

Introdução à Bioinformática - T1
Vários Cursos
– FACOM/UFMS –
Primeira Trabalho da disciplina¹
– Alinhamento de Sequências –

O principal objetivo deste trabalho é a implementação dos algoritmos vistos em sala para o problema do alinhamento de sequências (global, local e semi-global).

O seu programa deve receber duas sequências u e v construídas sobre um alfabeto qualquer Σ (tal que $'-' \notin \Sigma$) e uma função de pontuação qualquer $w : \bar{\Sigma} \times \bar{\Sigma}$ que atribui um valor inteiro para cada par de caracteres de $\Sigma \cup \{'-\'}$ e devolver um alinhamento ótimo (global, local ou semi-global) de u e v assim como o valor da similaridade entre elas. As sequências serão fornecidas em arquivo, enquanto que a função de pontuação será fornecida via terminal. A saída do seu programa deverá ser impressa no terminal.

As sequências de entrada serão fornecidas no formato FASTA. Nesse formato, a primeira linha do arquivo inclui a identificação da sequência e outras informações relacionadas a ela, tudo precedido do sinal '>'. As linhas seguintes, de tamanho 70, correspondem aos caracteres que constituem a sequência. Abaixo temos um exemplo de sequência no formato FASTA cuja identificação é AB010874.

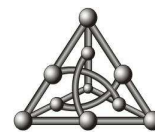
```
>AB010874 Homo sapiens gene da proteina ribossomal L41
TTCGCCTTTCTCTCGGCCTTAGCGCCATTTTTTTGGGTGAGTGTTTTTTGGTTCCTGCGTTGGGATTCCG
TGTACAATCCATAGACATCTGACCTCGGCACTTAGCATCATCACAGCAAATACTGTAGCCTTTCTCTC
TTTCCCTGTATAAACCTCTGCGCCATGAGAGCCAAGGTGAGCGGTCCTGGTAGTAAGCTTGGGAGGTAG
GAGTTGGCGAGTAGTAGCGGGGAGACGAAGGCAAGTCCGCCATACCTCCTGAACTACTGGGTTTCAAGGG
TGCCCAAGAGCTGTTGGGAGAGAGAAGGTAGTTTGTGAGAGAGCTAGCGGTTAAGTGCTATGGGTAGAGA
```

Figura 1: Exemplo de sequência no formato FASTA

A saída do seu programa deve conter o nome de cada uma das sequências e o alinhamento propriamente dito, com 30 caracteres por linha. Explícite as colunas do alinhamento com caracteres iguais e diferentes com os símbolos '=' e '!', respectivamente. As colunas com espaços devem ser explicitadas com o símbolo '-' na sequência onde espaços foram inseridos. A similaridade das sequências deve ser impressa na linha seguinte à última linha do alinhamento, no formato `similaridade:<valor da similaridade>`, onde o valor da similaridade é um inteiro.

A seguir pode ser visto um exemplo de saída, no formato especificado, para duas sequências quaisquer ACCAGTACACCAGATCACAGATAATAGAGACACAGATAACACAGAATAT e GGGAC TAGTAGATACCAGTA,

¹Versão 1 - Este documento pode sofrer modificações de acordo com discussões que tivermos em sala de aula ou no fórum de discussão.



denominadas seq1 e seq2 respectivamente, e uma função de pontuação tal que $w(a, a) = 1$, $w(a, b) = -1$ e $w(-, b) = w(a, -) = -2$.

```
seq1 ACCAGTACACCAGATC-----ACAGATAAT
      !  !!!  !!   |!!           !!  |!
seq2 G--GGA--CT---AGTAGATA---CC--AG
seq1 AGAGACACAGATAACACAGAATAT
      !!
seq2 TA-----
similaridade: -87
```

Figura 2: Exemplo de saída do programa

Você pode assumir que as sequências de entrada terão no máximo 1000 caracteres.

A implementação dos três algoritmos deve ser feita em um único programa. Esse programa deve ser escrito em C ou em C++, e permitir sua execução da seguinte forma:

```
<nome executável> [-g,-l,-s] -u <arquivo 1a. sequencia> -v <arquivo
2a. sequência> -i <w(a,a)> -d <w(a,b)> -e <w(-,b) ou w(a,-)>
```

Nas duas linhas acima, -g, -l e -s correspondem ao tipo de alinhamento desejado (global, local e semi-global, respectivamente). Já sobre os parâmetros -i, -d e -e, eles correspondem aos valores atribuídos pela função de pontuação para colunas com caracteres iguais, diferentes e espaço, respectivamente.

Deem uma olhada em como utilizar os argumentos argc e argv da main para permitir a execução do programa conforme descrito acima.

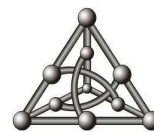
1 Entrega do trabalho

O código fonte do seu programa deve ser entregue até as **23h:55min do dia 06 de novembro de 2014** diretamente no Sistema de Suporte a Disciplinas da FACOM (ead.facom.ufms.br), na seção referente à disciplina Introdução à Bioinformática. Você pode submeter seu arquivo quantas vezes quiser, **observando que a última submissão é a que será considerada.** Vale salientar também que **nenhum trabalho será recebido fora do prazo.**

2 Critérios de correção

O principal critério de correção a ser utilizado pelo professor é a correção do programa. Ou seja, o número de casos de testes que ele resolve corretamente,

Sobre o código-fonte, ele também será corrigido com base nos seguintes critérios:



-
1. erros de compilação: programas com erros de compilação receberão nota 0 (zero).
 2. *warnings*: programas que apresentarem *warnings* ao serem compilados serão penalizados (por cada *warning* encontrado);
 3. clareza e organização: programas com código confuso (linhas longas, variáveis com nomes não-significativos, etc.) e desorganizado (sem indentação, sem comentários, etc.) também serão penalizados;
 4. eficiência: programas muito ineficientes também serão penalizados.

Durante a correção, os programas serão compilados com as opções `-Wall -ansi -pedantic` do gcc ou g++.

3 Conduta ética

O trabalho pode ser feito em dupla de dois ;). Cada grupo deve fazer o seu próprio trabalho. Não repasse para e nem copie o programa de outro grupo. Trabalhos considerados plagiados receberão nota 0 (zero).