

---

# Busca Exaustiva para Alinhamento de sequências

26 de abril de 2022

## VISÃO GERAL

O projeto apresentado é uma forma de obter uma solução aproximada de um problema de forma mais rápida, tendo em vista que os métodos clássicos para resolver problemas de alinhamento de sequências de dna são muito lentos.

## OBJETIVOS

1. Implementar um programa C++ para ler um arquivo contendo os tamanhos de duas sequências de DNA, seguidos das duas sequências, uma por linha. Calcular o score máximo utilizando o algoritmo acima, assim como as subsequências associadas a ele.
2. Implementar duas estratégias diferentes para calcular os alinhamentos entre os pares de subsequências do passo (2). No diretório do projeto, há um gerador de entradas disponibilizado como um notebook Python. Como se trata de uma busca exaustiva, recomenda-se começar a testar com tamanhos pequenos e ir aumentando gradativamente até atingir o tamanho máximo que a sua plataforma ainda consiga executar.

## MARCOS

### Compilar

Para compilar o programa foi utilizando o comando, mostrado a seguir:

```
g++ -Wall -O3 main.cpp -o main
```

### Rodar

---

Após compilar o código, o seguinte comando é utilizado para rodar o código:

```
./main <in.txt >out.txt
```

Sendo *in.txt* a entrada do programa e *out.txt* a saída

## Testes

### Teste1

Utilizando as sequências fornecidas na descrição do projeto, sendo elas:

AGCACACA

ACACACTA

A Partir do código “exaustiva” o qual utiliza o alinhamento local de smith-waterman, foi obtido o seguinte resultado:

```
score maximo: 12  
  
sa: AGCACACA  
  
sb: ACACACTA
```

Os arquivos de entrada e saída desse teste são respectivamente: *in.txt*, *out.txt*.

Já pelo código “exaustiva 2” o qual utiliza o alinhamento local de smith-waterman somente para sequências com tamanho diferentes, e a do projeto 2 quando o tamanho for igual. Foi obtido o seguinte resultado:

```
score maximo: 11  
  
sa: AGCACAC  
  
sb: ACACAC
```

Os arquivos de entrada e saída desse teste são respectivamente: *in.txt*, *out\_2.txt*.

### Teste2

---

Utilizando as sequências:

-A-GCGG-C-

AT-T--ATCT-T-TTCTAGT-TGT--TAGA---C--TT-G

O maior score obtido pelo código “exaustiva” foi:

```
score maximo: 7  
  
sa: -A-GCGG-C-  
  
sb: AT-T--ATCT-T-TTCTAGT-TGT--TAGA---C-
```

Os arquivos de entrada e saída desse teste são respectivamente: *in1.txt*, *out1.txt*.

O maior score obtido pelo código “exaustiva2” foi:

```
score maximo: 7  
  
sa: -A-GCGG-C-  
  
sb: AT-T--ATCT-T-TTCTAGT-TGT--TAGA---C-
```

Os arquivos de entrada e saída desse teste são respectivamente: *in1.txt*, *out1\_2.txt*.

Sendo assim, os resultados obtidos iguais.

## Teste3

Utilizando as sequências:

AC-TACGCGC

-CGACATACCCTC-TTCGTGGCCA-TA-C-TAGT-ATAAC

O maior score obtido pelo código “exaustiva” foi:

```
score maximo: 11  
  
sa: AC-TACGCGC  
  
sb: -CGACATACCC
```

Os arquivos de entrada e saída desse teste são respectivamente: *in2.txt*, *out2.txt*.

---

O maior score obtido pelo código “exaustiva2” foi:

```
score maximo: 11  
  
sa: AC-TACGCGC  
  
sb: -CGACATACCC
```

Os arquivos de entrada e saída desse teste são respectivamente: *in2.txt*, *out2\_2.txt*.

Sendo assim, os resultados obtidos iguais.