Comparação de Sequências Múltiplas

MQPILLL
MLR–LL–
MK–ILLL
MPPVLIL

Sejam $s_1..s_k$ sequências sobre o mesmo alfabeto. Um alinhamento múltiplo envolvendo $s_1..s_k$ é obtido inserindo espaços nas sequências tal que todas fiquem com o mesmo tamanho. Nenhuma coluna pode conter apenas brancos.

Alinhamentos múltiplos são mais frequentes envol vendo proteínas (sequências de aminoácidos).

A Medida SP(Sum of Pairs)

É uma medida para definir o **escore** de um alinhamento múltiplo.

Associa um valor a cada coluna, e o escore total é a soma dos valores das colunas do alinhamento.

Suponha um alinhamento de k sequências. Para encontrar o escore deste alinhamento, precisa-se de uma função que, dados os k valores de uma mesma posição nas diferentes sequências, devolva o valor do alinhamento (cada posição pode conter um caracter ou um espaço).

Poderia-se utilizar uma matriz de k-dimensões, mas...

A solução usual é definir funções de medida. Essas funções tem as seguintes características (desejadas):

1. A ordem dos argumentos não influencia o resultado;

2. A presença de vários caracteres iguais deve ser recompensada, e a de vários caracteres diferentes ou brancos deve ser penalizada.

Exemplo: Valores de uma coluna: I, -, I, V

$$SPscore(I,-,I,V) = p(I,-) + p(I,I) + p(I,V) +$$

$$p(-,I) + p(-,V) + p(I,V)$$

Se considerarmos $p(-,-) = 0$ temos

$$SPscore(\alpha) = \sum_{i<j} score(\alpha_{ij})$$

PEAALYGRFT---IKSDVW
PEAALYGRFT---IKSDVW
PESLAYNKF---SIKSDVW
PEALNYGRY---SSESDVW
PEALNYGWY---SSESDVW
PEVIRMQDDNPFSFQSDVY

PEAALYGRFT---IKSDVW
PEALNYGRY---SSESDVW

PEAALYGRFT-IKSDVW
PEALNYGRY-SSESDVW

Alinhamento Ótimo

É possível usar programação dinâmica, como no caso de duas sequências:

- usa-se um **array de k-dimensões** com tamanho $n+1$ em cada dimensão (supondo que todas sequências tem tamanho n);

- $a[i_1, \ldots, i_k]$ contém o escore do alinhamento ótimo envolvendo $s_1[1..i_1], \ldots, s_k[1..i_k]$.

Análise da complexidade

- Depois de inicializar a matriz $a[0, \ldots, 0] \leftarrow 0$, deve-se preencher a matriz: espaço ocupado $O(n^k) \Longrightarrow$ este também é um limite mínimo para o tempo...

- Cada entrada depende de $2^k - 1$ outras entradas previamente computadas, uma para cada composição da coluna corrente da matriz (com k sequências tem-se 2^k composições, mas remove-se a composição proibida somente de espaços);

- Acesso aos dados da matriz: $O(k)$ passos

- Computação do escore de uma coluna: usando o método SP temos $O(k^2)$ pois existem $\frac{k \times (k-1)}{2}$ pares de escores a considerar.

- Computação do valor da célula: envolve uma operação de máximo:

$$a[i] \leftarrow max_{b \neq 0}\{a[i-b] + SPscore(Column(s,i,b))\}$$

onde b varia entre todos os vetores de k valores booleanos e $Column(s,i,b)$ é a coluna obtida das sequências s na posição i, considerando o valor da célula ou um branco (dependendo de b)

Assim, a ordem de complexidade estimada para encontrar o alinhamento ótimo é $O(k^2 2^k n^k)$ se usarmos SP.

O problema do alinhamento múltiplo é NP-completo.

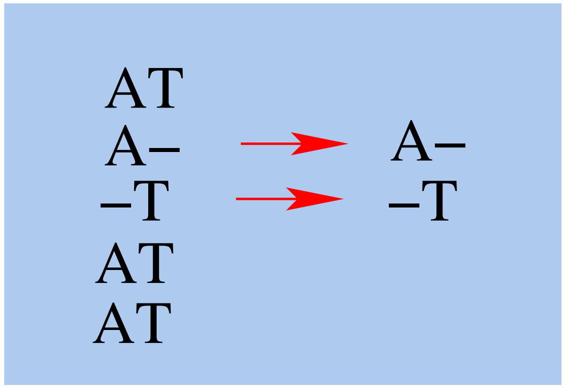
Heurística para Melhorar o Tempo

Temos k sequências de tamanho n, para $1 \leq i \leq k$, e queremos computar o melhor alinhamento de acordo com a medida SP. Só queremos computar o valor de *células relevantes* para o melhor alinhamento, e não todas as células. Mas quais são essas células e por quê?

A resposta está em olhar os pares de projeções da célula. Em um passo de pré-processamento, são criadas condições para permitir o cálculo do teste de relevância...

Teste de Relevância

Seja α um alinhamento ótimo envolvendo $s_1, \ldots, s_k$. Apesar de α ser ótimo, não necessariamente suas projeções são os melhores alinhamentos dos pares de sequências considerados.



Porém, pode-se determinar um *limite mínimo* para os escores das projeções, se tivermos um limite mínimo para o escore ótimo...

Teorema: Seja α um alinhamento ótimo envolvendo $s_1, \ldots, s_k$. Se $SPscore(\alpha) \geq L$ então

$$score(\alpha_{ij}) \geq L_{ij}$$

onde $L_{ij} = L - \sum_{x<y,(x,y)\neq(i,j)}(sim(s_x, s_y))$.

Prova:

$$SPscore(\alpha) \geq L$$
$$\sum_{i<j} score(\alpha_{ij}) \geq L$$
$$\sum_{x<y,(x,y)\neq(i,j)} score(\alpha_{xy}) \geq L - score(\alpha_{ij})$$
$$\sum_{x<y,(x,y)\neq(i,j)} sim(s_x, s_y) \geq L - score(\alpha_{ij})$$

Para testar se uma célula com índice $i = (i_1, \ldots, i_k)$ é relevante para um alinhamento ótimo com respeito a um limite mínimo L deve-se testar se todas as suas projeções satisfazem as condições do teorema, ou seja, i é relevante se

$$c_{xy}[i_x, i_y] \geq L_{xy}$$

para todos x e y tal que $1 \leq x < y \leq k$ onde c_{xy} é a matriz de escores totais para s_x e s_y (c_{xy} é obtida como a soma das matrizes a e b, que são as matrizes que consideram escores de prefixos e sufixos das sequências, como visto nas outras aulas).

Algortimo *Alinhamento Múltiplo*

input $s = (s_1, \ldots s_k)$ e limite mínimo L
output valor do alinhamento ótimo
for all x and y, $1 \leq x < y \leq k$ do
    compute c_{xy} */* o escore total para s_x e s_y*
for all x and y, $1 \leq x < y \leq k$ do
    $L_{xy} \leftarrow L - \sum_{p<q,(p,q)\neq(i,j)}(sim(s_p, s_q))$
$pool \leftarrow \{0\}$
while $pool \neq \emptyset$
    $i \leftarrow$the lexicographically smallest cell in $pool$
    $pool \leftarrow pool - \{i\}$
    if $c_{xy}[i_x, i_y] \geq L_{xy}, \forall x, y,\ 1 \leq x < y \leq k$ */* Relevance Test*
    then for all j dependent on i do
        if $j \notin pool$
        then $pool \leftarrow pool \cup \{j\}$
            $a[j] \leftarrow a[i] + SPscore(Column(s, i, j - i))$
        else
            $a[j] \leftarrow max\{a[j], a[i] + SPscore(Column(s, i, j - i))\}$
return $a[n_1, \ldots, n_k]$