

1. Introdução: Este código implementa o algoritmo Neighbor Joining (NJ) para construção de árvores filogenéticas. Ele gera uma estrutura de árvore para representar as relações evolutivas entre diferentes espécies com base em uma matriz de distâncias genéticas.
2. Classes e Estruturas de Dados:
 - a. `TreeNode`: Classe para representar um nó da árvore filogenética.
 - b. `label`: Rótulo do nó.
 - c. `children`: Lista de nós filhos.
 - d. `distance_to_parent`: Distância até o nó pai.
3. Funções Principais:
 - a. `calculateQ(d)`: Calcula a matriz Q usando a fórmula do algoritmo NJ.
 - b. `findLowestPair(q)`: Encontra o par de espécies com a menor distância na matriz Q.
 - c. `doDistOfPairMembersToNewNode(i, j, d)`: Calcula as distâncias para o novo nó criado.
 - d. `calculateNewDistanceMatrix(f, g, d)`: Calcula a nova matriz de distâncias após unir dois nós.
 - e. `build_tree_structure(d, labels)`: Constrói a estrutura da árvore filogenética com base nas distâncias e rótulos das espécies.
 - f. `print_tree(node, indent=0)`: Imprime a estrutura da árvore recursivamente.
 - g. `doNeighbourJoining(d, labels)`: Função principal que chama as outras para construir e imprimir a árvore.
4. Execução Principal:
 - a. A matriz de distâncias (`distMatrix`) e os rótulos das espécies (`labels`) são fornecidos como entrada para a função `doNeighbourJoining`.
5. Saída:
 - a. A estrutura final da árvore filogenética é impressa no console.

Warning: Division by zero avoided.

Final tree structure:

AB : 0

AB : 0.038500000000000006

A : 0.0

C : 0.0

AB : 0.006449999999999997

D : 0.0

BE : 0.09316666666666665

B : 0.0

E : 0.0

O código utiliza a biblioteca NumPy para manipulação eficiente de matrizes.
As distâncias são representadas como rótulos nos nós da árvore, indicando as relações evolutivas entre as espécies.