

**Universidade Federal do Rio Grande do Sul - Instituto de Informática**  
**INF05018 Biologia Computacional - Prof: Márcio Dorn**

**Rodrigo Okido (252745)**

**Resolução Lista IV**

Será apresentado aqui a resolução da lista IV. Todos os métodos serão explicados através de pseudocódigos informando a idéia geral das funções implementadas. Caso dúvidas referente a explicação, o código se encontra disponível bastante documentado para quaisquer esclarecimentos.

**Variáveis Globais:**

dist\_species = { }

Dicionário para armazenar a distância entre as duas espécies que serão comparadas.

**Function update\_matrix (matrix, a ,b) :** Faz a atualização da matriz no momento em que encontra o menor valor. Caso haja problemas em ordem dos índices, eles são ordenados no primeiro IF. Em seguida, são realizados 3 loops para atualizar a matriz corretamente. No primeiro, a linha “a” onde  $i < a$  são atualizados. No segundo, toda a coluna onde  $i > a$  é atualizada. E por fim na terceira, os valores restantes da linha  $i$  é atualizada. Ao final, suas redundâncias são atualizadas. As exclusões ao final são feitas devido ao fato de que a tabela diminui a cada iteração, tendo junções das espécies de menor distância realizadas.

Pseudocode:

```
linha = [ ]

#Reorders if necessary
Se b < a
    a = b
    b = a

#Start the update on the matrix
para i in range(0, a):
    row.append((matrix[a][i] + matrix[b][i])/2)

matrix[a] = row

para i in range(a+1, b):
    matrix[i][a] = (matrix[i][a] + matrix[b][i])/2
```

```

Para i in range(b+1, len(matrix)):
    matrix[i][a] = (matrix[i][a] + matrix[i][b])/2
    delete matrix[i][b]

delete matrix[b]

#Update labels
labels[a] = "[ labels [a] '-' labels[b] ]"
delete labels[b]

```

**Function find\_lowest\_distance (matrix, labels) :** Função para encontrar a menor distância atual na matriz. Salva a coordenada do menor número nas variáveis x e y e realiza o cálculo da distância entre as duas espécies escolhidas.

Pseudocode:

```

menor_distancia = 1000000000
X = 0
Y = 0

Para i, j em MATRIZ:
    Se matrix[ i ][ j ] < menor_distancia
        Menor_distancia = matrix [ i ][ j ]
        X = i
        Y = j

calcula_distancia = menor_distancia / 2
adiciona resultado de calcula_distancia num dicionário de distância (dist_species)

```

**Function upgma\_method (matrix, assoc\_label) :** Realiza o método UPGMA. Faz o chamamento dos métodos implementados e comentados anteriormente, e é executado enquanto houver elementos na matriz.

Pseudocode:

```

Enquanto tamanho (assoc_label) maior que 1:

    x,y = encontra_menor_distancia (matrix, assoc_label)
    update_matrix(matrix, x,y)

```

Retorna assoc\_label[0]

**Function main ( )** : Função principal de execução. Gerado uma lista de associação (hard coded) e a matriz que será utilizada para ser aplicado método UPGMA. Ao final os resultados serão impressos na tela.

Pseudocode:

```
assoc_label = ["GOR", "ORANG", "HUM", "CHI", "GIB"]

upgma_matrix = [
    [],
    [0.1890],
    [0.1100, 0.1790],
    [0.1130, 0.1920, 0.09405],
    [0.2150, 0.2110, 0.2050, 0.2140]
]

imprime upgma_method(upgma_matrix, assoc_label)
imprime "\nDistance between especies:\n", dist_species
```

**O resultado produzido será :**

[[[GOR - [HUM - CHI]] - ORANG] - GIB]

Distance between especies:

{'GOR - [HUM - CHI]': 0.05575, '[[GOR - [HUM - CHI]] - ORANG] - GIB': 0.1058125, 'HUM - CHI': 0.047025, '[GOR - [HUM - CHI]] - ORANG': 0.093625}

Na primeira, a resposta informa que Humanos têm uma relação próxima com Chimpanzés, e estes com Gorilas. Com isto, o próximo a entrar na ligação a estes três é o Orangotango, e a ligação é finalizada com o Gibão.

Na segunda, mostra o valor da diferença entre as espécies, ou seja, em quantas unidades seria necessário percorrer para uma espécie “virar” outra.