

**Instituto de
Computação**

UNIVERSIDADE ESTADUAL DE CAMPINAS



Alinhamento de Sequências

MO640 - Biologia Computacional / MC668 - Bioinformática

Zanoni Dias

2025

Instituto de Computação

Alinhamento de Sequências

Alinhamento Global

Alinhamento Semi-Global

Alinhamento Local

Pontuação para Alinhamento de Caracteres

Penalidades para Blocos de Buracos

Alinhamento Global usando Espaço Linear

Alinhamento Global de Sequências Similares

Algoritmo Sub-Quadrático para Alinhamento Global

Alinhamento de Sequências

O que é Alinhamento de Sequências?

- Um alinhamento de duas sequências de caracteres α e β é obtido inserindo espaços nas sequências, e então colocando uma sobre a outra, de modo que cada caracter ou espaço esteja emparelhado a um único caracter (ou a um espaço) da outra sequência.
- Não podem haver espaços alinhados com espaços.
- Exemplo:
 - Sequências:
 - $\alpha = \text{AAACTGCACAATCTTAATGCCCTTTTAT}$
 - $\beta = \text{GCGGATCAACTTATTCCATCTCTT}$
 - Alinhamento:
 - $\alpha' = \text{AAACTGCA-CAACTCTTCTTAATGCAGTC--CTTTTAT}$
 - $\beta' = \text{--GC-GGATCAACTCT-CTT-ATTCAGTCATCTCTT--}$

Por que Alinhar Sequências?

- Comparar sequências
- Localizar trechos homólogos
- Predição de estrutura secundária
- Predição de função
- Inferência filogenética

Como Comparar Alinhamentos?

- Alinhamento 1:
 - $\alpha' = \text{CAGG}\textcolor{red}{\text{ATGCTAGCA}}\textcolor{red}{\text{AAACATCGCGGG}}\textcolor{red}{\text{CGATAA}}$
 - $\beta' = \text{--GG}\textcolor{red}{\text{CATG}}\textcolor{red}{\text{TAGCACA}}\textcolor{red}{\text{CACGACGCTGGGAGAAT--}}$
- Alinhamento 2:
 - $\alpha'' = \text{CAGG}\textcolor{red}{\text{ATGCTAGCA}}\textcolor{red}{\text{AAACATCGC--GGG}}\textcolor{red}{\text{CGATAA--}}$
 - $\beta'' = \text{--GG}\textcolor{red}{\text{CATG--TAGCACA}}\textcolor{red}{\text{CACGA--CGCTGGGAG--AAT}}$
- Qual é o melhor?
- Critério objetivo: função de pontuação para alinhamentos.
- Exemplo 1: gap = -1, match = 2, mismatch = -4
 - Alinhamento 1: 3 gaps + 16 matches + 14 mismatches = -27
 - Alinhamento 2: 7 gaps + 22 matches + 6 mismatches = 13
- Exemplo 2: gap = -10, match = 1, mismatch = -1
 - Alinhamento 1: 3 gaps + 16 matches + 14 mismatches = -28
 - Alinhamento 2: 7 gaps + 22 matches + 6 mismatches = -54

Problema

Dadas duas sequências α e β com, respectivamente, m e n caracteres, e um critério de pontuação de alinhamentos, deseja-se obter o alinhamento de pontuação máxima entre estas duas sequências.

- Dado um esquema de pontuação e um alinhamento, é fácil determinar se este é o melhor alinhamento possível entre as duas sequências?
- Como ter certeza que um dado alinhamento é o melhor possível?
 - Calcular todos os alinhamentos possíveis!

Tipos de Alinhamentos

- *Global*: alinhamento de pontuação máxima envolvendo as duas sequências completas. Desejável em situações onde as sequências são similares, por exemplo, ao se alinhar genes ou proteínas homólogas.
- *Semi-Global (ou Semi-Local)*: não penaliza buracos criados nas pontas das sequências. Desejável, por exemplo, no caso de montagem de genomas, onde busca-se um alinhamento de pontuação máxima entre o prefixo de uma sequência e o sufixo da outra (ou vice-versa), ou quando as sequências possuem tamanhos muito diferentes entre si.
- *Local*: alinhamento de pontuação máxima entre qualquer par de subsequências (das sequências originais). Desejável, por exemplo, para se identificar trechos altamente conservados entre dois genomas.

Tipos de Alinhamentos

Global

A C T G T G T A A C C C - G C A A A C T G T A G C G C C G
A - T C T G T A C C C C T G C A A A - T G T A C G C - -

Semi-Global

A C T G T G T A C C C G - G C
 C C C C T G C A A A T T G T A C G G C C G

C G C C G A C G A C C C T G C A A A C T G T T A C A C A T
 C G C C C - T G C A A A

Local

A C T G T G T A A C G C - G C A A A C T G T A G C G C
 G A A C G A A A C C T G C A A A G G A A C C T G C C A

Alinhamento Global

Pontuação do Alinhamento Ótimo

- Inicialmente vamos considerar uma simplificação do problema de encontrar o alinhamento de pontuação ótima, onde estamos apenas interessados na pontuação desse alinhamento, e não no alinhamento em si.
- Posteriormente vamos mostrar como podemos construir o alinhamento desejado, com base na solução para obtenção da pontuação do alinhamento ótimo.
- A princípio vamos considerar abordagens de força bruta para obter a pontuação do alinhamento ótimo.

- Onde a primeira base de uma das sequências pode se alinhar?
 - Opção 1:
 - **C**[CAGCCGAATCGATCGCATG]
 - **-**[CATCAGCGATCGATCTTT]
 - Opção 2:
 - **C**[CAGCCGAATCGATCGCATG]
 - **C**[ATCAGCGATCGATCTTT]
 - Opção 3:
 - **-C**[CAGCCGAATCGATCGCATG]
 - **C-**[ATCAGCGATCGATCTTT]
 - Opção 4:
 - **-C**[CAGCCGAATCGATCGCATG]
 - **CA**[TCAGCGATCGATCTTT]
 - ...

Alinhamento Global - Força Bruta - 1ª Abordagem

- Onde a primeira base de uma das sequências pode se alinhar?
 - ...
 - Opção $2n - 2$:
 - -----C[CAGCCGAATCGATCGCATG]
 - CATCAGCGATCGATCTT[T]
 - Opção $2n - 1$:
 - -----C[CAGCCGAATCGATCGCATG]
 - CATCAGCGATCGATCTT-[T]
 - Opção $2n$:
 - -----C[CAGCCGAATCGATCGCATG]
 - CATCAGCGATCGATCTTT[]
 - Opção $2n + 1$:
 - -----C[CAGCCGAATCGATCGCATG]
 - CATCAGCGATCGATCTTT-[]

Algoritmo 1: Align

Input: $\alpha, a_i, a_f, \beta, b_i, b_f$

if $a_i > a_f$ **then**

return $gap \times (b_f - b_i + 1)$

end

if $b_i > b_f$ **then**

return $gap \times (a_f - a_i + 1)$

end

$max \leftarrow gap + Align(\alpha, a_i + 1, a_f, \beta, b_i, b_f)$

for all $b_k \in [b_i..b_f]$ **do**

$aux \leftarrow Align(\alpha, a_i + 1, a_f, \beta, b_k + 1, b_f)$

$aux1 \leftarrow gap \times (b_k - b_i) + \sigma(\alpha[a_i], \beta[b_k]) + aux$

$aux2 \leftarrow gap \times (b_k - b_i + 2) + aux$

if $aux1 > max$ **then**

$max \leftarrow aux1$

end

if $aux2 > max$ **then**

$max \leftarrow aux2$

end

end

return max

Complexidade e Tempo de Execução

- Complexidade:
 - $T(m, n) = \sum_{i=0}^n T(m-1, i) + (n+1)$
 - $T(m, n) \geq \sum_{i=0}^{n-1} T(m-1, i) + (n+1)$
 - $T(m, n) \geq \sum_{i=0}^{\min\{m, n\}-1} T(\min\{m, n\}-1, i) + (\min\{m, n\} + 1)$
 - $T(m, n) \geq \sum_{i=0}^{\min\{m, n\}-1} 2^i \geq 2^{\min\{m, n\}} - 1$
 - $T(m, n) = \Omega(2^{\min\{m, n\}})$
- Exemplo: computador de 3GHz (1 instrução por ciclo)
 - $m = n = 25$:
 - Tempo: $2^{25} / (3 \times 2^{30}) = 1$ centésimos de segundo
 - $m = n = 50$:
 - Tempo: $2^{50} / (3 \times 2^{30}) = 4$ dias
 - $m = n = 75$:
 - Tempo: $2^{75} / (3 \times 2^{30}) = 370$ mil anos
 - $m = n = 100$:
 - Tempo: $2^{100} / (3 \times 2^{30}) = 12$ trilhões de anos

- Quais as possíveis opções para a última coluna do alinhamento?
 - Opção 1:
 - [CCAGCCGAATCGATCGCAT] **G**
 - [CATCAGCGATCGATC] **T**
 - Opção 2:
 - [CCAGCCGAATCGATCGCAT] **G**
 - [CATCAGCGATCGATCT] **-**
 - Opção 3:
 - [CCAGCCGAATCGATCGCATG] **-**
 - [CATCAGCGATCGATC] **T**

Algoritmo 2: Align

Input: α, m, β, n

if $m = 0$ **then**
 return $gap \times n$

end

if $n = 0$ **then**
 return $gap \times m$

end

return $max \left\{ \begin{array}{l} Align(\alpha, m - 1, \beta, n - 1) + \sigma(\alpha[m], \beta[n]), \\ Align(\alpha, m - 1, \beta, n) + gap, \\ Align(\alpha, m, \beta, n - 1) + gap \end{array} \right\}$

Complexidade e Tempo de Execução

- Complexidade:
 - $T(m, n) = T(m - 1, n - 1) + T(m, n - 1) + T(m - 1, n) + \Theta(1)$
 - $T(m, n) \geq 3T(m - 1, n - 1) + \Theta(1)$
 - $T(m, n) \geq 3T(\min\{m, n\} - 1, \min\{m, n\} - 1) + \Theta(1)$
 - $T(m, n) = \Omega(3^{\min\{m, n\}})$
- Exemplo: computador de 3GHz (1 instrução por ciclo)
 - $m = n = 10$:
 - Tempo: $3^{10} / (3 \times 2^{30}) = 18$ milionésimos de segundo
 - $m = n = 20$:
 - Tempo: $3^{20} / (3 \times 2^{30}) = 1$ segundo
 - $m = n = 30$:
 - Tempo: $3^{30} / (3 \times 2^{30}) = 18$ horas
 - $m = n = 40$:
 - Tempo: $3^{40} / (3 \times 2^{30}) = 120$ anos
 - $m = n = 50$:
 - Tempo: $3^{50} / (3 \times 2^{30}) = 7$ milhões de anos

Contagem de Subproblemas

- O grande problema do método anterior é que o alinhamento de dois prefixos de α e de β podem ser recalculado muitas vezes.
- Existem quantos subproblemas distintos, envolvendo alinhamentos de prefixos não vazios de α e de β ?
 - Apenas $m \times n$ subproblemas!
- Memorização (*memoization*): evita o recálculo de subproblemas, armazenando os valores previamente calculados (usando um vetor, uma matriz ou um *hash*).
- No nosso caso, podemos usar uma matriz para armazenar os valores dos alinhamentos ótimos entre todos prefixos de α e todos os prefixos de β .

Algoritmo 3: Memoization

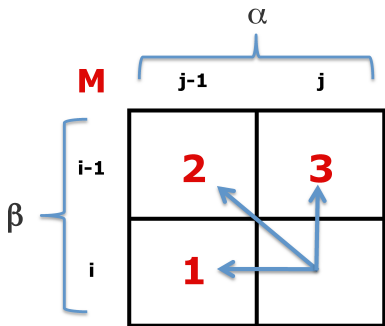
Input: α, m, β, n
for *all* $i \in [1..n]$ **do**
 for *all* $j \in [1..m]$ **do**
 $M[i, j] \leftarrow -\infty$
 end
end
return $\text{AlignMemoization}(\alpha, m, \beta, n)$

Algoritmo 4: AlignMemoization

Input: α, m, β, n
if $m = 0$ **then return** $\text{gap} \times n$;
if $n = 0$ **then return** $\text{gap} \times m$;
if $M[n - 1, m - 1] = -\infty$ **then** $M[n - 1, m - 1] \leftarrow \text{AlignMemoization}(\alpha, m - 1, \beta, n - 1)$;
if $M[n - 1, m] = -\infty$ **then** $M[n - 1, m] \leftarrow \text{AlignMemoization}(\alpha, m, \beta, n - 1)$;
if $M[n, m - 1] = -\infty$ **then** $M[n, m - 1] \leftarrow \text{AlignMemoization}(\alpha, m - 1, \beta, n)$;
return $\max \left\{ \begin{array}{l} M[n - 1, m - 1] + \sigma(\alpha[m], \beta[n]), \\ M[n - 1, m] + \text{gap}, \\ M[n, m - 1] + \text{gap} \end{array} \right\}$

- Ideias básicas:
 - Matriz M armazena os valores dos alinhamentos ótimos entre todos prefixos de α e todos os prefixos de β .
 - O valor do alinhamento ótimo entre as duas sequências deve estar armazenado na posição $M[n,m]$.
 - A matriz M pode ser preenchida, numa ordem adequada, sem a necessidade de nenhuma chamada recursiva.
 - Para simplificar o algoritmo, podemos armazenar na matriz M os valores dos alinhamentos ótimos de prefixos de α ou de β com a sequência vazia.
- Em 1970, Saul Needleman e Christian Wunsch propuseram o algoritmo de programação dinâmica para alinhamento de sequências.

Matriz de Programação Dinâmica



1	$M[i, j-1] + \text{gap}$	<table> <tr> <td>α</td><td>$j-1$</td><td>α_j</td></tr> <tr> <td>β</td><td>i</td><td>-</td></tr> </table>	α	$j-1$	α_j	β	i	-
α	$j-1$	α_j						
β	i	-						
2	$M[i-1, j-1] + \text{match/mismatch}$	<table> <tr> <td>α</td><td>$j-1$</td><td>α_j</td></tr> <tr> <td>β</td><td>$i-1$</td><td>β_i</td></tr> </table>	α	$j-1$	α_j	β	$i-1$	β_i
α	$j-1$	α_j						
β	$i-1$	β_i						
3	$M[i-1, j] + \text{gap}$	<table> <tr> <td>α</td><td>j</td><td>-</td></tr> <tr> <td>β</td><td>$i-1$</td><td>β_i</td></tr> </table>	α	j	-	β	$i-1$	β_i
α	j	-						
β	$i-1$	β_i						

Algoritmo 5: Global

Input: α, m, β, n

for *all* $i \in [0..n]$ **do**

$M[i, 0] \leftarrow gap \times i$

end

for *all* $j \in [1..m]$ **do**

$M[0, j] \leftarrow gap \times j$

end

for *all* $i \in [1..n]$ **do**

for *all* $j \in [1..m]$ **do**

$$M[i, j] \leftarrow \max \left\{ \begin{array}{l} M[i, j-1] + gap, \\ M[i-1, j-1] + \sigma(\alpha[j], \beta[i]), \\ M[i-1, j] + gap \end{array} \right\}$$

end

end

return $M[n, m]$

Complexidade e Tempo de Execução

- Complexidade:
 - $T(m, n) = \Theta(mn)$
- Exemplo: computador de 3GHz (1 instrução por ciclo)
 - $m = n = 10$:
 - Tempo: $(10 \times 10)/(3 \times 2^{30}) = 31$ bilionésimos de segundo
 - $m = n = 100$:
 - Tempo: $(100 \times 100)/(3 \times 2^{30}) = 3$ milionésimos de segundo
 - $m = n = 1000$:
 - Tempo: $(1000 \times 1000)/(3 \times 2^{30}) = 310$ milionésimos de segundo
 - $m = n = 10000$:
 - Tempo: $(10000 \times 10000)/(3 \times 2^{30}) = 31$ milésimos de segundo
 - $m = n = 100000$:
 - Tempo: $(100000 \times 100000)/(3 \times 2^{30}) = 3$ segundos
 - $m = n = 1000000$:
 - Tempo: $(1000000 \times 1000000)/(3 \times 2^{30}) = 5$ minutos

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5											
T	-10											
T	-15											
G	-20											
G	-25											
C	-30											
C	-35											
A	-40											
C	-45											

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3										
T	-10											
T	-15											
G	-20											
G	-25											
C	-30											
C	-35											
A	-40											
C	-45											

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2									
T	-10											
T	-15											
G	-20											
G	-25											
C	-30											
C	-35											
A	-40											
C	-45											

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7								
T	-10											
T	-15											
G	-20											
G	-25											
C	-30											
C	-35											
A	-40											
C	-45											

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10											
T	-15											
G	-20											
G	-25											
C	-30											
C	-35											
A	-40											
C	-45											

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5					
C	-30											
C	-35											
A	-40											
C	-45											

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7

α =

C

β =

C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7

α =

A C

β =

A C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7

α =

A A C

β =

- A C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7

α =

C A A C

β =

C - A C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=							T	C	A	A	C
β	=							-	C	-	A	C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=						G	T	C	A	A	C
β	=						C	-	C	-	A	C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=					G	G	T	C	A	A	C
β	=					G	C	-	C	-	A	C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=				G	G	G	T	C	A	A	C
β	=				G	G	C	-	C	-	A	C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=			T	G	G	G	T	C	A	A	C
β	=			T	G	G	C	-	C	-	A	C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=		C	T	G	G	G	T	C	A	A	C
β	=		T	T	G	G	C	-	C	-	A	C

Alinhamento Global

$\times$ β	α	A	C	T	G	G	G	T	C	A	A	C
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=	A	C	T	G	G	G	T	C	A	A	C
β	=	A	T	T	G	G	C	-	C	-	A	C

Alinhamento Global

$\times$	α	A	C	T	G	G	G	T	C	A	A	C
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
A	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
T	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39
T	-15	-7	-4	4	-1	-6	-11	-11	-16	-21	-26	-31
G	-20	-12	-9	-1	7	2	-3	-8	-13	-18	-23	-28
G	-25	-17	-14	-6	2	10	5	0	-5	-10	-15	-20
C	-30	-22	-14	-11	-3	5	8	3	3	-2	-7	-12
C	-35	-27	-19	-16	-8	0	3	6	6	1	-4	-4
A	-40	-32	-24	-21	-13	-5	-2	1	4	9	4	-1
C	-45	-37	-29	-26	-18	-10	-7	-4	4	4	7	7
α	=	A	C	T	G	G	G	T	C	A	A	C
β	=	A	T	T	G	G	C	-	C	-	A	C

- Como vimos no exemplo anterior, $M[8, 10] = M[8, 9] - 5 = M[7, 9] + 3 = 4$.
- Logo, na penúltima coluna do alinhamento podemos ter tanto um alinhamento de dois caracteres A ($M[7, 9] + 3$), quanto um alinhamento de um A em α com um buraco em β ($M[8, 9] - 5$).
- Sendo assim, podem existir múltiplos alinhamentos ótimos.
- Dois alinhamentos ótimos tem nomes especiais, considerando a ordem de desempate, caso o valor de um célula $M[i, j]$ possa ser obtido de duas ou mais formas diferentes:
 - *Downmost*: alinhamento obtido considerando a ordem de desempate $M[i, j - 1]$, $M[i - 1, j - 1]$ e $M[i - 1, j]$.
 - *Upmost*: alinhamento obtido considerando a ordem de desempate $M[i - 1, j]$, $M[i - 1, j - 1]$ e $M[i, j - 1]$.

Alinhamento Global

$\times$ β	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60	-65	-70	-75	-80	-85	-90	-95
G	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47	-52	-57	-62	-67	-72	-77	-82	-87
G	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39	-44	-49	-54	-59	-64	-69	-74	-79
G	-15	-7	-4	4	-1	-1	-6	-11	-16	-21	-26	-31	-36	-41	-46	-51	-56	-61	-66	-71
T	-20	-12	-9	-1	2	-3	2	-3	-8	-13	-18	-23	-28	-33	-38	-43	-48	-53	-58	-63
T	-25	-17	-14	-6	-3	0	0	5	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
G	-30	-22	-19	-11	-8	0	-2	0	3	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
C	-35	-27	-19	-16	-8	-5	-2	-4	-2	1	1	1	-4	-9	-14	-19	-24	-29	-34	-39
A	-40	-32	-24	-21	-13	-10	-7	-4	-1	-4	4	-1	-1	-1	-6	-11	-16	-21	-26	-31
C	-45	-37	-29	-26	-18	-15	-12	-9	-6	-3	-1	7	2	-3	-3	-3	-8	-13	-18	-23
G	-50	-42	-34	-31	-23	-20	-17	-14	-11	-8	-5	2	5	0	-5	0	-5	-5	-10	-15
C	-55	-47	-39	-31	-28	-20	-22	-19	-16	-8	-10	-3	0	3	3	-2	-2	-7	-7	-7

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Global: Alinhamento Downmost

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60	-65	-70	-75	-80	-85	-90	-95
G	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47	-52	-57	-62	-67	-72	-77	-82	-87
G	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39	-44	-49	-54	-59	-64	-69	-74	-79
G	-15	-7	-4	4	-1	-1	-6	-11	-16	-21	-26	-31	-36	-41	-46	-51	-56	-61	-66	-71
T	-20	-12	-9	-1	2	-3	2	-3	-8	-13	-18	-23	-28	-33	-38	-43	-48	-53	-58	-63
T	-25	-17	-14	-6	-3	0	0	5	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
G	-30	-22	-19	-11	-8	0	-2	0	3	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
C	-35	-27	-19	-16	-8	-5	-2	-4	-2	1	1	1	-4	-9	-14	-19	-24	-29	-34	-39
A	-40	-32	-24	-21	-13	-10	-7	-4	-1	-4	4	-1	-1	-1	-6	-11	-16	-21	-26	-31
C	-45	-37	-29	-26	-18	-15	-12	-9	-6	-3	-1	7	2	-3	-3	-3	-8	-13	-18	-23
C	-50	-42	-34	-31	-23	-20	-17	-14	-11	-8	-5	2	5	0	-5	0	-5	-5	-10	-15
G	-55	-47	-39	-31	-28	-20	-22	-19	-16	-8	-10	-3	0	3	3	-2	-2	-7	-7	-7
α	=	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	=	G	-	G	-	G	T	T	-	G	-	C	-	A	-	C	-	C	-	G
-7	=	3	-5	3	-5	3	3	3	-5	3	-5	3	-5	3	-5	3	-5	3	-5	3

Alinhamento Global: Alinhamento Upmost

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60	-65	-70	-75	-80	-85	-90	-95
G	-5	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47	-52	-57	-62	-67	-72	-77	-82	-87
G	-10	-2	1	1	-4	-9	-14	-19	-24	-29	-34	-39	-44	-49	-54	-59	-64	-69	-74	-79
G	-15	-7	-4	4	-1	-1	-6	-11	-16	-21	-26	-31	-36	-41	-46	-51	-56	-61	-66	-71
T	-20	-12	-9	-1	2	-3	2	-3	-8	-13	-18	-23	-28	-33	-38	-43	-48	-53	-58	-63
T	-25	-17	-14	-6	-3	0	0	5	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55
G	-30	-22	-19	-11	-8	0	-2	0	3	3	-2	-7	-12	-17	-22	-27	-32	-37	-42	-47
C	-35	-27	-19	-16	-8	-5	-2	-4	-2	1	1	1	-4	-9	-14	-19	-24	-29	-34	-39
A	-40	-32	-24	-21	-13	-10	-7	-4	-1	-4	4	-1	-1	-1	-6	-11	-16	-21	-26	-31
C	-45	-37	-29	-26	-18	-15	-12	-9	-6	-3	-1	7	2	-3	-3	-3	-8	-13	-18	-23
C	-50	-42	-34	-31	-23	-20	-17	-14	-11	-8	-5	2	5	0	-5	0	-5	-5	-10	-15
G	-55	-47	-39	-31	-28	-20	-22	-19	-16	-8	-10	-3	0	3	3	-2	-2	-7	-7	-7
α	=	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	=	G	-	G	-	G	-	T	-	-	-	-	T	-	G	C	A	C	C	G
-7	=	3	-5	3	-5	3	-5	3	-5	-5	-5	-5	3	-5	3	3	3	3	3	3

Exercício

Escreva um algoritmo, em pseudocódigo, que dadas duas sequências α e β , com, respectivamente, m e n caracteres, e uma matriz de pontuação de alinhamentos M , entre todos os pares de prefixos de α e de β , retorne um alinhamento ótimo entre α e β .

Exercício

Calcule a complexidade assintótica (usando a notação Θ) dos algoritmos Memoization e AlignMemoization.

Exercício

Prove ou dê um contraexemplo: se α for muito maior do que β ($|\alpha| \gg |\beta|$), então existe um alinhamento global ótimo entre α e β que não adiciona buracos em α . Considere um esquema de pontuação de alinhamento tal que $\text{gap} \leq \text{mismatch} < 0 < \text{match}$.

Exercício

Dadas duas sequências α e β , com respectivamente m e n caracteres, mostre como determinar em tempo polinomial se o alinhamento ótimo entre α e β é único.

Exercício

Mostre que, para todo $k \in \mathbb{N}$, existem duas sequências α e β , com respectivamente m e n caracteres ($k \leq n \leq m$), tal que o número de alinhamentos ótimos entre α e β é $\Omega(2^k)$.

Exercício

Mostre que, para todo $k \in \mathbb{N}$, existem duas sequências α e β , com respectivamente m e n caracteres ($m, n \in \mathcal{O}(k)$), tal que o número de alinhamentos ótimos entre α e β é $\Omega(2^k)$.

Problema

A distância Levenshtein ou distância de edição entre duas sequências de caracteres é dada pelo número mínimo de operações necessárias para transformar uma sequência na outra. Uma operação é definida como uma inserção, uma remoção ou uma substituição de um caracter. Foi definida pela primeira vez em 1965 por Vladimir Levenshtein.

- A distância de edição tem aplicações em várias áreas:
 - Verificação de erros escrita
 - Reconhecimento de fala
 - Análise de DNA
 - Detecção de plágio

Exemplo

genoma

gnoma (*remoção do “e”*)

gnomo (*substituição do “a” por um “o”*)

gnomos (*inserção do “s”*)

Exercício

Dadas duas sequências α e β com, respectivamente, m e n caracteres, mostre como calcular eficientemente a distância de edição entre elas.

Alinhamento Semi-Global

- Como alterar o algoritmo de Needleman e Wunsch para produzir alinhamentos semi-globais?
 - Não penalizar buracos no começo das sequências.
 - Alterar a inicialização da matriz, atribuindo valor zero para o alinhamento de qualquer prefixo com a subsequência vazia.
 - Não penalizar buracos no final das sequências.
 - Buscar o valor do alinhamento ótimo em todas as posições da última linha ou da última coluna da matriz.

Algoritmo 6: Semi-Global

Input: α, m, β, n

for *all* $i \in [0..n]$ **do**

$M[i, 0] \leftarrow 0$

end

for *all* $j \in [1..m]$ **do**

$M[0, j] \leftarrow 0$

end

for *all* $i \in [1..n]$ **do**

for *all* $j \in [1..m]$ **do**

$$M[i, j] \leftarrow \max \left\{ \begin{array}{l} M[i, j-1] + \text{gap}, \\ M[i-1, j-1] + \sigma(\alpha[j], \beta[i]), \\ M[i-1, j] + \text{gap} \end{array} \right\}$$

end

end

$\text{max} \leftarrow -\infty$

for *all* $i \in [0..n]$ **do**

if $M[i, m] > \text{max}$ **then** $\text{max} \leftarrow M[i, m]$;

end

for *all* $j \in [0..m-1]$ **do**

if $M[n, j] > \text{max}$ **then** $\text{max} \leftarrow M[n, j]$;

end

return max

Alinhamento Semi-Global

$\times$	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0																				
T	0																				
T	0																				
G	0																				
G	0																				
C	0																				
C	0																				
A	0																				
A	0																				
T	0																				
C	0																				
C	0																				
C	0																				
G	0																				
C	0																				

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Semi-Global

$\times$ β	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	-2	-2	3	-2	-2	3	-2	-2	-2	-2	-2	3	-2	3	-2	3	-2	3	-2	-2
T	0	-2	1	-2	6	1	-2	1	1	1	-4	1	-2	1	-2	1	-2	1	-2	6	1
T	0	-2	1	-1	1	9	4	-1	4	4	-1	-1	-1	-4	-1	-4	-1	-4	-1	1	4
G	0	-2	-4	-1	-3	4	7	7	2	2	2	-3	-3	-3	-6	2	-3	-3	-6	-3	-1
G	0	-2	-4	-6	-3	-1	2	10	5	0	0	0	-5	-5	-5	-3	0	-5	-5	-8	-5
C	0	-2	-4	-1	-6	-5	2	5	8	3	-2	-2	3	-2	-2	-7	0	-2	-2	-7	-10
C	0	-2	-4	-1	-3	-8	-2	0	3	6	1	-4	1	1	1	-4	-4	-2	1	-4	-9
A	0	3	-2	-6	-3	-5	-7	-4	-2	1	9	4	-1	4	-1	-1	-6	-1	-4	-1	-1
A	0	3	1	-4	-8	-5	-7	-9	-6	-4	4	7	2	2	2	-3	-3	-3	-3	-6	2
T	0	-2	6	1	-1	-5	-7	-9	-6	-3	-1	7	5	0	0	0	-5	-5	-5	0	-3
C	0	-2	1	9	4	-1	-2	-7	-11	-8	-5	2	10	5	3	-2	3	-2	-2	-5	-2
C	0	-2	-4	4	7	2	2	-3	-8	-13	-10	-3	5	8	8	3	1	1	1	-4	-7
C	0	-2	-4	-1	2	5	5	0	-5	-10	-15	-8	0	3	11	6	6	1	4	-1	-6
G	0	-2	-4	-6	-3	0	3	8	3	-2	-7	-12	-5	-2	6	14	9	4	-1	2	-3
C	0	-2	-4	-1	-6	-5	3	3	6	1	-4	-9	-9	-7	1	9	17	12	7	2	0

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Semi-Global

$\times$ β	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	-2	-2	3	-2	-2	3	-2	-2	-2	-2	-2	3	-2	3	-2	3	-2	3	-2	-2
T	0	-2	1	-2	6	1	-2	1	1	1	-4	1	-2	1	-2	1	-2	1	-2	6	1
T	0	-2	1	-1	1	9	4	-1	4	4	-1	-1	-1	-4	-1	-4	-1	-4	-1	1	4
G	0	-2	-4	-1	-3	4	7	7	2	2	2	-3	-3	-3	-6	2	-3	-3	-6	-3	-1
G	0	-2	-4	-6	-3	-1	2	10	5	0	0	0	-5	-5	-5	-3	0	-5	-5	-8	-5
C	0	-2	-4	-1	-6	-5	2	5	8	3	-2	-2	3	-2	-2	-7	0	-2	-2	-7	-10
C	0	-2	-4	-1	-3	-8	-2	0	3	6	1	-4	1	1	1	-4	-4	-2	1	-4	-9
A	0	3	-2	-6	-3	-5	-7	-4	-2	1	9	4	-1	4	-1	-1	-6	-1	-4	-1	-1
A	0	3	1	-4	-8	-5	-7	-9	-6	-4	4	7	2	2	2	-3	-3	-3	-3	-6	2
T	0	-2	6	1	-1	-5	-7	-9	-6	-3	-1	7	5	0	0	0	-5	-5	-5	0	-3
C	0	-2	1	9	4	-1	-2	-7	-11	-8	-5	2	10	5	3	-2	3	-2	-2	-5	-2
C	0	-2	-4	4	7	2	2	-3	-8	-13	-10	-3	5	8	8	3	1	1	1	-4	-7
C	0	-2	-4	-1	2	5	5	0	-5	-10	-15	-8	0	3	11	6	6	1	4	-1	-6
G	0	-2	-4	-6	-3	0	3	8	3	-2	-7	-12	-5	-2	6	14	9	4	-1	2	-3
C	0	-2	-4	-1	-6	-5	3	3	6	1	-4	-9	-9	-7	1	9	17	12	7	2	0

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Semi-Global

$\times$	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	-2	-2	3	-2	-2	3	-2	-2	-2	-2	-2	3	-2	3	-2	3	-2	3	-2	-2
T	0	-2	1	-2	6	1	-2	1	1	1	-4	1	-2	1	-2	1	-2	1	-2	6	1
T	0	-2	1	-1	1	9	4	-1	4	4	-1	-1	-1	-4	-1	-4	-1	-4	-1	1	4
G	0	-2	-4	-1	-3	4	7	7	2	2	2	-3	-3	-3	-6	2	-3	-3	-6	-3	-1
G	0	-2	-4	-6	-3	-1	2	10	5	0	0	0	-5	-5	-5	-3	0	-5	-5	-8	-5
C	0	-2	-4	-1	-6	-5	2	5	8	3	-2	-2	3	-2	-2	-7	0	-2	-2	-7	-10
C	0	-2	-4	-1	-3	-8	-2	0	3	6	1	-4	1	1	1	-4	-4	-2	1	-4	-9
A	0	3	-2	-6	-3	-5	-7	-4	-2	1	9	4	-1	4	-1	-1	-6	-1	-4	-1	-1
A	0	3	1	-4	-8	-5	-7	-9	-6	-4	4	7	2	2	2	-3	-3	-3	-3	-6	2
T	0	-2	6	1	-1	-5	-7	-9	-6	-3	-1	7	5	0	0	0	-5	-5	-5	0	-3
C	0	-2	1	9	4	-1	-2	-7	-11	-8	-5	2	10	5	3	-2	3	-2	-2	-5	-2
C	0	-2	-4	4	7	2	2	-3	-8	-13	-10	-3	5	8	8	3	1	1	1	-4	-7
C	0	-2	-4	-1	2	5	5	0	-5	-10	-15	-8	0	3	11	6	6	1	4	-1	-6
G	0	-2	-4	-6	-3	0	3	8	3	-2	-7	-12	-5	-2	6	14	9	4	-1	2	-3
C	0	-2	-4	-1	-6	-5	3	3	6	1	-4	-9	-9	-7	1	9	17	12	7	2	0
α																	C	A	C	T	A
β																	C	-	-	-	-

Alinhamento Semi-Global

$\times$	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A	
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
C	0	-2	-2	3	-2	-2	3	-2	-2	-2	-2	-2	3	-2	3	-2	3	-2	3	-2	-2	
T	0	-2	1	-2	6	1	-2	1	1	1	-4	1	-2	1	-2	1	-2	1	-2	6	1	
T	0	-2	1	-1	1	9	4	-1	4	4	-1	-1	-1	-4	-1	-4	-1	-4	-1	1	4	
G	0	-2	-4	-1	-3	4	7	7	2	2	2	-3	-3	-3	-6	2	-3	-3	-6	-3	-1	
G	0	-2	-4	-6	-3	-1	2	10	5	0	0	0	-5	-5	-5	-3	0	-5	-5	-8	-5	
C	0	-2	-4	-1	-6	-5	2	5	8	3	-2	-2	3	-2	-2	-7	0	-2	-2	-7	-10	
C	0	-2	-4	-1	-3	-8	-2	0	3	6	1	-4	1	1	1	-4	-4	-2	1	-4	-9	
A	0	3	-2	-6	-3	-5	-7	-4	-2	1	9	4	-1	4	-1	-1	-6	-1	-4	-1	-1	
A	0	3	1	-4	-8	-5	-7	-9	-6	-4	4	7	2	2	2	-3	-3	-3	-3	-6	2	
T	0	-2	6	1	-1	-5	-7	-9	-6	-3	-1	7	5	0	0	0	-5	-5	-5	0	-3	
C	0	-2	1	9	4	-1	-2	-7	-11	-8	-5	2	10	5	3	-2	3	-2	-2	-5	-2	
C	0	-2	-4	4	7	2	2	-3	-8	-13	-10	-3	5	8	8	3	1	1	1	-4	-7	
C	0	-2	-4	-1	2	5	5	0	-5	-10	-15	-8	0	3	11	6	6	1	4	-1	-6	
G	0	-2	-4	-6	-3	0	3	8	3	-2	-7	-12	-5	-2	6	14	9	4	-1	2	-3	
C	0	-2	-4	-1	-6	-5	3	3	6	1	-4	-9	-9	-7	1	9	17	12	7	2	0	
α														-	A	C	G	C	A	C	T	A
β														C	C	C	G	C	-	-	-	-

Alinhamento Semi-Global

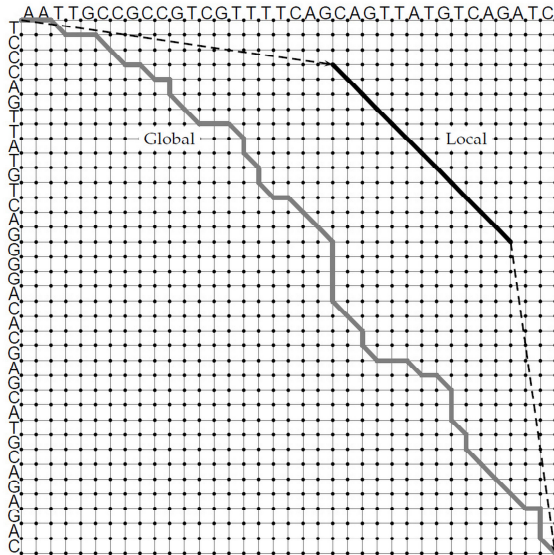
$\times$	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	-2	-2	3	-2	-2	3	-2	-2	-2	-2	-2	3	-2	3	-2	3	-2	3	-2	-2
T	0	-2	1	-2	6	1	-2	1	1	1	-4	1	-2	1	-2	1	-2	1	-2	6	1
T	0	-2	1	-1	1	9	4	-1	4	4	-1	-1	-1	-4	-1	-4	-1	-4	-1	1	4
G	0	-2	-4	-1	-3	4	7	7	2	2	2	-3	-3	-3	-6	2	-3	-3	-6	-3	-1
G	0	-2	-4	-6	-3	-1	2	10	5	0	0	0	-5	-5	-5	-3	0	-5	-5	-8	-5
C	0	-2	-4	-1	-6	-5	2	5	8	3	-2	-2	3	-2	-2	-7	0	-2	-2	-7	-10
C	0	-2	-4	-1	-3	-8	-2	0	3	6	1	-4	1	1	1	-4	-4	-2	1	-4	-9
A	0	3	-2	-6	-3	-5	-7	-4	-2	1	9	4	-1	4	-1	-1	-6	-1	-4	-1	-1
A	0	3	1	-4	-8	-5	-7	-9	-6	-4	4	7	2	2	2	-3	-3	-3	-3	-6	2
T	0	-2	6	1	-1	-5	-7	-9	-6	-3	-1	7	5	0	0	0	-5	-5	-5	0	-3
C	0	-2	1	9	4	-1	-2	-7	-11	-8	-5	2	10	5	3	-2	3	-2	-2	-5	-2
C	0	-2	-4	4	7	2	2	-3	-8	-13	-10	-3	5	8	8	3	1	1	1	-4	-7
C	0	-2	-4	-1	2	5	5	0	-5	-10	-15	-8	0	3	11	6	6	1	4	-1	-6
G	0	-2	-4	-6	-3	0	3	8	3	-2	-7	-12	-5	-2	6	14	9	4	-1	2	-3
C	0	-2	-4	-1	-6	-5	3	3	6	1	-4	-9	-9	-7	1	9	17	12	7	2	0
α		C	T	T	C	G	T	T	A	-	T	C	A	C	G	C	A	C	T	A	
β		C	T	T	G	G	C	C	A	A	T	C	C	C	G	C	-	-	-	-	

Alinhamento Semi-Global

$\times$	α	A	T	C	T	T	C	G	T	T	A	T	C	A	C	G	C	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	-2	-2	3	-2	-2	3	-2	-2	-2	-2	-2	3	-2	3	-2	3	-2	3	-2	-2
T	0	-2	1	-2	6	1	-2	1	1	1	-4	1	-2	1	-2	1	-2	1	-2	6	1
T	0	-2	1	-1	1	9	4	-1	4	4	-1	-1	-1	-4	-1	-4	-1	-4	-1	1	4
G	0	-2	-4	-1	-3	4	7	7	2	2	2	-3	-3	-3	-6	2	-3	-3	-6	-3	-1
G	0	-2	-4	-6	-3	-1	2	10	5	0	0	0	-5	-5	-5	-3	0	-5	-5	-8	-5
C	0	-2	-4	-1	-6	-5	2	5	8	3	-2	-2	3	-2	-2	-7	0	-2	-2	-7	-10
C	0	-2	-4	-1	-3	-8	-2	0	3	6	1	-4	1	1	1	-4	-4	-2	1	-4	-9
A	0	3	-2	-6	-3	-5	-7	-4	-2	1	9	4	-1	4	-1	-1	-6	-1	-4	-1	-1
A	0	3	1	-4	-8	-5	-7	-9	-6	-4	4	7	2	2	2	-3	-3	-3	-3	-6	2
T	0	-2	6	1	-1	-5	-7	-9	-6	-3	-1	7	5	0	0	0	-5	-5	-5	0	-3
C	0	-2	1	9	4	-1	-2	-7	-11	-8	-5	2	10	5	3	-2	3	-2	-2	-5	-2
C	0	-2	-4	4	7	2	2	-3	-8	-13	-10	-3	5	8	8	3	1	1	1	-4	-7
C	0	-2	-4	-1	2	5	5	0	-5	-10	-15	-8	0	3	11	6	6	1	4	-1	-6
G	0	-2	-4	-6	-3	0	3	8	3	-2	-7	-12	-5	-2	6	14	9	4	-1	2	-3
C	0	-2	-4	-1	-6	-5	3	3	6	1	-4	-9	-9	-7	1	9	17	12	7	2	0
α	A	T	C	T	T	C	G	T	T	A	-	T	C	A	C	G	C	A	C	T	A
β	-	-	C	T	T	G	G	C	C	A	A	T	C	C	C	G	C	-	-	-	-

Alinhamento Local

Alinhamento Global x Alinhamento Local



Alinhamento Local - Abordagem Ingênua

- O alinhamento local ótimo entre duas sequências α e β é o melhor alinhamento (global) entre uma subsequência de α e uma subsequência de β .
- É possível usar o algoritmo de alinhamento global de Needleman e Wunsch para produzir alinhamentos locais?
- Quantas subsequências (não nulas) as sequências α e β , formadas respectivamente por m e n caracteres, possuem?
 - Respectivamente, $m(m+1)/2 = \Theta(m^2)$ e $n(n+1)/2 = \Theta(n^2)$ subsequências.
- Algoritmo ingênuo: testar todos os pares de subsequências e escolher aquele para com o maior valor de alinhamento global.
- Qual a complexidade do algoritmo ingênuo para alinhamento local?

Análise de Complexidade

- Complexidade:

$$T(m, n) = \sum_{\alpha'=1}^m \sum_{\alpha''=\alpha'}^m \sum_{\beta'=1}^n \sum_{\beta''=\beta'}^n \Theta((\alpha'' - \alpha' + 1)(\beta'' - \beta' + 1))$$

- Limite superior:

$$T(m, n) \leq \frac{m(m+1)}{2} \frac{n(n+1)}{2} O(mn) = O(m^3 n^3)$$

- Limite inferior:

$$T(m, n) \geq \sum_{\alpha'=1}^{\lfloor \frac{m}{3} \rfloor} \sum_{\alpha''=m-\lfloor \frac{m}{3} \rfloor+1}^m \sum_{\beta'=1}^{\lfloor \frac{n}{3} \rfloor} \sum_{\beta''=n-\lfloor \frac{n}{3} \rfloor+1}^n \Omega(\lceil \frac{m}{3} \rceil \lceil \frac{n}{3} \rceil)$$

$$T(m, n) \geq \lfloor \frac{m}{3} \rfloor \lfloor \frac{m}{3} \rfloor \lfloor \frac{n}{3} \rfloor \lfloor \frac{n}{3} \rfloor \Omega(\lceil \frac{m}{3} \rceil \lceil \frac{n}{3} \rceil) = \Omega(m^3 n^3)$$

- Complexidade:
 - $T(m, n) = \Theta(m^3 n^3)$
- Exemplo: computador de 3GHz (1 instrução por ciclo)
 - $m = n = 10$:
 - Tempo: $10^6 / (3 \times 2^{30}) = 310$ bilionésimos de segundo
 - $m = n = 100$:
 - Tempo: $100^6 / (3 \times 2^{30}) = 310$ segundos
 - $m = n = 1000$:
 - Tempo: $1000^6 / (3 \times 2^{30}) = 10$ anos
 - $m = n = 10000$:
 - Tempo: $10000^6 / (3 \times 2^{30}) = 10$ bilhões de anos

Alinhamento Local

- Como alterar o algoritmo de Needleman e Wunsch para produzir alinhamentos locais?
 - Redefinir o significado das células da matriz M .
 - A célula $M[i,j]$ deve representar o valor do alinhamento ótimo entre todos os sufixos das sequências $\alpha[1..j]$ e $\beta[1..i]$.
 - Qual o valor mínimo que uma célula pode armazenar?
 - O valor correspondente ao alinhamento de um sufixo vazio com qualquer outro, ou seja, zero.
 - Qual o valor do alinhamento local ótimo entre as duas sequências?
 - O maior valor armazenado em qualquer uma das posições da matriz.
 - Como recuperar o alinhamento local ótimo?
 - O alinhamento deve começar numa célula com valor zero e terminar na célula de maior valor da matriz (obter alinhamento na ordem inversa).
- Em 1981, Temple Smith e Michael Waterman propuseram o algoritmo de programação dinâmica para alinhamento local de sequências.

Algoritmo 7: Local

Input: α, m, β, n

for *all* $i \in [0..n]$ **do**

$M[i, 0] \leftarrow 0$

end

for *all* $j \in [1..m]$ **do**

$M[0, j] \leftarrow 0$

end

for *all* $i \in [1..n]$ **do**

for *all* $j \in [1..m]$ **do**

$$M[i, j] \leftarrow \max \left\{ \begin{array}{l} 0, \\ M[i, j-1] + gap, \\ M[i-1, j-1] + \sigma(\alpha[j], \beta[j]), \\ M[i-1, j] + gap \end{array} \right\}$$

end

end

$max \leftarrow 0$

for *all* $i \in [1..n]$ **do**

for *all* $j \in [1..m]$ **do**

if $M[i, j] > max$ **then** $max \leftarrow M[i, j];$

end

end

return max

Alinhamento Local

$\times$	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0																				
T	0																				
T	0																				
G	0																				
G	0																				
C	0																				
C	0																				
A	0																				
C	0																				
T	0																				
C	0																				
C	0																				
C	0																				
G	0																				
C	0																				

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Local

$\times$	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Local

$\times \backslash \alpha$		G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1

Gap = -5

Match = +3

Mismatch = -2

Alinhamento Local

× β	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1

α

β

C

C

Alinhamento Local

× β	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1

α

β

C

G

C

C

Alinhamento Local

× β	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	3	0	0	3	0	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1
		T C C T G C																			

Alinhamento Local

$\times$	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1
	α											C	T	C	C						
	β											C	T	G	C						

Alinhamento Local

× β	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1
		α β																			
		A A																			
		C C																			
		T T																			
		C G																			

Alinhamento Local

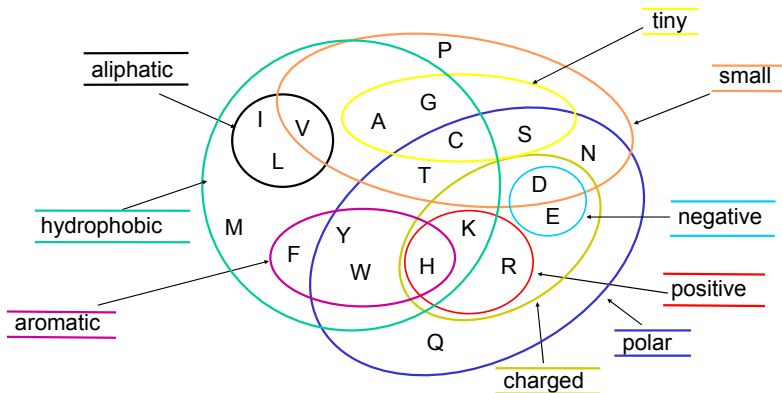
$\times$	α	G	A	C	A	A	C	G	T	T	A	C	T	G	C	T	T	A	C	T	A
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	3	0	0	3	0	0	0	0	3	0	0	3	0	0	0	3	0	0
T	0	0	0	0	1	0	0	1	3	3	0	0	6	1	0	6	3	0	0	6	1
T	0	0	0	0	0	0	0	0	4	6	1	0	3	4	0	3	9	4	0	3	4
G	0	3	0	0	0	0	0	3	0	2	4	0	0	6	2	0	4	7	2	0	1
G	0	3	1	0	0	0	0	3	1	0	0	2	0	3	4	0	0	2	5	0	0
C	0	0	1	4	0	0	3	0	1	0	0	3	0	0	6	2	0	0	5	3	0
C	0	0	0	4	2	0	3	1	0	0	0	3	1	0	3	4	0	0	3	3	1
A	0	0	3	0	7	5	0	1	0	0	3	0	1	0	0	1	2	3	0	1	6
C	0	0	0	6	2	5	8	3	0	0	0	6	1	0	3	0	0	0	6	1	1
T	0	0	0	1	4	0	3	6	6	3	0	1	9	4	0	6	3	0	1	9	4
C	0	0	0	3	0	2	3	1	4	4	1	3	4	7	7	2	4	1	3	4	7
C	0	0	0	3	1	0	5	1	0	2	2	4	1	2	10	5	0	2	4	1	2
C	0	0	0	3	1	0	3	3	0	0	0	5	2	0	5	8	3	0	5	2	0
G	0	3	0	0	1	0	0	6	1	0	0	0	3	5	0	3	6	1	0	3	0
C	0	0	1	3	0	0	3	1	4	0	0	3	0	1	8	3	1	4	4	0	1
											A	C	T	C	C						
											A	C	T	G	C						

Pontuação para Alinhamento de Caracteres

Pontuação para Alinhamento de Caracteres

- Simples (para nucleotídeos):
 - Match: alinhamento de dois caracteres iguais
 - Mismatch: alinhamento de dois caracteres diferentes
- Matrizes de Substituições (para aminoácidos):
 - Família PAM
 - Família BLOSUM

Propriedades dos Aminoácidos



- Desenvolvida por Margaret Dayhoff (1978).
- PAM: *Percent (or Point) of Accepted Mutations*.
- Distância 1-PAM: uma mutação (substituição) a cada 100 aminoácidos em média.
- Duas proteínas com distância k -PAM não necessariamente possuem $k\%$ de diferença entre seus aminoácidos.
- A similaridade esperada entre duas sequências protéicas com distância 200-PAM é de cerca de 25%.

- Seja $F^k(a_i, a_j)$ a probabilidade de um aminoácido a_i mutar para o aminoácido a_j em sequências com distância k -PAM e seja $freq(a_j)$ a frequência do aminoácido a_j .
- Fórmula geral para matrizes PAM:

$$PAM_k(a_i, a_j) = \left\lceil 10 \times \log \frac{F^k(a_i, a_j)}{freq(a_j)} \right\rceil$$

- Valores mais utilizados: $k = 40$, $k = 120$ e $k = 250$.

- Desenvolvida por Steven Henikoff e Jorja Henikoff (1992).
- BLOSUM: *BLOcks of Amino Acid SUBstitution Matrix*
- BLOCKS: banco de dados de alinhamentos múltiplos de blocos conservados (sem gaps) de 504 grupos de proteínas.
- $BLOSUM_k$ considera apenas sequências com uma similaridade maior do que $k\%$ entre suas bases.

- Seja $freq_k(a_i, a_j)$ a frequência de alinhamentos de aminoácidos a_i com aminoácidos a_j e seja $freq_k(a_i)$ a frequência do aminoácido a_i , em ambos os casos, em sequências com similaridade maior do que $k\%$ entre suas bases (considerando o banco de dados BLOCKS).
- Fórmula geral para matrizes BLOSUM:

$$BLOSUM_k(a_i, a_j) = \left\lceil \log_2 \frac{freq_k(a_i, a_j)}{freq_k(a_i) \times freq_k(a_j)} \right\rceil$$

- $BLOSUM_{62}$: matriz padrão para alinhamento de proteínas (BLAST).

BLOCKS

A	A	B	C	D	A	-	-	-	B	B	C	D	A
-	A	B	C	D	A	-	A	-	B	B	C	B	B
B	B	B	C	D	A	B	A	-	B	C	C	A	A
A	A	A	C	D	A	C	-	D	C	B	C	D	-
C	C	B	A	D	A	B	-	D	B	B	D	C	C
A	A	A	C	A	A	-	-	-	B	B	C	C	C

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W

- A família PAM assume um modelo evolucionário baseado em árvores filogenéticas. A família BLOSUM não assume nenhum modelo evolucionário, mas considera blocos conservados de proteínas.
- A Entropia Relativa (H) de uma matriz de substituição é dada por:

$$H = \sum_{i=1}^{20} \sum_{j=1}^i \text{freq}(a_i) \text{freq}(a_j) \sigma(a_i, a_j)$$

- Equivalência entre matrizes (com base em sua Entropia Relativa):

$$PAM_{100} \iff BLOSUM_{90} \mid PAM_{200} \iff BLOSUM_{52}$$

$$PAM_{120} \iff BLOSUM_{80} \mid PAM_{250} \iff BLOSUM_{45}$$

$$PAM_{160} \iff BLOSUM_{60} \mid PAM_{400} \iff BLOSUM_{30}$$

Erro no Cálculo das Matrizes da Família BLOSUM

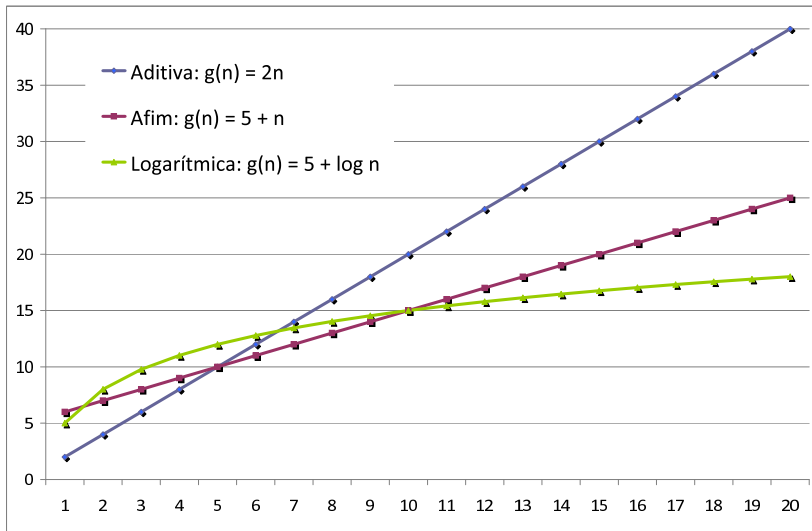
- Em 2008, Mark Styczynski e Kyle Jensen mostraram que as matrizes BLOSUM, comumente utilizadas na prática, foram calculadas de forma errada, devido a um erro de software.
- Este erro, que afeta cerca de 15% das posições das matrizes BLOSUM, passou despercebido por mais de 15 anos.
- Teste computacionais, realizados pela mesma dupla de pesquisadores, mostraram que as matrizes “erradas” apresentam melhores resultados na prática do que as matrizes “corretas”.

Penalidades para Blocos de Buracos

Penalidades para Blocos de Buracos

- Ao invés de penalizar buracos individualmente, podemos penalizar blocos de buracos consecutivos num alinhamento.
- Existem várias opções de funções de penalização de blocos de buracos.
- Funções Aditivas: $g(k_1 + k_2) = g(k_1) + g(k_2)$
 - Linear: $g(k) = a \times k$
- Funções Sub-Aditivas: $g(k_1 + k_2) < g(k_1) + g(k_2)$
 - Afim: $g(k) = a + b \times k$
 - Funções Convexas: $g(k) - g(k - 1) > g(k + 1) - g(k)$
 - Logarítmica: $g(k) = a + b \times \log k$

Exemplos de Funções para Penalidades de Buracos



Penalidades para Blocos de Buracos

- Exemplos de diferenças entre alinhamentos com mesma matriz de pontuação de alinhamento de bases, mas com penalidades diferentes para blocos de buracos.
- Alinhamento global ótimo com pontuação aditiva:

```
G C G C G T T A G A C T A G C A C C G
G - G - G T T - G - C - A - C - C - G
```

Penalidades para Blocos de Buracos

- Exemplos de diferenças entre alinhamentos com mesma matriz de pontuação de alinhamento de bases, mas com penalidades diferentes para blocos de buracos.
- Alinhamento global ótimo com pontuação sub-aditiva:

G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
-	-	G	C	G	T	T	-	-	-	-	-	-	G	C	A	C	C	G

Função Genérica de Penalidade para Buracos

- *Importante:* inserir um novo buraco em um alinhamento tem custo diferente caso o alinhamento corrente termine num buraco ou não.
- Para preencher a célula $M[i, j]$ devemos considerar 3 opções:
 - Valor do alinhamento ótimo representado pela célula $M[i - 1, j - 1]$ acrescido do alinhamento dos caracteres $\alpha[j]$ e $\beta[i]$.
 - Valor do alinhamento ótimo representado por uma célula $M[i, j - k]$, acrescido de um alinhamento entre $\alpha[j - k + 1..j]$ e um bloco formado por k buracos (para $1 \leq k \leq j$).
 - Valor do alinhamento ótimo representado por uma célula $M[i - k, j]$, acrescido de um alinhamento entre $\beta[i - k + 1..i]$ e um bloco formado por k buracos (para $1 \leq k \leq i$).
- Algoritmo proposto por Michael Waterman, Temple Smith e William Beyer em 1976.

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6							*				
C	7											
A	8											
C	9											

$$M[6,6] + \sigma(G,C)$$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7						*					
A	8											
C	9											

$k = 1$

$M[7,6] + g(1)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7					*						
A	8											
C	9											

$k = 2$

$M[7,5] + g(2)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7				*							
A	8											
C	9											

$k = 3$

$M[7,4] + g(3)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7			*								
A	8											
C	9											

$k = 4$

$M[7,3] + g(4)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7		*									
A	8											
C	9											

$k = 5$

$M[7,2] + g(5)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7	*										
A	8											
C	9											

$k = 6$

$M[7,1] + g(6)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	*											
A	8											
C	9											

$k = 7$

$M[7,0] + g(7)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7							*				
A	8											
C	9											

$k = 1$

$M[6,7] + g(1)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5							*				
C	6											
C	7											
A	8											
C	9											

$k = 2$

$M[5,7] + g(2)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3											
G	4							*				
G	5											
C	6											
C	7											
A	8											
C	9											

$k = 3$

$M[4,7] + g(3)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2											
T	3							*				
G	4											
G	5											
C	6											
C	7											
A	8											
C	9											

$k = 4$

$M[3,7] + g(4)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1											
T	2							*				
T	3											
G	4											
G	5											
C	6											
C	7											
A	8											
C	9											

$k = 5$

$M[2,7] + g(5)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	7	8	9	10	11
A	1							*				
T	2											
T	3											
G	4											
G	5											
C	6											
C	7											
A	8											
C	9											

$k = 6$

$M[1,7] + g(6)$

Função Genérica de Penalidade para Buracos

$\times$ β	α	C	C	T	G	T	G	G	C	A	A	C
	0	1	2	3	4	5	6	*	8	9	10	11
A	1											
T	2											
T	3											
G	4											
G	5											
C	6											
C	7											
A	8											
C	9											

$k = 7$

$M[0,7] + g(7)$

Função Genérica de Penalidade para Buracos

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24
G	-6	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20
G	-7	-3	1	0	-6	-2	-8	-9	-10	-6	-12	-13	-14	-15	-11	-17	-18	-19	-20	-16
G	-8	-4	-5	4	-2	-3	-9													
T	-9																			
T	-10																			
G	-11																			
C	-12																			
A	-13																			
C	-14																			
C	-15																			
G	-16																			
α	=	G	C	G	C	G	T													
β	=	G	G	G	-	-	-													
-9	=	3	-2	3	-6	-1	-6													

Open Gap = -5

Match = +3

Extend Gap = -1

Mismatch = -2

Função Genérica de Penalidade para Buracos

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24
G	-6	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20
G	-7	-3	1	0	-6	-2	-8	-9	-10	-6	-12	-13	-14	-15	-11	-17	-18	-19	-20	-16
G	-8	-4	-5	4	-2	-3	-9													
T	-9																			
T	-10																			
G	-11																			
C	-12																			
A	-13																			
C	-14																			
C	-15																			
G	-16																			
α	=	G	C	G	C	G	T													
β	=	G	G	G	-	-	-													
-9	=	3	-2	3	-6	-6	-1													
										Open Gap = -5					Match = +3					
										Extend Gap = -1					Mismatch = -2					

Função Genérica de Penalidade para Buracos

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24
G	-6	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20
G	-7	-3	1	0	-6	-2	-8	-9	-10	-6	-12	-13	-14	-15	-11	-17	-18	-19	-20	-16
G	-8	-4	-5	4	-2	-3	-4													
T	-9																			
T	-10																			
G	-11																			
C	-12																			
A	-13																			
C	-14																			
C	-15																			
G	-16																			
α	=	G	C	G	C	G	T													
β	=	G	G	G	-	-	-													
-4	=	3	-2	3	-6	-1	-1													

Open Gap = -5

Match = +3

Extend Gap = -1

Mismatch = -2

Função Genérica de Penalidade para Buracos

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24
G	-6	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20
G	-7	-3	1	0	-6	-2	-8	-9	-10	-6	-12	-13	-14	-15	-11	-17	-18	-19	-20	-16
G	-8	-4	-5	4	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17
T	-9	-5	-6	-2	2	-4	0	-1	-7	-8	-9	-10	-6	-12	-13	-14	-15	-16	-17	-18
T	-10	-6	-7	-3	-4	0	-1	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14
G	-11	-7	-8	-4	-5	-1	-2	-3	1	0	-6	-7	-8	-9	-5	-11	-12	-13	-14	-10
C	-12	-8	-4	-5	-1	-7	-3	-4	-5	-1	-2	-3	-9	-10	-11	-2	-8	-9	-10	-11
A	-13	-9	-10	-6	-7	-3	-9	-5	-1	-7	2	-4	-5	-6	-7	-8	1	-5	-6	-7
C	-14	-10	-6	-7	-3	-9	-5	-6	-7	-3	-4	5	-1	-2	-3	-4	-5	4	-2	-3
C	-15	-11	-7	-8	-4	-5	-11	-7	-8	-9	-5	-1	3	-3	-4	0	-6	-2	7	1
G	-16	-12	-13	-4	-10	-1	-7	-8	-9	-5	-6	-2	-3	1	0	-6	-2	-3	1	10

Open Gap = -5

Extend Gap = -1

Match = +3

Mismatch = -2

Função Genérica de Penalidade para Buracos

$\times$	α	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	0	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24
G	-6	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17	-18	-19	-20
G	-7	-3	1	0	-6	-2	-8	-9	-10	-6	-12	-13	-14	-15	-11	-17	-18	-19	-20	-16
G	-8	-4	-5	4	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17
T	-9	-5	-6	-2	2	-4	0	-1	-7	-8	-9	-10	-6	-12	-13	-14	-15	-16	-17	-18
T	-10	-6	-7	-3	-4	0	-1	3	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14
G	-11	-7	-8	-4	-5	-1	-2	-3	1	0	-6	-7	-8	-9	-5	-11	-12	-13	-14	-10
C	-12	-8	-4	-5	-1	-7	-3	-4	-5	-1	-2	-3	-9	-10	-11	-2	-8	-9	-10	-11
A	-13	-9	-10	-6	-7	-3	-9	-5	-1	-7	2	-4	-5	-6	-7	-8	1	-5	-6	-7
C	-14	-10	-6	-7	-3	-9	-5	-6	-7	-3	-4	5	-1	-2	-3	-4	-5	4	-2	-3
C	-15	-11	-7	-8	-4	-5	-11	-7	-8	-9	-5	-1	3	-3	-4	0	-6	-2	7	1
G	-16	-12	-13	-4	-10	-1	-7	-8	-9	-5	-6	-2	-3	1	0	-6	-2	-3	1	10
α	=	G	C	G	C	G	T	T	A	G	A	C	T	A	G	C	A	C	C	G
β	=	-	-	G	G	G	T	T	-	-	-	-	-	-	G	C	A	C	C	G
10	=	-6	-1	3	-2	3	3	3	-6	-1	-1	-1	-1	-1	3	3	3	3	3	3

Função Genérica de Penalidade para Buracos

Algoritmo 8: Global

Input: α, m, β, n

$M[0, 0] \leftarrow 0$

for all $i \in [1..n]$ **do** $M[i, 0] \leftarrow g(i)$;

for all $j \in [1..m]$ **do** $M[0, j] \leftarrow g(j)$;

for all $i \in [1..n]$ **do**

for all $j \in [1..m]$ **do**

$\max \leftarrow M[i-1, j-1] + \sigma(\alpha[j], \beta[j])$

for all $k \in [1..j]$ **do**

if $\max < M[i, j-k] + g(k)$ **then** $\max \leftarrow M[i, j-k] + g(k)$;

end

for all $k \in [1..i]$ **do**

if $\max < M[i-k, j] + g(k)$ **then** $\max \leftarrow M[i-k, j] + g(k)$;

end

$M[i, j] \leftarrow \max$

end

end

return $M[n, m]$

- Complexidade:

$$T(m, n) = \sum_{i=1}^m \sum_{j=1}^n \Theta(i + j + 1)$$

- Limite superior:

$$T(m, n) \leq mnO(m + n + 1) = O(m^2n + mn^2)$$

- Limite inferior:

$$T(m, n) \geq \sum_{i=\lfloor \frac{m}{2} \rfloor + 1}^m \sum_{j=\lfloor \frac{n}{2} \rfloor + 1}^n \Omega(i + j + 1)$$

$$T(m, n) \geq \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor \Omega\left(\left\lceil \frac{m}{2} \right\rceil + \left\lceil \frac{n}{2} \right\rceil\right) = \Omega(m^2n + mn^2)$$

Complexidade e Tempo de Execução

- Complexidade:
 - $T(m, n) = \Theta(mn(m + n)) = \Theta(m^2n + mn^2)$
- Exemplo: computador de 3GHz (1 instrução por ciclo)
 - $m = n = 10$:
 - Tempo: $(10 \times 10 \times 20)/(3 \times 2^{30}) = 31$ bilionésimos de segundo
 - $m = n = 100$:
 - Tempo: $(100 \times 100 \times 200)/(3 \times 2^{30}) = 621$ milionésimos de segundo
 - $m = n = 1000$:
 - Tempo: $(1000 \times 1000) \times 2000/(3 \times 2^{30}) = 621$ milésimos de segundo
 - $m = n = 10000$:
 - Tempo: $(10000 \times 10000) \times 20000/(3 \times 2^{30}) = 10$ minutos
 - $m = n = 100000$:
 - Tempo: $(100000 \times 100000 \times 200000)/(3 \times 2^{30}) = 7$ dias
 - $m = n = 1000000$:
 - Tempo: $(1000000 \times 1000000 \times 2000000)/(3 \times 2^{30}) = 19$ anos