

Alinhamento Múltiplo de Sequências

Zanoni Dias

Instituto de Computação – Unicamp

6 de setembro de 2012

Alinhamento Múltiplo de Sequências

- Dadas k seqüências $\alpha_1, \alpha_2, \dots, \alpha_k$ sobre um alfabeto \mathcal{A} com, respectivamente, n_1, n_2, \dots, n_k caracteres, obter um alinhamento $\alpha = \{\alpha'_1, \alpha'_2, \dots, \alpha'_k\}$, sobre o alfabeto $\mathcal{A}' = \mathcal{A} \cup \{-\}$, tal que, $|\alpha'_1| = |\alpha'_2| = \dots = |\alpha'_k| = n$, e α_i possa ser obtida através da remoção de todos os buracos (-) de α'_i (para todo $1 \leq i \leq k$).
- O alinhamento normalmente é representado por uma matriz de dimensões n e k , onde as linhas representam as seqüências.
- Uma coluna, por definição, não pode conter apenas buracos.
- Dado um esquema de pontuação para alinhamentos múltiplos, desejamos encontrar o alinhamento de maior pontuação possível.
- O problema do Alinhamento Múltiplo de Sequências é também conhecido como MSA (*Multiple Sequence Alignment*).

Alinhamento Múltiplo de Sequências

```
P E A A L Y G R F T I K S D V W
E A A L Y G R F T I E S D V W
P E S L A Y N K F S I K S D V W
P E A L N Y G R Y S S E S D V W
P E A L N Y G W Y S S E S D V W
P E V I R M Q D D N P F S F Q S D V Y
```

Alinhamento Múltiplo de Sequências

```
P E A A L Y G R F T - - - I K S D V W
- E A A L Y G R F T - - - I E S D V W
P E S L A Y N K F - - - S I K S D V W
P E A L N Y G R Y - - - S S E S D V W
P E A L N Y G W Y - - - S S E S D V W
P E V I R M Q D D N P F S F Q S D V Y
```

Como Pontuar um Alinhamento Múltiplo de Sequências

- Soma da pontuação de todas as colunas do alinhamento.
 - ▶ Necessita de uma função de pontuação de colunas.
- Exemplo de funções de pontuação de colunas:
 - ▶ Generalização da matriz de similaridade, com k dimensões.
 - ▶ Soma de Pares (SP-score: Sum-of-Pairs).
 - ▶ Entropia da coluna.

Pontuação baseada em Entropia

- Quanto mais similar forem os símbolos de uma coluna, menor a entropia.
- A pontuação de uma alinhamento pode ser obtido pela soma das entropias das colunas.
- Neste caso, estamos interessados num alinhamento de entropia mínima.
- Fórmula da entropia de uma coluna:

$$-\sum_{x \in \mathcal{A}'} p_x \log_2 p_x$$

onde p_x é a frequência do símbolo x na coluna.

- Note que se $p_x = 1$, ou seja, a coluna contiver apenas o símbolo x , então a entropia da coluna será $-1 \log_2 1 = 0$.
- Caso, a coluna contiver mais de um símbolo, então a entropia será positiva. Exemplo, $p_A = p_C = p_T = p_G = \frac{1}{4}$, então a entropia será $-4(\frac{1}{4} \log_2 \frac{1}{4}) = 2$.
- A entropia de uma coluna pode ser calculada em $\Theta(|\mathcal{A}| + k)$.

Soma de Pares

- Considera a soma, par a par, das similaridades de todos os símbolos da coluna.
- Fórmula da Soma de Pares para uma coluna c :

$$\sum_{1 \leq i < j \leq k} \sigma(\alpha_i[c], \alpha_j[c])$$

- A Soma de Pares de uma coluna pode ser calculada em:
 - ▶ Tempo $\Theta(k^2)$
- Soma de pares pode ser usada para avaliar o alinhamento como um todo, e com isso considerar esquemas de penalidade sub-aditivos para buracos.
- Neste caso teríamos:

$$\sum_{1 \leq i < j \leq k} sim(\alpha_i, \alpha_j)$$

Sequência Consenso

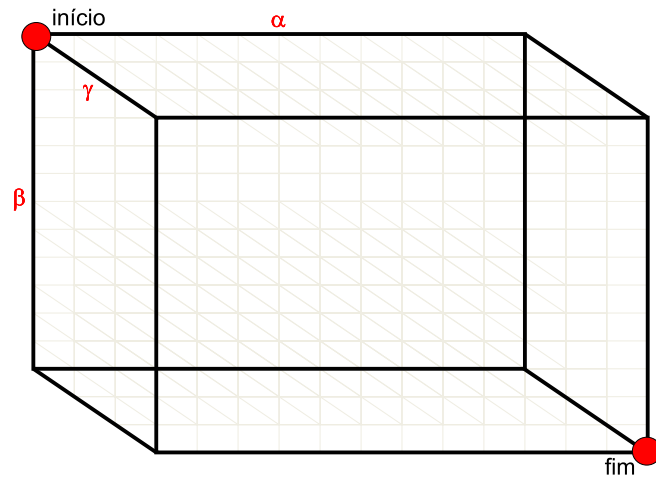
- Em muitas aplicações, além do alinhamento das sequências, deseja-se obter uma sequência que represente o consenso do alinhamento.
- Método ingênuo: coluna a coluna, fazer uma “votação”, escolhendo a base mais comum.
- A sequência consenso (C) pode ser obtida, coluna a coluna, escolhendo o símbolo que maximiza a soma das similaridade entre ele e todos os demais símbolos da coluna, ou seja:

$$\text{maximize} \sum_{i=1}^k \sigma(C[c], \alpha_i[c])$$

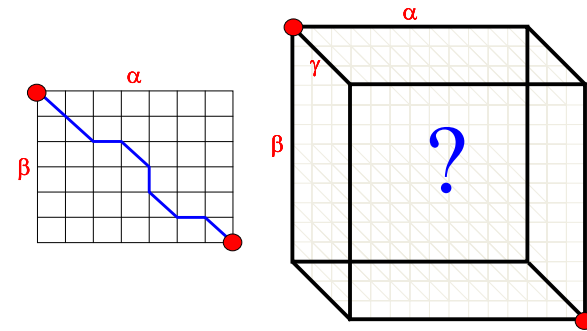
com $C[c] \in \mathcal{A}'$, para toda coluna c do alinhamento múltiplo.

- A sequência consenso pode ser obtida em $\Theta(|\mathcal{A}|kn)$.

Alinhamento de Três Sequências



Alinhamento de Três Sequências



Alinhamento de Três Sequências

- Generalização do algoritmo de Needleman e Wunsch para alinhamento de duas seqüências.
- Matriz de Programação Dinâmica deverá ser tridimensional (cada dimensão representará uma seqüência a ser alinhada).

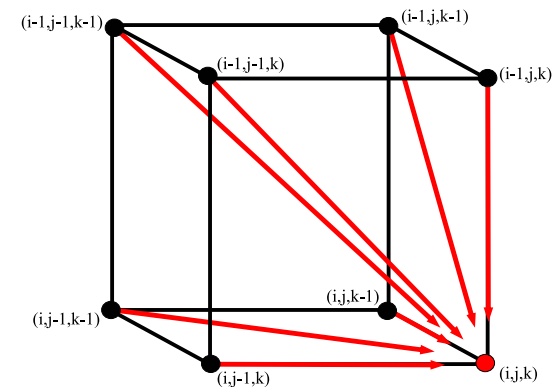
- Fórmula de recorrência usada no preenchimento da matriz:

$$M[i,j,k] \leftarrow \max \left\{ \begin{array}{l} M[i-1,j,k] + \sigma(\alpha_1[i], -, -) \\ M[i,j-1,k] + \sigma(-, \alpha_2[j], -) \\ M[i,j,k-1] + \sigma(-, -, \alpha_3[k]) \\ M[i-1,j-1,k] + \sigma(\alpha_1[i], \alpha_2[j], -) \\ M[i-1,j,k-1] + \sigma(\alpha_1[i], -, \alpha_3[k]) \\ M[i,j-1,k-1] + \sigma(-, \alpha_2[j], \alpha_3[k]) \\ M[i-1,j-1,k-1] + \sigma(\alpha_1[i], \alpha_2[j], \alpha_3[k]) \end{array} \right.$$

- Complexidade de tempo e espaço:

▶ $\Theta(n^3)$

Alinhamento de Três Sequências



Alinhamento de k Sequências

- Generalização do algoritmo de Needleman e Wunsch para alinhamento de duas sequências.
- Matriz de Programação Dinâmica deverá ser k -dimensional (cada dimensão representará uma sequência a ser alinhada).
- Cada célula da matriz dependerá de outras $2^k - 1$ células.
- Quanto custa preencher cada célula?
 - ▶ Usando Soma de Pares: $\Theta(k^2 2^k)$
 - ▶ Usando Entropia: $\Theta((|A| + k)2^k)$
- Complexidade de tempo total:
 - ▶ $\Omega(k 2^k n^k)$
- Complexidade de espaço:
 - ▶ $\Theta(n^k)$
- Lusheng Wang e Tao Jiang provaram em 1994 que o problema do alinhamento múltiplo de sequências é \mathcal{NP} -Completo.

Alinhamento de k Sequências Redução do Espaço de Busca

- Método para redução de tempo de processamento quando usa-se Soma de Pares para pontuar cada coluna.
- Antes de expandir uma célula (e atualizar a similaridade das células que são influenciadas por ela), verificar se ela é relevante, ou seja, se ela pode fazer parte do alinhamento múltiplo.
- O método usa as matrizes de pontuação total entre todos os pares de sequências a serem alinhadas.
- A Matriz de Pontuação Total (c) entre as sequências α e β , de tamanho m e n , é definida como:

$$c[i, j] = a[i, j] + b[i, j]$$

onde:

$$a[i, j] = \text{sim}(\alpha[1..i], \beta[1..j])$$

$$b[i, j] = \text{sim}(\alpha[i+1..n], \beta[j+1..m])$$

Alinhamento de k Sequências Matriz de Pontuação de Prefixos

a	α	A	C	T	G	A	G	T	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40
A	-5	3	-2	-7	-12	-17	-22	-27	-32
T	-10	-2	1	1	-4	-9	-14	-19	-24
T	-15	-7	-4	4	-1	-6	-11	-11	-16
G	-20	-12	-9	-1	7	2	-3	-8	-13
A	-25	-17	-14	-6	2	10	5	0	-5
G	-30	-22	-19	-11	-3	5	13	8	3

Alinhamento de k Sequências Matriz de Pontuação de Sufixos

b	3	-5	-7	-10	-8	-16	-19	-27	-30	A
	-5	0	-3	-5	-8	-11	-14	-22	-25	T
	-8	-3	2	-6	-3	-6	-9	-17	-20	T
	-16	-11	-6	-1	-6	-1	-9	-12	-15	G
	-24	-19	-14	-9	-4	-9	-4	-7	-10	A
	-32	-27	-22	-17	-12	-7	-7	-2	-5	G
	-40	-35	-30	-25	-20	-15	-10	-5	0	β
	A	C	T	G	A	G	T	C	α	

Alinhamento de k Sequências Matriz de Pontuação Total

β	α	A	C	T	G	A	G	T	C
α	3	-10	-17	-25	-28	-41	-49	-62	-70
A	-10	3	-5	-12	-20	-28	-36	-49	-57
T	-18	-5	3	-5	-7	-15	-23	-36	-44
T	-31	-18	-10	3	-7	-7	-20	-23	-31
G	-44	-31	-23	-10	3	-7	-7	-15	-23
A	-57	-44	-36	-23	-10	3	-2	-2	-10
G	-70	-57	-49	-36	-23	-10	3	3	3

Alinhamento de k Sequências Redução do Espaço de Busca

Teorema

Seja α um alinhamento ótimo entre as sequências $\alpha_1, \alpha_2, \dots, \alpha_k$ e α_{ij} a projeção do alinhamento entre α_i e α_j . Se $SP\text{-score}(\alpha) \geq L$, então:

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

onde:

$$L_{ij} = L - \sum_{1 \leq x < y \leq k, (x,y) \neq (i,j)} sim(\alpha_x, \alpha_y)$$

Alinhamento de k Sequências Redução do Espaço de Busca

Lema

Se a célula $M[i_1, i_2, \dots, i_k]$ é relevante, então:

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

para todo par x e y , tal que, $1 \leq x < y \leq k$, onde c_{xy} é a matriz de pontuação total entre α_x e α_y .

Alinhamento de k Sequências Redução do Espaço de Busca

Algoritmo 1: MSA

```

Data:  $k, n_1, n_2, \dots, n_k, \alpha_1, \alpha_2, \dots, \alpha_k, L$ 
for all  $x$  e  $y, 1 \leq x < y \leq k$  do Calcule  $c_{xy}$ 
for all  $x$  e  $y, 1 \leq x < y \leq k$  do  $L_{xy} \leftarrow L - \sum_{1 \leq p < q \leq k, (p,q) \neq (x,y)} sim(\alpha_p, \alpha_q)$ 
 $M[n_1, n_2, \dots, n_k] \leftarrow -\infty$ 
 $pool \leftarrow \{0\}$ 
while  $pool \neq \emptyset$  do
   $i \leftarrow$  the lexicographically smallest cell in the  $pool$ 
   $pool \leftarrow pool \setminus i$ 
  if  $c_{xy}[i_x, i_y] \geq L_{xy}$ , for all pair  $x$  e  $y$ , where  $1 \leq x < y \leq k$  then
    for each cell  $j$  dependent on  $i$  do
      if  $j \notin pool$  then
         $pool \leftarrow pool \cup \{j\}$ 
         $M[j] \leftarrow M[i] + SP\text{-score}(Column(\alpha, j - 1))$ 
      end
    else
       $M[j] \leftarrow \max(M[j], M[i] + SP\text{-score}(Column(\alpha, j - 1)))$ 
    end
  end
end
end
return  $M[n_1, n_2, \dots, n_k]$ 

```

Similaridade x Distância

Alinhamento de k Sequências Redução do Espaço de Busca

- Complexidade de tempo:
 - $\Theta(k^2 n^2 + k^4 + r 2^k k^2)$
onde r é o número de células relevantes.
 - Pior caso: $r = \Theta(n^k)$
 - Logo, a complexidade de pior caso é $\Theta(n^k 2^k k^2)$
- Método proposto por Humberto Carrillo e David Lipman em 1988.
- Implementado no programa MSA, de David Lipman, Stephen Altschul e John Kececioglu (1989).

- Propriedades de distância (ou métrica) para sequências:
 - $\delta(x, x) = 0$, para todo $x \in \mathcal{A}$.
 - $\delta(x, y) > 0$, com $x \neq y$, para todo $x, y \in \mathcal{A}$.
 - $\delta(x, y) = \delta(y, x)$, para todo par $x, y \in \mathcal{A}$.
 - $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$, para toda tripla $x, y, z \in \mathcal{A}$.
- Distância não é adequada para uso em comparação local.
- Se $\sigma(x, x) = M$ e $\sigma(x, -) = g$, para todo $x \in \mathcal{A}$, então podemos usar as seguintes definições:
 - $\delta(x, y) = M - \sigma(x, y)$.
 - $g' = -g + \frac{M}{2}$.
onde g' é a o "custo" de se alinhar uma base com um buraco, de tal forma que a seguinte relação de equivalência é verdadeira:
 - $sim(\alpha, \beta) + dist(\alpha, \beta) = \frac{M}{2}(m + n)$.
- Equivalência descrita por Temple Smith, Michael Waterman e Walter Fitch, em 1981.

Compatibilidade de Alinhamentos de Pares de Sequências

```

α1 = A A A A T T T T
α2 = T T T T G G G G
α3 = A A A A G G G G

α1 = A A A A T T T T - - - -
α2 = - - - - T T T T G G G G

α2 = - - - - T T T T G G G G
α3 = A A A A - - - - G G G G

α1 = A A A A T T T T - - - -
α3 = A A A A - - - - G G G G
    
```

Compatibilidade de Alinhamentos de Pares de Sequências

```

α1 = A A A A T T T T
α2 = T T T T G G G G
α3 = A A A A G G G G

α1 = A A A A T T T T - - - -
α2 = - - - - T T T T G G G G
α3 = A A A A - - - - G G G G
    
```

Incompatibilidade de Alinhamentos de Pares de Sequências

$\alpha_1 =$ A A A A T T T T
 $\alpha_2 =$ T T T T G G G G
 $\alpha_3 =$ G G G G A A A A

 $\alpha_1 =$ A A A A T T T T - - - -
 $\alpha_2 =$ - - - - T T T T G G G G

 $\alpha_2 =$ T T T T G G G G - - - -
 $\alpha_3 =$ - - - - G G G G A A A A

 $\alpha_1 =$ - - - - A A A A T T T T
 $\alpha_3 =$ G G G G A A A A - - - -

Alinhamento Estrela

- Construir um alinhamento múltiplo, usando uma sequência como âncora para as demais.
- Como escolher a âncora?
 - ▶ Use cada uma das sequências como âncora e retorne apenas o melhor alinhamento.
 - ▶ Use a sequência que maximiza a soma das similaridades em relação a todas as demais sequências.
- Passos:
 - ▶ Calcule os alinhamentos entre todos os pares de sequências.
 - ▶ Escolha a âncora.
 - ▶ Adicione, uma a uma, as demais sequências ao alinhamento.
 - ★ Use a regra: “once a gap, always a gap”.
- O valor do alinhamento ótimo obtido pode ser usado como limite inferior (L) para o algoritmo de Carrillo e Lipman.
- Complexidade:
 - ▶ $\Theta(k^2n^2)$

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T
 $\alpha_3 =$ A T C C A A T T T T
 $\alpha_4 =$ A T C T T C T T
 $\alpha_5 =$ A C T G A C C

Alinhamento Estrela

sim	α_1	α_2	α_3	α_4	α_5	soma
α_1		22	-1	4	-4	21
α_2	22		-1	4	-7	18
α_3	-1	-1		4	-14	-12
α_4	4	4	4		-4	8
α_5	-4	-7	-14	-4		-29
soma	21	18	-12	8	-29	

Match = 3

Mismatch = -2

Gap = -5

Alinhamento Estrela

dist	α_1	α_2	α_3	α_4	α_5	soma
α_1		5,0	29,5	21,5	28,0	84,0
α_2	5,0		29,5	21,5	31,0	87,0
α_3	29,5	29,5		23,0	39,5	121,5
α_4	21,5	21,5	23,0		26,5	92,5
α_5	28,0	31,0	39,5	26,5		125,0
soma	84,0	87,0	121,5	92,5	125,0	

Match = 0 Mismatch = -5 Gap = -6,5

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T

$\alpha_1 =$ A T T G C C A T T
 $\alpha_4 =$ A T C T T C - T T

$\alpha_1 =$ A T T G C C A T T
 $\alpha_5 =$ A C T G A C C - -

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_2 =$ A T G G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_2 =$ A T G G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T

 $\alpha_1 =$ A T T G C C A T T
 $\alpha_4 =$ A T C T T C - T T

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_2 =$ A T G G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T
 $\alpha_4 =$ A T C T T C - - - T T

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_2 =$ A T G G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T
 $\alpha_4 =$ A T C T T C - - - T T

 $\alpha_1 =$ A T T G C C A T T
 $\alpha_5 =$ A C T G A C C - -

Alinhamento Estrela

$\alpha_1 =$ A T T G C C A - - T T
 $\alpha_2 =$ A T G G C C A - - T T
 $\alpha_3 =$ A T C - C A A T T T T
 $\alpha_4 =$ A T C T T C - - - T T
 $\alpha_5 =$ A C T G A C C - - - -

Alinhamento Estrela

- Seja:
 - $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$: o conjunto das k seqüências a serem alinhadas.
 - α^* : o alinhamento estrela de α .
 - α' : o alinhamento ótimo de α .
 - α_c : a seqüência usada como âncora do alinhamento estrela.
 - $dist(\alpha_i, \alpha_j)$: distância ótima entre α_i e α_j .
 - $dist'(\alpha_i, \alpha_j)$: distância entre α_i e α_j no alinhamento ótimo.
 - $dist^*(\alpha_i, \alpha_j)$: distância entre α_i e α_j no alinhamento estrela.
 - $V(\alpha) = \sum_{1 \leq i, j \leq k} dist(\alpha_i, \alpha_j) \leq 2 \text{ SP-Score}(\alpha')$.
 - $V(\alpha') = \sum_{1 \leq i, j \leq k} dist'(\alpha_i, \alpha_j) = 2 \text{ SP-Score}(\alpha')$.
 - $V(\alpha^*) = \sum_{1 \leq i, j \leq k} dist^*(\alpha_i, \alpha_j) = 2 \text{ SP-Score}(\alpha^*)$.
- Note que: $V(\alpha) \leq V(\alpha') \leq V(\alpha^*)$.

Alinhamento Estrela

Alinhamento Estrela

Lema

Para quaisquer seqüências α_i e α_j , com $1 \leq i, j \leq k$, temos que:
 $dist^*(\alpha_i, \alpha_j) \leq dist^*(\alpha_i, \alpha_c) + dist^*(\alpha_c, \alpha_j) = dist(\alpha_i, \alpha_c) + dist(\alpha_c, \alpha_j)$.

Teorema

$$V(\alpha^*)/V(\alpha) \leq 2 \frac{(k-1)}{k} < 2$$

Prova:

$$V(\alpha^*) = \sum_{1 \leq i, j \leq k} dist^*(\alpha_i, \alpha_j) \leq \sum_{1 \leq i, j \leq k} [dist(\alpha_i, \alpha_c) + dist(\alpha_c, \alpha_j)]$$

Note que $dist(\alpha_i, \alpha_c) (= dist(\alpha_c, \alpha_i))$ aparece $2(k-1)$ vezes no somatório do lado direito da expressão anterior, logo:

$$V(\alpha^*) \leq 2(k-1) \sum_{1 \leq j \leq k} dist(\alpha_c, \alpha_j) = 2(k-1)M$$

Por outro lado temos:

$$V(\alpha) = \sum_{1 \leq i, j \leq k} dist(\alpha_i, \alpha_j) \geq k \sum_{1 \leq i \leq k} dist(\alpha_i, \alpha_c) = kM$$

Logo:

$$V(\alpha^*)/V(\alpha) \leq \frac{2(k-1)M}{kM} = 2 \frac{(k-1)}{k} < 2$$

□

Logo, Alinhamento Estrela é um algoritmo de 2-aproximação.

Algoritmos de Aproximação para Alinhamento Múltiplo de Sequências

(usando Soma de Pares com matrizes métricas)

- Daniel Gusfield, 1993.
 - ▶ Aproximação: $2 - 2/k$.
 - ▶ Complexidade: $O(k^2 n^2)$.
- Pavel Pevzner, 1992.
 - ▶ Aproximação: $2 - 3/k$.
 - ▶ Complexidade: $O(n^3 k^3 + k^4)$.
- Vineet Bafna, Eugene Lawler e Pavel Pevzner, 1997.
 - ▶ Aproximação: $2 - 1/k$.
 - ▶ Complexidade: $O(k^{l+1}(2^k + k m(l, n)))$
Onde $m(l, n)$ é a complexidade de se obter um alinhamento múltiplo ótimo de l sequências de tamanho n .
Usando MSA, temos que $m(l, n) = O(l^2 2^l n^l)$.

Alinhamento Progressivo

- Consiste em construir um alinhamento múltiplo a partir de alinhamentos de pares de sequências e/ou de alinhamentos.
- Descrito inicialmente por Hogeweg e Hesper (1984) e depois reinventado por Feng e Doolittle (1987) e Taylor (1988).
- Características:
 - ▶ Simples e efetivo para MSA.
 - ▶ Requer pouco tempo e memória.
 - ▶ Bom desempenho para sequências homólogas e relativamente bem conservadas.
 - ▶ Problema: natureza gulosa e muito sensível ao esquema de pontuação.

Algoritmos de Aproximação para Alinhamento Múltiplo de Sequências

(usando Soma de Pares com matrizes genéricas)

- Winfried Just, 2001.
 - ▶ MSA \in $\mathcal{MAX}\text{-SNP}$ -Difícil.
 - ▶ Não existe um esquema de aproximação polinomial (PTAS - *Polynomial Time Approximation Scheme*) para MSA, a menos que $\mathcal{P} = \mathcal{NP}$.

Alinhamento Progressivo

- Etapas:
 1. Computar alinhamentos de todos os pares de sequências.
 2. Construir uma árvore guia.
 3. Construir o alinhamento múltiplo guiado pela árvore.
- Construção de árvore guia:
 - ▶ UPGMA (Sneath e Sokal, 1973)
 - ▶ Neighbor-Joining (Saitou e Nei, 1987)
- Construção do alinhamento múltiplo:
 - ▶ Seleção do par a incluir no alinhamento.
 - ▶ Alinhar duas sequências/alinhamentos.
- Programas que implementam alinhamento progressivo:
 - ▶ Clustal W (Thompson et al., 1994)
 - ▶ MUSCLE (Edgar, 2004)
 - ▶ T-COFFEE (Notredame et al., 2000)
 - ▶ ProbCons (Do et al., 2005)

Alinhamento Progressivo

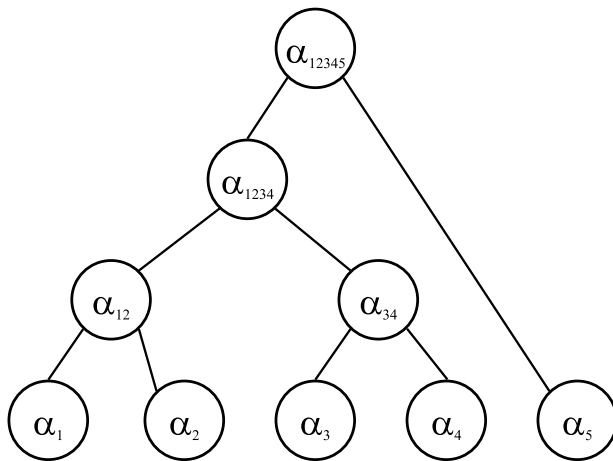
$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T
 $\alpha_3 =$ A T C C A A T T T T
 $\alpha_4 =$ A T C T T C T T
 $\alpha_5 =$ A C T G A C C

Alinhamento Progressivo

sim	α_1	α_2	α_3	α_4	α_5	soma
α_1		22	-1	4	-4	21
α_2	22		-1	4	-7	18
α_3	-1	-1		4	-14	-12
α_4	4	4	4		-4	8
α_5	-4	-7	-14	-4		-29
soma	21	18	-12	8	-29	

Match = 3 Mismatch = -2 Gap = -5

Alinhamento Progressivo



Alinhamento Progressivo

$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T

Alinhamento Progressivo

$\alpha_1 =$ A T T G C C A T T
 $\alpha_2 =$ A T G G C C A T T
 $\alpha_3 =$ A T C C A A T T T T
 $\alpha_4 =$ A T C - - T T C T T

Alinhamento Progressivo

$\alpha_1 =$ A T T G C C A - T T
 $\alpha_2 =$ A T G G C C A - T T
 $\alpha_3 =$ A T C C A A T T T T
 $\alpha_4 =$ A T C - - T T C T T

Alinhamento Progressivo

$\alpha_1 =$ A T T G C C A - T T
 $\alpha_2 =$ A T G G C C A - T T
 $\alpha_3 =$ A T C C A A T T T T
 $\alpha_4 =$ A T C - - T T C T T
 $\alpha_5 =$ A C T G A C C - - -

Alinhamento de Dois Alinhamentos

- Generalização do algoritmo de alinhamento de duas sequências.
- Cada célula da matriz de programação dinâmica representará o valor do melhor alinhamento possível entre os prefixos de dois alinhamentos.
- Para calcular o custo de se alinhar duas colunas de uma alinhamento, basta calcular o valor da soma de pares (ou entropia) para a nova coluna gerada.
- Complexidade:
 - ▶ Usando soma de pares: $\Theta(mnk^2)$.
 - ▶ Usando entropia: $\Theta(mn(k + \mathcal{A}))$.

Alinhamento Iterativo

- Consiste em refinar alinhamentos através de uma série de ciclos.
- Geralmente é usado para melhorar alinhamentos previamente construídos.
- Problema: requer muito tempo e depende de outros métodos auxiliares.
- Programas que implementam alinhamento múltiplo iterativo:
 - ▶ PRRP: refinamento de um alinhamento progressivo (Gotoh, 1993).
 - ▶ SAGA: algoritmo genético (Notredame e Higgins, 1996).
 - ▶ HMMER: Modelo Ocultos de Markov (Eddy, 1998).