

TRABALHO FINAL DE ALGORITMOS - (SÍMIOS).

INSTITUTO FEDERAL CATARINENSE - CAMPUS VIDEIRA

CIÊNCIAS DA COMPUTAÇÃO

GABRIEL DE SOUSA ARAUJO

Relatório do programa que detecta uma sequência de DNA.

A primeira coisa a se fazer foi dividir o código em 8 módulos a fim de facilitar a criação do programa e deixá-lo mais limpo, sendo estes 8 módulos os seguintes:

Obs: 8 módulos com maior utilidade para o funcionamento do programa.

Um módulo para alocar matriz de acordo com o tamanho que o usuário informou. para isso se foi usado a função “calloc”, assim alocando dinamicamente a matriz.

Outro módulo para carregar a matriz com as bases nitrogenadas (A, T, C, G), informadas também pelo usuário, para isso foi utilizado duas estruturas de repetição, uma para linha e outra para a coluna e também foi utilizado da função “tolower” disponibilizada pela biblioteca “ctype.h”, essa função faz com que os caracteres digitados fiquem todos minúsculos.

Agora vem as comparações para encontrar uma sequência de 4 bases nitrogenadas iguais, começando pela linha, que foi necessário 2 estruturas de repetição uma para a linha que começa em 0 e vai até o tamanho informado pelo usuário e outra para a coluna que inicia-se também em 0 porém percorre até o tamanho - 1, caso a base atual fosse equivalente a próxima base na mesma linha, é armazenado em uma variável chamada “igual” o valor de +1, caso não seja equivalente é definido o valor de “igual” igual a 1, em caso de a variável chegue a 4 é retornado o valor 1 alertando que aquele DNA é pertencente a um símio, já no caso de que não haja 4 sequências nas linhas o módulo retorna 0.

Para constatar se havia sequências nas colunas o método foi muito parecido com o método da linha, invertendo apenas as duas estruturas de repetição, onde a coluna começa em 0 e vai até “tamanho” e a linha começa em 0 e vai até “tamanho” - 1.

Na diagonal principal foi mais simples, ao olharmos para uma matriz percebemos que as posições da linha e da coluna que cruzam a diagonal principal são sempre iguais, por isso dessa vez foi preciso apenas uma estrutura de repetição, já na parte de comparação usamos apenas uma variável como por exemplo:

```
mat[i][i] == mat[i+1][i+1].
```

Após olhar a diagonal principal se é preciso olhar as diagonais acima e abaixo dela também e para isso se foi observado que para a diagonal acima é preciso que a coluna seja sempre maior que a linha e para a diagonal abaixo é o contrário a linha precisa ser maior que a coluna. Sabendo-se disso foi preciso de 2 estruturas de repetição, uma para “controlar” a matriz inteira e uma para a linha/coluna (depende se está fazendo a acima ou a abaixo da diagonal principal, como explicado anteriormente) e também foi preciso de um contador simples, com tudo isso foi preciso apenas inicializar a estrutura da “matriz” com 0 indo até “tamanho” linha/coluna igual a 0 e a outra estrutura linha/coluna iniciando com 1 + “matriz” e caso diagonal acima vai até “tamanho”, entretanto se diagonal abaixo vai até “tamanho” - 1 e sempre que essa estrutura rodar é preciso incrementar +1 em linha/coluna.

Para se olhar a diagonal secundária usamos do mesmo método anterior, onde irá olhar a diagonal secundária, depois o que tem acima dela e por último o que tem abaixo da diagonal secundária, para isso também utilizamos duas estruturas de repetição e

um contador simples, porém o contador simples será inicializado com “tamanho” - “matriz” e ao invés de acontecer um incremento acontece uma redução em linha/coluna, isso se dá pelo motivo de que para analisar essas diagonais se é preciso começar pela última posição da coluna.

Após todas comparações de sequências, foi preciso criar um módulo para receber esses outro módulos e com a utilização do “if” verificamos se algum módulo retornou 1 sinalizando que o DNA era de símio, caso isso não ocorra este módulo retorna 0 para a função “main” sinalizando ser o DNA de um humano.