



Trabalho 2 de Biologia Computacional

Lista 2

Vilmar Dorneles Aprato Neto
22/08/2018

Para a abordagem dos cinco problemas da primeira lista foi utilizada a linguagem de programação Python 3.5, foram utilizados dois sistemas operacionais de teste, Linux Mint 16 e Windows 10. Os teste foram feitos em uma máquina Quad-Core com processador Intel I5-4590K e 8GB de Memória RAM.

[illegible]

Os resultados mostraram que tanto vaca quanto cavalo tem a mesma similaridade, com 87% com a hemoglobina humana. Sendo assim os dois os mais similares.

Pseudo-código

Foi utilizado como base o pseudo-código disponibilizado do algoritmo de Needleman Wunsch no wikipedia : https://pt.wikipedia.org/wiki/Algoritmo_Needleman-Wunsch

Inicialização*MatrixScore(sequence1,sequence2):*

m, n = tamanho de sequence 1 e sequence2

Inicia-se uma matriz com valores zeros de tamanho $m+1$ e $n+1$: score

Para i de 0 até $m+1$:

score[i][0] <- gap * i

Para j de 0 até $n+1$:

score[0][j] <- gap * j

Para i de 0 até $m+1$:

Para j de 0 até $n+1$:

Choice1 \leftarrow score($i-1, j-1$) + S(A($i-1$), B($j-1$)))

Choice2 \leftarrow score($i-1, j$) + d

Choice3 \leftarrow score($i, j-1$) + d

score(i, j) \leftarrow max(Choice1, Choice2, Choice3)

retorna score

S(char1,char2): # funcao referente ao score entre dois caracteres

se char1 == char2:

retorna match

se char1 == "-" ou char2 == "-":

retorna gap

senão:

retorna mismatch

Traceback(A,B,F) # onde F é a matrix de Score

```

AlignmentA ← ""
AlignmentB ← ""
i ← length(A)
j ← length(B)
while (i > 0 AND j > 0)
{
  Score ← F(i,j)
  ScoreDiag ← F(i - 1, j - 1)
  ScoreUp ← F(i, j - 1)
  ScoreLeft ← F(i - 1, j)
  if (Score == ScoreDiag + S(A(i-1), B(j-1)))
  {
    AlignmentA ← A(i-1) + AlignmentA
    AlignmentB ← B(j-1) + AlignmentB
    i ← i - 1
    j ← j - 1
  }
  else if (Score == ScoreLeft + d)
  {
    AlignmentA ← A(i-1) + AlignmentA
    AlignmentB ← "-" + AlignmentB
    i ← i - 1
  }
  otherwise (Score == ScoreUp + d)
  {
    AlignmentA ← "-" + AlignmentA
    AlignmentB ← B(j-1) + AlignmentB
    j ← j - 1
  }
}
while (i > 0)
{
  AlignmentA ← A(i-1) + AlignmentA

```

```
    AlignmentB ← "-" + AlignmentB
    i ← i - 1
}
while (j > 0)
{
    AlignmentA ← "-" + AlignmentA
    AlignmentB ← B(j-1) + AlignmentB
    j ← j - 1
}
```

Referências:

https://pt.wikipedia.org/wiki/Algoritmo_Needleman-Wunsch