TEMA Tend. Mat. Apl. Comput., **8**, No. 3 (2007), 319-328.

Complexidade de Alinhamento de Seqüências Biológicas

R.T. BRITO[1], Faculdade de Computação e Informática, Universidade Presbiteriana Mackenzie, Rua da Consolação, 930, 01302-907 São Paulo, SP, Brasil.

Resumo. Neste trabalho, apresentamos uma demonstração simples, completa e acessível (inclusive a alunos de graduação de Ciência da Computação) do fato de que o problema de Alinhamento de Seqüências Biológicas é NP-difícil, baseada na demonstração de Wang e Jiang [10], um resultado freqüentemente citado, mas com a prova normalmente omitida.

1. Introdução

A maneira típica para comparar seqüências biológicas (que é uma das mais corriqueiras tarefas em Biologia Computacional) é construir, a partir das seqüências de interesse, um alinhamento de tais seqüências.

Intuitivamente, um alinhamento é uma forma de inserir espaços nas seqüências de maneira que elas fiquem todas com o mesmo comprimento (vide Figura 1).

```
TAGGTCA
TAGCT␣A
```

Figura 1: Alinhamento entre `TAGGTCA e TAGCTA.`

Alinhamentos de interesse são aqueles que exibem, de maneira explícita, quais foram as regiões preservadas ou alteradas durante o processo evolutivo (e.g., via mutação) e, para isso, costuma-se atribuir pontuações a alinhamentos e a tratar o problema de construir um alinhamento entre várias seqüências como um problema de otimização.[2]

As aplicações de alinhamentos são inúmeras [7] em Biologia Computacional, variando desde a simples observação de regiões mais propensas à mutação ou à estabilidade até servir como ponto de partida para a construção de Árvores Filogenéticas ("Árvores Evolutivas"), que é outro pilar da Biologia Computacional.

Alinhamentos são, portanto, de inestimável importância e, freqüentemente, os alinhamentos necessários para biólogos consistem de várias seqüências.

[1]rbrito@{mackenzie,ime.usp}.br. Durante a elaboração deste trabalho, o autor recebeu apoio financeiro do CNPq.

[2]Para efeitos práticos, é costumeiro tratar o problema como um problema de maximização, enquanto, para efeitos teóricos, é usual tratar o problema como um problema de minimização.

Infelizmente, não se conhece um algoritmo que seja *eficiente* para construir alinhamentos ótimos de várias seqüências. Os algoritmos conhecidos para resolver o problema de otimização de encontrar um melhor alinhamento de k seqüências de tamanho n cada têm complexidade de tempo e de espaço $\Omega(n^k)$, que é insatisfatório do ponto de vista prático [4, 6].

Toda a dificuldade em obter algoritmos rápidos para encontrar alinhamentos ótimos obviamente nos conduz à pergunta de se é possível, de fato, desenvolver algoritmos que sejam rápidos (mais precisamente, de tempo de execução polinomial no tamanho da entrada) para o problema de alinhar várias seqüências (doravante denotado por Problema AVS). Uma questão um pouco mais ambiciosa é descobrir não apenas se é ou não possível projetar um algoritmo de tempo polinomial para o problema, mas de descobrir quais são os recursos de tempo necessários para um algoritmo qualquer que o resolva.

Embora não saibamos responder às perguntas acima de maneira direta (à semelhança de muitos outros casos, conforme exposto por Garey e Johnson [5, Capítulo 1]), pode-se mostrar que para uma ampla classe de instâncias, o Problema AVS é pelo menos tão difícil, em termos de complexidade de tempo, quanto outros problemas combinatórios, no sentido de que se houver um algoritmo que resolva o Problema AVS em tempo polinomial para qualquer entrada, então cada um dos problemas de uma grande classe, a classe dos *Problemas NP*, também admitirá um algoritmo de tempo polinomial.

A primeira demonstração de dificuldade (no sentido de NP-completude) do Problema AVS foi publicada em 1994 por Wang e Jiang [10]. Posteriormente, outras demonstrações de que o Problema AVS (em sua versão de decisão) é NP-completo surgiram [1, 8]. Para uma demonstração detalhada desses fatos mais recentes, remetemos o leitor às referências bibliográficas [2].

O objetivo de nosso presente trabalho é exibir uma demonstração derivada daquela proposta por Wang e Jiang, mas que seja detalhada e elementar. Por utilizar apenas técnicas elementares de contagem, supomos que ela seja acessível a alunos de graduação de Ciência da Computação.

Para nossa discussão, supomos que o leitor esteja familiarizado com conceitos básicos de Teoria de Complexidade de Algoritmos. A notação adotada é padrão e pode ser encontrada em diversos livros-texto comuns sobre algoritmos [3, 5]. Introduções ao Problema AVS podem ser encontradas em vários artigos ou livros-texto [2, 4, 7].

2. O Problema AVS é NP-difícil (Wang e Jiang)

O Problema AVS, conforme mencionado na Introdução, é normalmente formulado como um problema de otimização [4, 6, 7, 8].

Como pré-requisito para a função objetivo, é usada uma matriz c que a cada possível par de caracteres atribui um número racional e, para uma coluna, do alinhamento, atribui, como pontuação daquela coluna, a soma das pontuações de todos os pares de caracteres daquela coluna (a chamada *Pontuação* SP ou *soma-de-pares*):

Definição 2.1 (Pontuação SP de uma Coluna)**.** *Fixada uma matriz de pontuação*

de pares de caracteres c, definimos a função de pontuação SP *que mapeia uma coluna* $\mathcal{C}$ *com k caracteres à sua pontuação* $\mathrm{SP}(\mathcal{C})$ *por*

$$\mathrm{SP}(\mathcal{C}) = \sum_{1 \leq i < i' \leq k} c(\mathcal{C}[i], \mathcal{C}[i']),$$

onde $\mathcal{C}[i]$ *denota o i-ésimo caractere da coluna* $\mathcal{C}$*.*

Para um alinhamento A, sua Pontuação SP é definida como a soma das pontuações de cada coluna e a formulação do Problema AVS é, normalmente, a seguinte:

Definição 2.2 (Pontuação SP de um Alinhamento)**.** *A pontuação* $\mathrm{SP}(A)$ *de um alinhamento A das seqüências* $s_1, \ldots, s_k$ *é definida por*

$$\mathrm{SP}(A) = \sum_{j=1}^{l} \mathrm{SP}(A[\cdot, j]) = \sum_{j=1}^{l} \sum_{1 \leq i < i' \leq k} c(A[i, j], A[i', j]), \tag{2.1}$$

onde $A[\cdot, j]$ *denota a j-ésima coluna de A e l é o número de colunas de A.*

Para simplificar a notação, vamos confundir a matriz de pontuação de caracteres c com a pontuação SP determinada por ela e, com isso em mente, podemos enunciar o Problema AVS:

Problema 2.1 (Problema AVS)**.** *Dados um inteiro* $k \geq 2$ *e k seqüências* $s_1, \ldots, s_k$ *sobre um alfabeto* Σ *fixado e fixada uma função de pontuação c, encontrar um alinhamento A cujo custo* $c(A)$ *seja igual a* $c(s_1, \ldots, s_k)$*.*

A demonstração de Wang e Jiang de que o Problema AVS é NP-difícil considera a versão de decisão do problema e mostra, efetivamente, que a versão de decisão do Problema AVS é NP-completa e que, portanto, a versão de otimização é NP-difícil. Formalmente, a versão de decisão do Problema AVS é a seguinte:

Problema 2.2 (Problema AVS, versão de decisão)**.** *Dadas k seqüências* $s_1, \ldots, s_k$ *e dado um inteiro C, decidir se existe um alinhamento das seqüências com custo menor ou igual a C.*

A demonstração dada por Wang e Jiang é feita reduzindo-se o Problema da Superseqüência Comum de Menor Comprimento (em inglês, *Shortest Common Supersequence*, que abreviamos aqui por SC-MÍN) ao Problema AVS.

Para estabelecermos formalmente o Problema SC-MÍN, precisamos de uma definição. Dadas k seqüências $s_1, \ldots, s_k$ sobre um alfabeto Σ, dizemos que uma seqüência $s \in \Sigma^*$ é uma *superseqüência* de $s_1, \ldots, s_k$ se, para cada cada seqüência $s_i = s_i[1] \cdots s_i[n_i]$, existirem palavras $w_0, \ldots, w_{n_i} \in \Sigma^*$ tais que s seja da forma $s = w_0 s_i[1] w_1 \ldots s_i[n_i] w_{n_i}$. Aqui, o comprimento da seqüência s_i é $n_i = |s_i|$, para $i = 1, \ldots, k$. Em palavras, esta definição significa que s é uma superseqüência de $s_1, \ldots, s_k$ se s contiver cada s_i (onde, possivelmente, os caracteres de s_i estão espaçados em s por caracteres do alfabeto).

O Problema da Superseqüência Comum de Menor Comprimento pode ser enunciado como:

Problema 2.3 (Problema SC-MÍN, versão de decisão)**.** *Dadas k seqüências $s_1, \ldots, s_k$ e dado um inteiro L, decidir se existe uma superseqüência s de $s_1, \ldots, s_k$ com comprimento no máximo L.*

O Problema SC-MÍN é um problema NP-completo [5] e, para mostrar que o Problema AVS também é NP-completo, basta mostrar uma redução de tempo polinomial do Problema SC-MÍN ao Problema AVS, uma vez que, como SC-MÍN é NP-completo, todo problema da classe NP reduz-se polinomialmente também a AVS.

Antes de darmos a redução, observemos que:

- se $L < \max\{n_i\}$, a resposta ao SC-MÍN é NÃO, uma vez que toda superseqüência s tem comprimento no mínimo igual ao comprimento das seqüências s_i, isto é, não existe superseqüência de $s_1, \ldots, s_k$ de comprimento menor que o comprimento da seqüência mais longa;

- se $L \geq \sum n_i$, a resposta ao SC-MÍN é SIM, uma vez que a seqüência $s = s_1 s_2 \cdots s_k$, igual à concatenação das seqüências de entrada, é, naturalmente, uma superseqüência das seqüências de entrada e seu comprimento é menor ou igual a L.

Assim, para a redução, podemos supor que o inteiro L é tal que $\max\{n_i\} \leq L \leq \sum n_i$. Essa hipótese é necessária para que a redução possa ser feita em tempo polinomial. É fácil ver que um algoritmo de redução pode decidir se essa hipótese a respeito de L é válida ou não em tempo linear no tamanho da entrada.

A redução propriamente dita é feita tomando-se uma instância do Problema SC-MÍN com k seqüências $S = \{s_1, \ldots, s_k\}$ sobre o alfabeto $\{0, 1\}$ e um inteiro L (de acordo com a hipótese acima) e construindo-se $L+1$ instâncias para o Problema AVS. Para facilitar a discussão, definimos como $||S||$ o comprimento total das seqüências do multiconjunto S, isto é $||S|| = \sum n_i$. Como abuso de linguagem, vamos nos referir ao multiconjunto S como conjunto S apenas.

A j-ésima instância do Problema AVS, para $j = 0, \ldots, L$, construída a partir da instância do Problema SC-MÍN, é dada pelo conjunto de seqüências $S_j = S \cup \{a^j, b^{L-j}\}$ e pelo inteiro $C = k||S|| + (k+1)L$, onde os caracteres a e b são caracteres novos que não pertencem ao alfabeto binário $\{0, 1\}$ sobre o qual as seqüências s_i são descritas.

A matriz de pontuação c usada para o Problema AVS é:

	0	1	a	b	␣
0	2	2	1	2	1
1	2	2	2	1	1
a	1	2	0	2	1
b	2	1	2	0	1
␣	1	1	1	1	0

(2.2)

É importante notar que a matriz c, embora simétrica, não satisfaz aos axiomas de métrica pois, por exemplo, $c(0, 0) = 2 > 0$.

Cada instância (S_j, C) do Problema AVS possui duas seqüências extras (a^j e b^{L-j}) em relação à instância de SC-MÍN e estas duas seqüências juntas possuem comprimento igual a L. A concatenação das duas seqüências pode ser interpretada como uma escrita em unário do inteiro L. Entretanto, por nossa hipótese de que $L \leq ||S||$, isso não apresenta um problema à polinomialidade do tamanho de cada instância do Problema AVS (ou mesmo do número de instâncias geradas pela redução, que é $L+1$), nem à polinomialidade do número de passos realizados pela redução.

Definido como a redução produz as instâncias do Problema AVS, passamos agora a verificar que a redução mapeia instâncias do Problema SC-MÍN cuja resposta é SIM a um conjunto de instâncias para o qual a resposta ao Problema AVS é SIM a pelo menos uma instância e vice-versa. Mais especificamente, vamos mostrar que uma instância $(S = \{s_i\}, L)$ nas condições acima admite uma superseqüência s de comprimento L se e somente se alguma instância $(S_j = S \cup \{a^j, b^{L-j}\}, C)$ admite um alinhamento com custo no máximo C. Disso seguirá o fato de que o Problema AVS com pontuação SP é NP-completo.

Antes da demonstração, entretanto, dependemos de alguns fatos auxiliares.

Lema 2.1 (Pontuação Constante)**.** *Seja A um alinhamento das $k+2$ seqüências de S_j, para algum j em $\{0, \ldots, L\}$. Então, a pontuação do alinhamento $A_{|S}$, o alinhamento induzido em S por A, é igual a $(k-1)||S||$.*

Demonstração. A prova é feita observando-se a pontuação de uma coluna qualquer do alinhamento $A_{|S}$. Fixemos uma coluna $\mathcal{C}$ de $A_{|S}$. Cada coluna de $A_{|S}$ possui k linhas. Em $\mathcal{C}$ não estão presentes caracteres a's ou b's, por hipótese, o que significa que os únicos caracteres possivelmente presentes são 0's, 1's ou $\sqcup$'s. Suponhamos que $\mathcal{C}$ tenha r_0 caracteres 0's, r_1 caracteres 1's e $r_\sqcup = k - (r_0 + r_1)$ caracteres $\sqcup$'s. Sua pontuação SP é igual a

$$\begin{aligned}
c(\mathcal{C}) &= \binom{r_0}{2}c(0,0) + \binom{r_1}{2}c(1,1) + \binom{r_\sqcup}{2}c(\sqcup,\sqcup) + r_0 r_1 c(0,1) + r_0 r_\sqcup c(0,\sqcup) + \\
&\quad r_1 r_\sqcup c(1,\sqcup) \\
&= r_0(r_0-1) + r_1(r_1-1) + 2r_0r_1 + r_0r_\sqcup + r_1r_\sqcup \\
&= r_0^2 - r_0 + r_1^2 - r_1 + 2r_0r_1 + r_0r_\sqcup + r_1r_\sqcup \\
&= (r_0^2 + 2r_0r_1 + r_1^2) - (r_0 + r_1) + r_\sqcup(r_0 + r_1) \\
&= (r_0 + r_1)^2 - (r_0 + r_1) + r_\sqcup(r_0 + r_1) \\
&= (r_0 + r_1 + r_\sqcup - 1)(r_0 + r_1) \\
&= (k-1)(r_0 + r_1).
\end{aligned}$$

Logo, a pontuação de uma coluna qualquer de $A_{|S}$ é proporcional (com fator de proporcionalidade $k-1$) ao número de caracteres de $s_1, \ldots, s_k$ que estão presentes naquela coluna.

Somando-se as pontuações de todas as colunas, concluímos que a pontuação SP de $A_{|S}$ é igual a $(k-1)||S||$, uma vez que existem $||S||$ caracteres de $s_1, \ldots, s_k$ em $A_{|S}$, de onde segue o resultado enunciado no lema. □

Lema 2.2 (Desmembramento de Colunas)**.** *Seja j um inteiro fixado entre 0 e L (inclusive) e seja A um alinhamento das seqüências de S_j. Então existe um alinhamento A' de S_j de mesma pontuação que A e tal que, em A', todas as colunas contenham só caracteres $\sqcup$'s, 0's e a's ou só caracteres $\sqcup$'s, 1's e b's.*

Demonstração. Para a demonstração do lema, mostraremos uma operação que chamaremos *desmembramento de coluna*, que pode ser aplicada às colunas de A que não forem de uma das formas desejadas (isto é, cujos caracteres não sejam apenas $\sqcup$'s, 0's e a's ou $\sqcup$'s, 1's e b's) para obtermos um novo alinhamento em que o número de colunas indesejadas seja 1 a menos do que em A e cuja pontuação seja igual à de A. Com a repetição desse processo até que todas colunas sejam de uma das formas desejadas, obtemos o alinhamento A' procurado.

A operação de desmembramento de coluna é feita da seguinte forma: seja $\mathcal{C}$ uma coluna genérica de A que contenha uma certa quantidade de caracteres $\sqcup$'s, 0's, 1's, a's e b's. A partir dessa coluna, definimos duas outras colunas, $\mathcal{C}'$ e $\mathcal{C}''$, de forma que $\mathcal{C}'$ contenha 0's nas linhas em que $\mathcal{C}$ tinha 0's, a's onde $\mathcal{C}$ tinha a's e $\sqcup$'s nas linhas em que $\mathcal{C}$ tinha $\sqcup$'s, 1's ou b's. A coluna $\mathcal{C}''$ é definida de maneira "complementar", tendo 1's nas linhas em que $\mathcal{C}$ tinha 1's, b's onde $\mathcal{C}$ tinha b's e $\sqcup$'s onde $\mathcal{C}$ tinha $\sqcup$'s, 0's ou a's. O alinhamento após a operação de desmembramento da coluna é o alinhamento A com a coluna $\mathcal{C}$ substituída pelo par de colunas $\mathcal{C}'$ e $\mathcal{C}''$.

Vejamos agora que o novo alinhamento possui a mesma pontuação que o alinhamento original A: como as únicas colunas que foram envolvidas no processo de desmembramento são $\mathcal{C}$, $\mathcal{C}'$ e $\mathcal{C}''$, as outras colunas permanecem com sua pontuação inalterada e podem ser ignoradas no cálculo da variação da pontuação.

Vamos supor que a coluna $\mathcal{C}$ tivesse r_0 caracteres 0, r_1 caracteres 1, $r_\sqcup$ caracteres $\sqcup$, r_a caracteres a e r_b caracteres b. Observe-se que tanto r_a quanto r_b são, no máximo, 1 e que, portanto, não é possível que uma coluna tenha um par a–a ou b–b alinhados, embora seja possível que ela tenha um par a–b.

O custo da coluna $\mathcal{C}$ é

$$\begin{aligned}
c(\mathcal{C}) &= \binom{r_0}{2}c(0,0) + \binom{r_1}{2}c(1,1) + \binom{r_\sqcup}{2}c(\sqcup,\sqcup) + \binom{r_a}{2}c(a,a) + \binom{r_b}{2}c(b,b) + \\
&\quad (r_0r_1c(0,1) + r_0r_\sqcup c(0,\sqcup) + r_0r_ac(0,a) + r_0r_bc(0,b)) + \\
&\quad (r_1r_\sqcup c(1,\sqcup) + r_1r_ac(1,a) + r_1r_bc(1,b)) + (r_\sqcup r_ac(a,\sqcup) + r_\sqcup r_bc(b,\sqcup)) + \\
&\quad r_ar_bc(a,b) \\
&= 2\binom{r_0}{2} + 2\binom{r_1}{2} + 2r_0r_1 + r_0r_\sqcup + r_0r_a + 2r_0r_b + \\
&\quad r_1r_\sqcup + 2r_1r_a + r_1r_b + r_\sqcup r_a + r_\sqcup r_b + 2r_ar_b \\
&= 2\binom{r_0}{2} + 2\binom{r_1}{2} + r_0(2r_1 + r_\sqcup + r_a + 2r_b) + r_1(r_\sqcup + 2r_a + r_b) + \\
&\quad r_\sqcup(r_a + r_b) + 2r_ar_b.
\end{aligned}$$

Na expansão acima, usamos os fatos de que $c(\sqcup,\sqcup) = 0$ e que $\binom{r_a}{2} = \binom{r_b}{2} = 0$. De

maneira similar, o custo da coluna $\mathcal{C}'$ é

$$\begin{aligned} c(\mathcal{C}') &= \binom{r_0}{2}c(0,0) + \binom{r_\sqcup + r_1 + r_b}{2}c(\sqcup,\sqcup) + \binom{r_a}{2}c(a,a) + \\ &\quad r_0(r_1 + r_\sqcup + r_b)c(0,\sqcup) + r_0 r_a c(0,a) + (r_1 + r_\sqcup + r_b)r_a c(a,\sqcup) \\ &= 2\binom{r_0}{2} + r_0(r_1 + r_\sqcup + r_b) + r_0 r_a + (r_1 + r_\sqcup + r_b)r_a \\ &= 2\binom{r_0}{2} + r_0(r_1 + r_\sqcup + r_a + r_b) + r_1 r_a + r_\sqcup r_a + r_a r_b, \end{aligned}$$

uma vez que a coluna $\mathcal{C}'$ possui $r_\sqcup + r_1 + r_b$ caracteres $\sqcup$'s. Analogamente ao caso de $\mathcal{C}'$, o custo da coluna $\mathcal{C}''$ é

$$c(\mathcal{C}'') = 2\binom{r_1}{2} + r_0(r_1 + r_b) + r_1(r_\sqcup + r_a + r_b) + r_\sqcup r_b + r_a r_b.$$

Logo, podemos ver facilmente que $c(\mathcal{C}) = c(\mathcal{C}') + c(\mathcal{C}'')$ e que, portanto, podemos substituir a coluna $\mathcal{C}$ pelo par de colunas $(\mathcal{C}', \mathcal{C}'')$ em A e obter um novo alinhamento de mesma pontuação.

Repetindo-se o desmembramento de colunas para as colunas de A que não estiverem no formato desejado, podemos concluir que existe um alinhamento A' cujas colunas são compostas apenas por $\sqcup$'s, 0's e a's ou apenas por $\sqcup$'s, 1's e b's e que possui a mesma pontuação que o alinhamento A original, conforme desejado. □

Lema 2.3. *Seja A um alinhamento de $S_j = S \cup \{a^j, b^{L-j}\}$ cujas colunas contenham apenas caracteres 0's, a's e $\sqcup$'s ou apenas caracteres 1's, b's e $\sqcup$'s. Então, a contribuição das seqüências a^j e b^{L-j} à pontuação* SP *de A é $||S|| + (k+1)L + q$, onde q é o número de caracteres 0's e 1's que não estão em colunas com a's ou b's.*

Demonstração. Para este lema, vamos proceder de maneira análoga ao Lema 2.1, calculando a pontuação referente a cada coluna e o total referente a todas as colunas.

Seja $\mathcal{C}$ uma coluna de A e suponhamos que esta coluna possua um caractere a. Então, pela hipótese, temos que a coluna é composta por r_0 caracteres 0's e $r_\sqcup$ caracteres $\sqcup$'s nas primeiras k linhas, por 1 caractere a na penúltima linha e por 1 caractere $\sqcup$ na última linha. A contribuição dos dois últimos caracteres à pontuação de $\mathcal{C}$ é, assim, $r_0 c(0,a) + r_0 c(0,\sqcup) + r_\sqcup c(\sqcup,a) + r_\sqcup c(\sqcup,\sqcup) + 1c(a,\sqcup) = (k+1) + r_0$.

Analogamente, se $\mathcal{C}$ é uma coluna de A que possui um caractere b, a contribuição das duas últimas linhas de $\mathcal{C}$ à pontuação SP de $\mathcal{C}$ é $(k+1) + r_1$, onde r_1 é a quantidade de 1's em $\mathcal{C}$. É importante observar que as contribuições que calculamos são válidas até mesmo para os casos em que $r_0 = 0$ ou que $r_1 = 0$, porque, nesses casos, as contribuições das últimas linhas à pontuação da coluna é de $(k+1)$.

Logo, somando as contribuições de todas as colunas que contenham caracteres a's ou b's, temos que a contribuição total das duas últimas linhas de tais colunas é igual a $(k+1)L + ||S|| - q$, onde $L = j + (L-j)$ é o número de colunas que contém caracteres a's ou b's e q é o número de caracteres 0's ou 1's que estão em colunas que *não* contém a's ou b's.

Se $\mathcal{C}$ é uma coluna de A de acordo com as hipóteses do lema, que possui 0's, mas que não possui a's, então $\mathcal{C}$ possui seus dois últimos caracteres sendo $\sqcup$'s. Nesse

caso, se $\mathcal{C}$ possui r_0 caracteres 0's e $r_\sqcup$ caracteres $\sqcup$'s em suas k primeiras linhas, a contribuição das duas últimas linhas à pontuação SP de $\mathcal{C}$ é $2r_0c(0,\sqcup)+2r_\sqcup c(\sqcup,\sqcup)+\binom{2}{2}c(\sqcup,\sqcup)=2r_0$. Analogamente, se $\mathcal{C}$ possui 1's, mas não possui b's, a contribuição das duas últimas linhas à pontuação de $\mathcal{C}$ é $2r_1$, onde r_1 é o número de 1's de $\mathcal{C}$.

Assim, a contribuição total das colunas que contém 0's mas que não contém a's ou que contém 1's mas que não contém b's é igual a $2q$, onde q é o número de caracteres 0's ou 1's nestas colunas.

Portanto, a contribuição total das duas últimas linhas à pontuação SP de A é igual a $||S||-q+(k+1)L+2q=||S||+(k+1)L+q$, como desejávamos mostrar. □

Corolário 2.3.1. *Nas condições do lema anterior, se a contribuição das linhas que contém a^j e b^{L-j} à pontuação* SP *de A é no máximo $||S||+(k+1)L$, então todo caractere* 0 *das seqüências de S está alinhado a um caractere a e todo caractere* 1 *das seqüências de S está alinhado a um caractere b.*

Demonstração. Basta ver que, nesse caso, $q=0$ caracteres das seqüências de S estão fora de colunas que contém um caractere a ou b. □

Estamos agora prontos para a demonstração do Teorema de Wang e Jiang.

Teorema 2.1 (Wang e Jiang, 1994)**.** *O Problema AVS é NP-completo.*

Demonstração. Naturalmente, o Problema AVS está em NP, pois dado um alinhamento (ótimo ou não) A, para uma instância de um conjunto $S=\{s_1,\ldots,s_k\}$ de seqüências e um inteiro C, pode-se decidir se $c(A)\leq C$ em tempo polinomial.

Vamos mostrar agora que a redução do Problema SC-MÍN ao Problema AVS (em suas versões de decisão) mapeia instâncias cuja resposta seja SIM a instâncias cuja resposta seja SIM e vice-versa.

Suponhamos que A seja um alinhamento de custo no máximo $C=k||S||+(k+1)L$ de $S_j=S\cup\{a^j,b^{L-j}\}$, para algum j. Pelo Lema 2.2, podemos supor que as colunas de A sejam organizadas de forma que seus caracteres sejam apenas 0's, a's e $\sqcup$'s ou apenas 1's, b's e $\sqcup$'s. Pelo Lema 2.1, sabemos que $A_{\downarrow S}$ possui pontuação exatamente $(k-1)||S||$ e que, portanto, a contribuição das seqüências a^j e b^{L-j} à pontuação de A é de, no máximo, $C-(k-1)||S||=||S||+(k+1)L$. Pelo Corolário 2.3.1, segue que cada 0 deve estar alinhado a um a e cada 1 deve estar alinhado a um b em A e, portanto, não há caractere das seqüências de S que não esteja alinhado a um a ou a um b.

Podemos construir uma superseqüência s para as seqüências de S associando um caractere 0 a cada coluna com um caractere a e um caractere 1 a cada coluna com um caractere b. A seqüência s assim construída é uma superseqüência das seqüências de $S=\{s_i\}$, já que cada caractere 0 ou 1 de s_i corresponde a um caractere 0 ou 1 de s. Naturalmente, a seqüência s possui comprimento $j+(L-j)=L$ e s é, portanto, uma solução para o Problema SC-MÍN com valor igual a L, o que significa que a redução mapeia instâncias do Problema AVS cuja resposta seja SIM a instâncias do Problema SC-MÍN com resposta SIM.

Vamos agora proceder à demonstração do outro caso. Seja s uma superseqüência das seqüências de S, com $|s|=L$, e seja j o número de 0's de s (de forma que, naturalmente, $L-j$ seja o número de 1's de s). Consideremos o conjunto $S_j=$

$S \cup \{a^j, b^{L-j}\}$. Para cada $s' \in S$, consideremos um alinhamento de s e s' em que os 0's de s' estejam alinhados apenas a 0's de s e em que os caracteres 1's de s' estejam alinhados a 1's de s. Naturalmente, como $|s'| \leq |s|$, podemos ter caracteres de s alinhados a espaços em s'.

Observe-se que cada um dos k alinhamentos considerados acima possui comprimento exatamente L e que, por construção, em nenhuma coluna desses alinhamentos há pares 0–1 alinhados.

O alinhamento A com pontuação desejada será obtido usando-se a seqüência s como uma espécie de "seqüência guia", da seguinte maneira: a i-ésima linha do alinhamento A conterá a seqüência s_i (com possíveis espaços inseridos) de forma que a a l'-ésima coluna do alinhamento A contenha os caracteres que estavam alinhados ao l'-ésimo caractere de s, para todo $1 \leq l' \leq L$. As duas últimas linhas do alinhamento A (isto é, as linhas $k+1$ e $k+2$) conterão um "espelho" de s em forma de a's e de b's, da seguinte maneira: nas colunas de s em que havia 0's, a $(k+1)$-ésima linha de A contém a's (portanto, j caracteres a's) e espaços nas colunas restantes desta linha; nas colunas de s em que havia 0's, a $(k+2)$-ésima linha de A contém espaços e, nas colunas em que havia 1's, a linha contém b's. Note-se que a seqüência s não faz parte de A.

Com essa construção, o alinhamento A possui exatamente L colunas e cada uma de suas colunas possui apenas caracteres 0's, a's e espaços ou 1's, b's e espaços. Além disso, não há caracteres 0's em colunas sem a's ou caracteres 1's em colunas sem b's. Logo, pelo Lema 2.3, a contribuição referente às seqüências de a's e b's à pontuação de A é $(k+1)L + ||S||$.

Esse alinhamento possui pontuação igual a $(k-1)||S||$ pela parte referente às primeiras k seqüências, conforme o Lema 2.1.

Portanto, o alinhamento A obtido pelo método descrito possui pontuação igual a $(k-1)||S|| + (k+1)L + ||S|| = k||S|| + (k+1)L = C$, como desejávamos mostrar. □

A idéia básica por trás da redução do Problema SC-Mín ao Problema AVS dada por Wang e Jiang é o fato de que se s é uma superseqüência das seqüências de $S = \{s_i\}$, então s pode ser alinhada a cada seqüência sem que caracteres diferentes fiquem alinhados em uma mesma coluna.

Outro fato a ressaltar é que como S está fixado, k e $||S||$ estão também fixados. Assim, a única parte variável no custo $C = k||S|| + (k+1)L$ de um alinhamento das seqüências de S é o valor L.

3. Resultados Relacionados

Conforme já citamos, a primeira demonstração de dificuldade do Problema AVS foi publicada em 1994 por Wang e Jiang [10]. A demonstração daquele artigo, apesar de concisa, restringe-se ao caso de pontuação SP em que a matriz de pontuação não é uma métrica (a matriz (2.2) atribui pontuação não-nula a pares de caracteres iguais).

Em 2001, Bonizzoni e Vedova mostraram que o Problema AVS, em sua versão de decisão, é NP-completo também para uma matriz de pontuação que satisfaz aos axiomas de métrica [1], que era uma questão em aberto até então. Em um artigo

posterior [8], Winfried Just mostrou que o problema de alinhar seqüências é NP-difícil para uma ampla classe de matrizes de pontuação (que inclui as matrizes do artigo de Bonizzoni e Vedova, conforme mostrado em [2]).

No mesmo artigo, Just mostrou também que existe uma matriz de pontuação para a qual o Problema AVS é MAXSNP-difícil. Essa matriz, no entanto, não é uma métrica (ela atribui pontuação 0 a caracteres diferentes do alfabeto).

Uma questão em aberto sobre o Problema AVS diz respeito ao fato de o problema ser ou não MAXSNP-difícil para matrizes que sejam métricas. Na realidade, não se sabe se o Problema AVS é MAXSNP-difícil para matrizes de pontuação que tenham elementos nulos na diagonal principal e diferentes de zero fora da diagonal, mesmo que a matriz não seja uma métrica [8, 9].

Abstract. In our work, we present a simple, complete and accessible proof of the fact that the Multiple Sequence Alignment of biological sequences is NP-hard. The proof is based on the work of Wang and Jiang [10], which is a frequently cited result, but with its proof commonly omitted. Given the elementary combinatorial tools used in our proof, we think that it is accessible even to undergraduate students in Computer Science.

Referências

[1] P. Bonizzoni, G.D. Vedova, The complexity of multiple sequence alignment with SP-score that is a metric, *Theoretical Computer Science*, **259** (2001), 63–79.

[2] R.T. Brito, "Alinhamento de Seqüências Biológicas", Dissertação de Mestrado, IME, USP, São Paulo, SP, 2003.

[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, "Introduction to Algorithms", The MIT Press, 1990.

[4] G. Fuellen, "Multiple Alignment", *Complexity International 4*, url: `http://www.csu.edu.au/ci/vol04/mulali/mulali.html`, 1997.

[5] M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman and Company, 1979.

[6] S.K. Gupta, J.D. Kececioglu, A.A. Schäffer, Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment, *Journal of Computational Biology*, **2**, No. 3 (1995), 459–472.

[7] D. Gusfield, "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology", Cambrige Press, 1997.

[8] W. Just, Computational complexity of multiple sequence alignment with SP-score, *Journal of Computational Biology*, **8**, No. 6 (2001), 615–623.

[9] W. Just, Comunicação Pessoal, Maio, 2002.

[10] L. Wang, T. Jiang, On the complexity of multiple sequence alignment, *Journal of Computational Biology*, **1** (1994), 337–348.