

Relatório busca exaustiva - Supercomputação

O código foi baseado no pseudocódigo dos professores. Foi utilizada a linguagem de programação C++ para realização. O código em si consiste de uma busca exaustiva para cálculo da pontuação de subsequências de dois sequenciamentos genéticos. Foram implementadas duas estratégias para o passo dois.

O código está todo comentado, é possível ler os comentários e o código e entendê-lo. Mas aqui vai uma breve explicação sobre o código truncando as subsequências para calcular a pontuação. Primeiramente lemos as duas sequências e seus tamanhos e guardamos em dois vetores e em duas variáveis, respectivamente. Depois criamos as duas matrizes que irão conter as subsequências de *a* e *b*. Recursivamente populamos essas matrizes com todas as subsequências possíveis obedecendo a ordem, ou seja, são *substrings*. Depois criamos a matriz de pontuações que irá conter todas as pontuações das comparações entre as subsequências de *a* com as subsequências de *b*. Em seguida, pulamos essa matriz com os resultados das comparações entre as subsequências, de acordo com o que para match (dois), *mismatch* (menos um) e *gap* (menos um), truncando as subsequências levando em consideração a menor subsequência. Após isso fazemos a busca pela maior pontuação e as subsequências relacionadas e imprimimos.

O segundo código é bem parecido com o primeiro com a diferença que implementa *branch and bound* e fica mais rápido. Para isso, toda vez que entramos no *loop* de calcular as pontuações calculamos o máximo que a subsequência pode gerar de pontos calculando duas vezes o número de elementos que faltam somado à pontuação atual. Caso esse valor for maior que o maior valor de pontuação continuamos o *loop* caso não paramos o *loop*.

Para a primeira entrada `in1_busca_exaustiva.txt` (que tem dez elementos em cada sequência) o primeiro programa (sem *branch and bound*) encontrou uma pontuação máxima de sete em 0.05 segundos e as duas subsequências abaixo:

T--TG-GTT
T-TTG

Já o segundo programa (com *branch and bound*) chegou a pontuação de cinco em 0.01 segundos com as seguintes subsequências:

T--TG-GT
T-TTGCC

Para a segunda entrada `in2_busca_exaustiva.txt` (que tem vinte elementos em cada sequência) o primeiro programa achou uma pontuação máxima de onze e as seguintes subsequências, em 0.65 segundos:

GCCC---C-GAGC

GGCG--GC-G

Já o segundo programa chegou a pontuação de dois, em 0.30 segundos com as seguintes subsequências:

-GAGGGCCC---C-

-GGCG--GC-GC-

Para a terceira entrada in3_busca_exaustiva.txt (que tem trinta elementos em cada sequência) o primeiro programa chegou a uma pontuação de quatorze em 199.62 segundos com as seguintes subsequências:

TTACCGGCTACCCAAAC-CTAG-GGT-GAA

TCCCCGGCTA

Já o segundo programa chegou a uma pontuação de dois em 197.41 segundos com as seguintes subsequências:

CTACCCAAAC-CTAG-GGT-GAA

-G--TCA-AC-CT-TGTAA-TC

Para a quarta entrada in4_busca_exaustiva.txt (que tem vinte elementos na primeira sequência e trinta elementos na segunda sequência) o primeiro programa encontrou uma pontuação de seis em 100.36 segundos com as seguintes subsequências:

T-CATATGCGACCTA

ACCAT-TGC-CCAAT

Já o segundo programa também encontrou uma pontuação de seis mas em 99.47 segundos e obteve as seguintes subsequências, iguais ao primeiro programa:

T-CATATGCGACCTA

ACCAT-TGC-CCAAT

Para n e m maiores que 30 o código demora muito para executar.