

# TRABALHO PRÁTICO I

## Algoritmos Avançados de Bioinformática

Neste trabalho prático é esperado que apliques os conhecimentos sobre grafos adquirido nas aulas de Algoritmos Avançados de Bioinformática. Para tal iremos analisar um modelo metabólico do organismo *E.coli*. Faz *download* dos ficheiros presentes na pasta partilhada “TrabalhoPrático” e que representam os metabolitos, reações e matriz estequiométrica presentes no modelo.

Descrição dos ficheiros:

- **metabolites.csv** – cada linha tem o identificador de um composto presente no modelo
- **reactions.csv** – cada linha contém 3 campos (identificador da reação, *lower bound*, *upper bound*) os *bounds* representam o intervalo de valor de fluxo que a reação pode ter. Valores negativos no *lower bound* indicam que a reação pode ocorrer no sentido inverso ao presente no modelo. Isto é, se a reação  $r1 = A \rightarrow B$  tiver *lower bound* negativo a reação pode ser de  $B \rightarrow A$ .
- **matrix.csv** – cada linha é composta por 3 campos. O primeiro indica o índice do metabolito, o segundo o índice da reação e o terceiro o coeficiente. Se o coeficiente for positivo significa que o composto é produzido, se for negativo significa que é consumido pela reação. Considerando o exemplo:  $r1 = A \rightarrow B$  sendo a lista de metabolitos  $[A, B]$  e lista de reações  $[r1]$  o ficheiro **matrix.csv** terá o seguinte conteúdo

0,0,-1

1,0,1

Com base no código desenvolvido nas aulas cria uma script python que permite responder às seguintes questões (para cada alínea cria uma função com a assinatura especificada):

1. Cria um grafo com base na rede metabólica presente nos 3 ficheiros.

*MyGraph* *perg\_1* (*file\_meta*, *file\_reac*, *file\_mat*)

2. Qual o número de reações e metabolitos.

(*num\_reac*, *num\_meta*) *perg\_2* (*graph*)

3. Identifica os 10 metabolitos que participam num maior número de reações.

*list perg\_3 (graph)*

4. Cria um gráfico com a distribuição do número de reações por cada um dos possíveis graus do tipo “inout”.

*perg\_4 (graph)*

5. Identifica os metabolitos que são *dead ends*, isto é, metabolitos que são produzidos e não há reações que os consumam.

*list perg\_5 (graph)*

6. Considerando uma lista de metabolitos como *uptake* dado pelo utilizador, implementa uma função que retorna quais os produtos excretados pelo modelo.

*list perg\_6 ( graph, uptake)*

7. Cria uma função que dado um metabolito dado pelo utilizador devolve a lista de reações que produzem esse metabolito.

*list perg\_7 (graph, meta)*

8. Valida que no modelo não existem reações que contenham o mesmo metabolito como reagente e produto, se existir devolve uma lista contendo essas reações, caso contrário devolve None.

*list perg\_8 (graph)*

9. Implementa uma função que dado o metabolito origem e metabolito destino retorne todos os caminhos possíveis entre estes dois nós, onde o caminho é composto pela sequência de nós que permite da origem chegar ao destino.

*list perg\_9 (graph, orig, dest)*

10. Implementa uma função que retorne a lista de ciclos existentes na rede. Clico: é a sequência de nodos que dado um ponto inicial consegue voltar ao mesmo ponto.

*list perg\_10 (graph)*