ao indicar medicamentos e alimentos no tratamento de pacientes, pois obviamente proteínas criticas exigem maior atenção e cuidado para um bom funcionamento do metabolismo e da vida como um todo.

Exemplo:

3 1

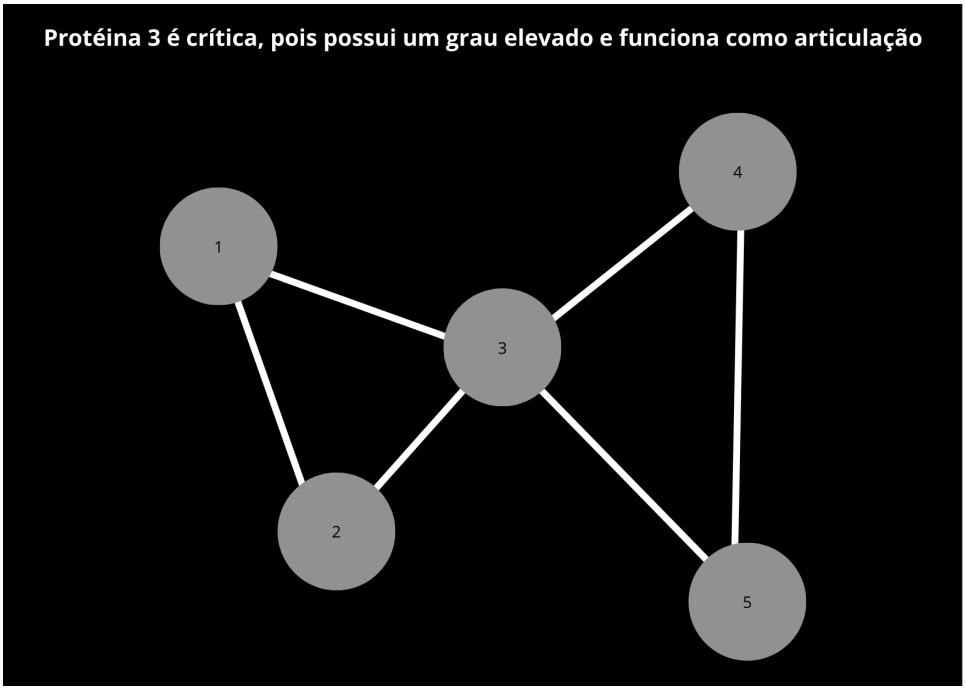
3 2

3 3

3 4

1 2

4 5



Nesse caso em específico, conseguimos ver que, todas as proteínas com maior grau, estarão sempre entre as articulações, ou seja, sua importância é crítica, no sentido de serem necessárias para a realização de diversas ligações proteicas e sustentarem ligações diversas com outras conjuntos dentro do grafo completo.

Sua solução usando algoritmos:

Inicialmente, por se tratar de um arquivo .mtx, precisei a partir do próprio código Ignorar as duas primeiras linhas do cabeçalho, pois A primeira linha indica que é um arquivo no formato MatrixMarket, especificando que a matriz está em formato de coordenadas, contendo um padrão de matriz simétrica. A segunda linha mostra as dimensões da matriz: 7393 linhas, 7393 colunas e um total de 25569 elementos não nulos. As linhas subsequentes denotam as coordenadas dos elementos não nulos na matriz. Por exemplo, as linhas "179 1", "472 1" e "953 1" indicam que existem elementos não nulos nas coordenadas (179, 1), (472, 1) e (953, 1) na matriz. Além disso, no início do código, é fornecido uma referencia de identificação sobre a base da dados que utilizei, coletada em <https://networkrepository.com/>.

Sobre os algoritmos utilizados, foram alguns fundamentais para as informações que obtivemos nesse projeto:

- Verificação Direta: A ideia aqui é que, como nosso grafo foi armazenado em uma lista de adjacência, conseguimos saber, de cada vertice, qual é o seu vizinho. Logo, é só fazer uma busca dentro dos vizinho de um vértice X, em busca de uma comparação na qual o vizinho seja
= X, assim, se retornar True, temos um SelfLoop, ou laço. Outra ideia semelhante é analisar quais vertices na lista possuíam os maiores números de vizinhos, sabendo assim quais deles eram os mais críticos/importantes.

- Deep First Search: O famoso DFS foi usado no código com o intuito de detectar a existencia de ciclos. A ideia principal por trás do código é que a partir de um vértice agora cinza, consigamos encontrar um vizinho cinza (aresta de retorno), mostrando a existência de um ciclo.

- Tarjan: Esse famoso algoritmos foi utilizado com o fim de achar quais vértices funcionam como articulações. A ideia por aqui é que se, durante a DFS, já que tarja se utiliza da estrutura de um DFS, um vértice A possui um vizinho B tal que o valor mínimo de descoberta de B é maior ou igual ao tempo de descoberta de A, então o vértice A é uma articulação. Isso significa que a remoção de A aumentaria o número de componentes conectados no grafo, indicando sua importância na conexão do grafo como um todo.

Ferramenta/software desenvolvido:

A linguagem utilizada nesse projeto foi o python, por ter várias bibliotecas que auxiliaram na produção, e por sua facilidade de uso e entendimento. As bibliotecas foram: networks, para lidar com o grafo em sí, matplotlib para a visualização propriamente gráfica do Grafo, e finalmente a collections, para a criação do grafo. Assim, a ideia é criar o grafo, obter informações cruciais do mesmo, e finalmente fazer a renderização do grafo na tela.

Para utilizar, basta utilizar do makefile, no caso, make run! **********

Ao executar o projeto, voce será capaz principalmente de saber quais as 10 proteínas críticas do seu conjunto metabólico de amostra, e uma lista de proteínas a mais que são essenciais na ligação de diversos componentes/grupos protéicos.

Resultados:

No caso da base de dados que utilizei, as top 10 proteínas foram: 215, 663, 708, 209, 76, 299, 61, 1411, 331, 204. Essas então seriam as 10 proteínas mais importantes para o funcionamento metabólico do corpo humano, sendo necessário uma maior atenção na reposição e produção das mesmas, e tratamentos e estudos seriam mais necessários voltados a essas. É interessante também saber que todas essas, também são articulações, intensificando sua importância, ou seja, participam como proteínas que são essenciais para a comunicação entre diferentes partes de uma célula ou entre células, coordenando funções biológicas complexas.

Limitações:

Os principais pontos de destaque nesse tópico são a quantidade de proteínas críticas, uma vez que só mostra as 10 proteínas/vertices que tem maior impacto, esse número poderia facilmente ser alterado, porém, parta fins de agilidade no ler das informações, creio que 10 é um numero aceitável. E o outro, é que utilizei apenas de duas ideias principais para descobrir as proteínas mais importantes, que é a Grau de importância e a verificação de articulações, poderia se expandir para mapeamento de padrões, onde encontraria cadeias proteicas mais recorrentes, mas não foi o caso.

* Logo em seguida, temos dois .png que são referentes a imagem do Grafo em tempo real, juntamente com os top 10 vértices mais importantes: