Relatório da Disciplina
MAC 5701 - Tópicos em Ciência da Computação

Francisco Elói Soares de Araújo
Orientador: Prof. Dr. José Augusto Ramos Soares

2 de Julho de 2001

1 Introdução

Esse relatório contém um estudo preliminar dos assuntos que iremos estudar durante o nosso Programa de Doutorado no Departamento de Computação do Instituto de Matemática e Estatística da Universidade de São Paulo.

Estudamos o Problema Alinhamento de Múltiplas Seqüências (*"Multiple Sequence Alignment"*), que é uma das questões mais estudadas em biologia molecular computacional e cujos resultados têm aplicações, principalmente, em duas grandes áreas da biologia que são: encontrar subregiões altamente conservadas em um conjunto de seqüências biológicas; e inferir a história evolutiva de algumas espécies através das respectivas seqüências a elas associadas. Além da biologia, o alinhamento de seqüências têm, também, aplicações em outras áreas, tais como recuperação de informações, reconhecimento de padrões e correção de erros.

Nas Seções 2, 3 e 4 definimos os problemas aqui estudados e na Seção 5 apresentamos uma bibliografia anotada dos principais artigos referentes a esses problemas.

2 Definição de Alinhamento e de Valor de um Alinhamento

Definição 1 Considere um alfabeto Σ e um conjunto $S = \{s_1, \ldots, s_k\}$ de seqüências, onde cada seqüência s_i possui $|s_i|$ caracteres em Σ. Seja $\Delta \notin \Sigma$ e $\Sigma_\Delta = \{\Delta \cup \Sigma\}$. Um *alinhamento* de S é uma matriz

$$A_S : \{1, \ldots, k\} \times \{1, \ldots, l(A_S)\} \to \Sigma_\Delta$$

que possui as seguintes propriedades.

1. para todo $i = 1, \ldots, k$, existem $j_1, \ldots, j_{|s_i|}$, $1 \leq j_1 < j_2 < \ldots < j_{|s_i|} \leq l(A_S)$, tais que

$$s_i = A_S[i, j_1] \ldots A_S[i, j_{|s_i|}]$$

e, para todo $j \notin \{j_1, \ldots, j_{|s_i|}\}$,

$$A_S[i,j] = \Delta;$$

2. Para todo j, existe i tal que

$$A_S[i,j] \neq \Delta.$$

Como conseqüência da definição acima, segue que $\max_i |s_i| \leq l(A_S) \leq \sum_{i=1}^{k} |s_i|$.

Chamamos de *espaço* o caracter Δ.

Denotamos o alinhamento A_S simplesmente por A se S estiver claro no contexto e dizemos que $l(A)$ é o *tamanho do alinhamento* A.

Para um conjunto S de seqüências, um alinhamento A de S e $T \subseteq S$, denotamos por A_T, o alinhamento de A_S induzido por T. Mais precisamente, A_T é um alinhamento das seqüências em T tal que, se o i-ésimo caracter de s e o h-ésimo caracter de t estão em uma mesma coluna em A_T, então esses também estão em uma mesma coluna em A_S.

Um alinhamento A_S serve como um instrumento para avaliar as igualdades e as diferenças entre as seqüências de S, e as colunas com caracteres distintos podem ser interpretadas como sendo ou substituições, ou inserções, ou remoções de caracteres suficientes para transformar uma seqüência em uma outra seqüência de S.

Para determinarmos a qualidade de um alinhamento do conjunto S, definimos uma função objetivo $V : \mathscr{A}_S \to \mathbb{R}$, onde $\mathscr{A}_S$ (ou simplesmente $\mathscr{A}$ se S estiver claro no contexto) é o conjunto de todos os alinhamentos de S. O valor $V(A)$ representa o grau de semelhança entre as seqüências de S, de acordo com o alinhamento A.

Considerando a natureza como sendo parcimoniosa quanto ao fato de transformar uma seqüencia em outra, estamos interessados em ou maximizar ou minimizar o valor de $V(A_S)$, dependendo se atribuímos um custo que seja tão grande ou tão pequeno, respectivamente, quanto maior o grau de semelhança entre as seqüências de S. Embora na prática as funções de maximização e de minimização sejam igualmente utilizadas, desde que essas duas medidas, em muitos casos, são relacionadas por uma simples fórmula, podendo uma medida ser obtida a partir de outra, conforme a obra *"Introduction to Computational Molecular Biology"*, de J. Meidanis e J. C. Setubal (pp 90) — PWS Publishing Co., 1997, trataremos neste texto somente considerando o problema de minimizar o valor de $V(A_S)$.

Dizemos que

$$D(S) = \min_{A \in \mathscr{A}} V(A)$$

é a *distância* entre os elementos de S.

Um alinhamento A, tal que $V(A) = D(S)$ é dito *alinhamento ótimo* dos elementos de S.

3 Funções Objetivo para o Alinhamento de duas Seqüências

As funções objetivo para o Problema Alinhamento de Múltiplas Seqüências que estudamos são derivadas de uma outra função objetivo v para o alinhamento de duas seqüências. Nesta seção, mostramos três funções objetivo para alinhar duas seqüências que podem ser utilizadas para alinhamento de múltiplas seqüências.

3.1 Distância de Edição

A maneira mais simples de atribuirmos um valor para um alinhamento A entre duas seqüências s_1 e s_2 é associarmos um custo para cada par de caracteres em cada coluna de A, através de uma *matriz de pontuação* M. Mais precisamente, conhecendo o custo $M[a, b]$, onde $a, b \in \Sigma_\Delta$, determinamos

$$v(A) = \sum_{j=1}^{l(A)} M[A[1, j], A[2, j]].$$

Na prática, nem toda matriz de pontuação tem significado biológico. As matrizes utilizadas, em geral, são chamadas de *matrizes métricas de pontuação*, e têm as seguintes propriedades de uma métrica

1. $M[i, j] > 0$, para todo $i \neq j$;

2. $M[i, i] = 0$ para todo i;

3. $M[i, j] = M[j, i]$, para todo i, j;

4. $M[i, j] \leq M[i, l] + M[l, j]$, para todo i, j, l.

3.2 Distância de Edição com função Peso para Lacunas

Além da matriz de pontuação, o critério de pontuação de uma alinhamento pode levar em conta o número de espaços consecutivos na seqüência alinhada. Isto deve-se ao fato de que a perda de um fragmento (vários caracteres consecutivos) pode ter ocorrido em um único evento de mutação, enquanto que a perda de caracteres não consecutivos deve ter ocorrido em tantos eventos de mutação quanto o número de caracteres perdidos.

Dizemos que qualquer seqüência maximal de espaços de uma seqüência em um determinado alinhamento é uma *lacuna* ("*gap*"). Dependendo da aplicação, podemos incorporar diferentes funções de peso W_g na definição de função v. A idéia é que W_g irá contribuir para penalizar um alinhamento com muitas lacunas.

3.3 Distância de Edição Normalizada

Uma outra variante para o critério utilizado para a definição da função objetivo é normalizar a pontuação obtida.

Imagine, por exemplo, que temos duas seqüências s_1 e s_2, $|s_1| = |s_2| = 1000$ e $v(s_1, s_2) = 1$, com alinhamento correspondente A; e que temos duas seqüências s'_1 e s'_2, $|s'_1| = |s'_2| = 1$ e $v(s'_1, s'_2) = 1$, com alinhamento correspondente A'. Embora $v(s_1, s_2) = v(s'_1, s'_2)$, intuitivamente percebemos que s_1 se "parece" muito mais com s_2, do que s'_1 e s'_2.

Uma maneira de capturar essa diferença é normalizar o valor em relação ao tamanho do alinhamento.

Podemos, então, definir a *Distância de Edição Normalizada* entre duas seqüências, em relação a algum critério de pontuação como sendo o valor mínimo de um alinhamento A, dividido por $l(A)$.

4 Versões do Problemas do Alinhamento de Múltiplas Seqüências

De acordo com a aplicação podemos enunciar diferentes versões para o Problemas de Alinhamento de Múltiplas Seqüências.

4.1 Critério de Pontuação-SP

Um critério muito utilizado leva em consideração a soma dos valores de todos os alinhamentos entre duas seqüências. Mais precisamente, enunciamos

Problema 1 *Dado um conjunto de seqüências $S = \{s_1, \ldots, s_k\}$, encontrar uma alinhamento A que minimiza o valor de*

$$V(A) = \sum_{i<h} v(A_{\{s_i, s_h\}}).$$

4.2 Alinhamento por Árvores

Nesta versão, estamos interessados em inferir as seqüências que representam ancestrais comuns hipotéticos de um conjunto de espécies atuais representadas por outro conjunto de seqüências.

Formalmente, para o conjunto $S = \{s_1, \ldots, s_k\}$ e T de seqüências, uma *árvore evolutiva* $G = (S \cup T, E)$ é uma árvore que têm vértices $S \cup T$, onde S representam as folhas e T os vértices internos e E é o conjunto de arestas de G.

O Problema *Alinhamento por Árvores* (*"Optimal Evolutionary Tree", "Multiple Tree Alignment", ou simplesmente "Tree Alignment"*), é o seguinte.

Problema 2 *Dado um conjunto de seqüências S, encontrar um conjunto de seqüências T, uma árvore evolutiva $G = (S \cup T, E)$, e um alinhamento A para $S \cup T$, tal que*

$$V(A_{S \cup T}) = \sum_{(s_1, s_2) \in E} v(A_{\{s_1, s_2\}})$$

seja mínimo.

4.3 Alinhamento por Árvores para uma dada Filogenia

Uma variante do Problema 2 é o Problema *Alinhamento por Árvores com uma dada Filogenia* ("*Tree Alignment with a given Phylogeny*") que, dado um conjunto de seqüências em S, supõem conhecer a árvore evolutiva $G = (T \cup S, E)$, sem, entretanto, conhecer as seqüências que representam os nós de T.

Problema 3 *Dados um conjunto S de k seqüências, uma árvore evolutiva G com k folhas, cada qual associada com uma única seqüência em S, determinar um conjunto T de seqüências tal que cada uma seqüência represente um único vértice interno de G e um alinhamento A tal que*

$$V(A_{S \cup T}) = \sum_{(s_1, s_2) \in E} v(A_{\{s_1, s_2\}})$$

seja mínimo.

Uma variante do Problema 3, dada a seguir, privilegia o menor tamanho da maior distância entre duas seqüências.

Problema 4 *Dado um conjunto S de k seqüências, uma árvore evolutiva G com k folhas, cada qual associada com uma única seqüência em S, determinar um conjunto T de seqüências tal que cada uma seqüência represente um único vértice interno de G e um alinhamento A tal que*

$$\max_{(s_1, s_2) \in E} v(A_{\{s_1, s_2\}})$$

seja mínimo.

Um caso especial do Problema 3 é ter como entrada uma árvore que é uma estrela. Este é um modelo razoável para a história evolutiva para alguns tipos de entrada. Por exemplo, ele é apropriado quando comparamos proteínas homólogas de um primata, um carnívoro, um insetívoro e um ungulado, porque todos eles divergem aproximadamente ao mesmo tempo a partir de um ancestral comum. Esse problema é o Problema *Alinhamento por Estrela* ("*star-aligment*") que pode ser assim enunciado.

Problema 5 *Dado um conjunto $S = \{s_1, \ldots, s_k\}$ de k seqüências, determinar uma seqüência t e um alinhamento A para o conjunto $S \cup \{t\}$, tal que*

$$V(A) = \sum_{i=1}^{k} v(A_{\{s_i,t\}}),$$

seja mínimo.

4.4 Alinhamento de Múltiplas Seqüências com número limitado de Lacunas

Dado um alinhamento A de um conjunto $S = \{s_1, \ldots, s_k\}$, dizemos que A é uma 0-lacuna, se, para cada $i = 1, \ldots, k$ existe $j = 0, \ldots, n-1$, tal que

$$A[i, j+1]A[i, j+2] \ldots A[i, j+|s_i|] = s_i.$$

O *Problema Alinhamento de Múltiplas Seqüências* 0*-lacuna* (*"Gap-0 Multiple Sequence Alignment"*) pode ser assim enunciado.

Problema 6 *Dado um conjunto S de seqüências, encontrar uma alinhamento A que é uma* 0*-lacuna e que minimiza o valor de $V(A)$.*

Se $j = 0, 1$, para todo i, dizemos que A é uma 0-1-lacuna.

O *Problema Alinhamento de Múltiplas Seqüências* 0-1*-lacuna* (*"Gap-*0-1 *Multiple Sequence Alignment"*) pode ser assim enunciado.

Problema 7 *Dado um conjunto S de seqüências, onde toda seqüência de S tem o mesmo tamanho, encontrar uma alinhamento A que é uma* 0-1*-lacuna e que minimiza $V(A)$.*

4.5 Alinhamento de Múltiplas Seqüências com Lacunas "bem distribuídos"

Definição 2 Dado um conjunto S de seqüências, onde cada seqüência de S tem o mesmo tamanho, um *alinhamento c-diagonal* A é um alinhamento tal que, para quaisquer p, i e j, se o p-ésimo caracter de s_i e o p-ésimo caracter de s_j estão nas p'-ésima e p''-ésima colunas de A, respectivamente, então

$$|p' - p''| \leq c.$$

Podemos enunciar a seguinte versão do Problema Alinhamento Múltiplo de Seqüências com o critério de pontuação-SP.

Problema 8 *Para um inteiro c, dado um conjunto S, determinar um alinhamento A que minimiza a pontuação-SP e é um alinhamento c-diagonal.*

Podemos combinar outros critérios de pontuação para definir outros problemas que exigem que a saída seja um alinhamento c-diagonal.

5 Bibliografia anotada

Nesta seção descrevemos os resultados obtidos nos principais artigos estudados.

1. S. F. Altschul and D. Lipman. *Trees, stars, and multiple sequence aligment.* SIAM Journal on Applied Math, 49, No 1, pp 197—209, 1989.

   Neste artigo, os autores descrevem um algoritmo para o Problema 3 usando uma extensão do algoritmo de Carrilo e Lipman para o Problema 1. O algoritmo determina um limitante superior para o valor da função objetivo e encontra um alinhamento ótimo percorrendo um reticulado reduzido determinado pelo limitante superior.

2. A. N. Arlsan and Ö. Eğecioğlu. *An Efficient Uniform-Cost Normalizer Edit Distance Algorithm.* Proc. 6-th String Processing and Information Retrieval Conference (SPIRE'99), IEEE Computer Society, Sep. 1999, Cancun, Mexico, pp. 8-15.
   http://www.cs.ucsb.edu/TRs/Docs/TRCS99-14.ps

   Neste artigo, os autores descrevem um algoritmo que gasta tempo $O(mn \log n)$ para o Problema de Distância de Edição Normalizada entre duas seqüências, de tamanho m e n, quando o custo entre as operações de edição são uniformes.

3. V. Bafna and E. Lawler and P. Pevzner. *Approximation algorithms for multiple sequence alignment.* Theoretical Computer Science, 182, pp 233—244, 1997.

   Para qualquer l fixo, os autores descrevem um algoritmo $(2 - \frac{l}{k})$-aproximação para o problema de alinhamento de k seqüências de tamanho n usando o critério de pontuação-SP. O tempo gasto por esse algoritmo é $O(k^3 + g(2l+5, n))$, onde $g(l, n)$ é o tempo gasto para determinar o alinhamento ótimo com peso para cada subconjunto de l seqüências.

   Além disso, eles descrevem um algoritmo probabilístico que, para qualquer $\epsilon > 0$, devolve um alinhamento tal que, com probabilidade $1 - \epsilon$, tem um fator de aproximação de $(2 - \frac{l}{k})$, gastando tempo $O(k^2 \lceil \log(\frac{k}{\epsilon}) \rceil g(2l, n))$.

4. P. Bonizzoni and G. Della Vedova. *The complexity of multiple alignment with SP-score that is a metric.* Theoretical Computer Science, 2001.

   Mostram que o Problema 1 é NP-difícil, mesmo quando a matriz de pontuação é uma métrica e o alfabeto tem apenas dois símbolos. Para isto, eles reduzem o Problema "Vertex Cover" para o Problema 1.

5. H. Carrilo and D Lipman. *The multiple sequence alignment problem in biology.* SIAM Journal on Applied Math, 48, No. 5, pp 1073—1082, 1988.

   Para o problema de alinhamento de seqüências múltiplas com o critério de pontuação-SP, os autores descrevem um algoritmo baseado em programação dinâmica que requer tempo na ordem do produto das seqüências de entrada. Esse algoritmo, através de um método heurístico, determina um alinhamento cuja pontuação não é "muito maior" que a pontuação do alinhamento ótimo. Essa pontuação é usada então, como limitante superior do problema que diminui o número de pontos que devem ser computados na matriz de programação dinâmica k-dimensional, onde k é o número de seqüências a serem alinhadas, para determinar o alinhamento ótimo.

6. S. Chan and A. Wong and D. Chiu. *A survey of multiple sequence comparison methods.* Bulletin of Mathematical Biology, 54, pp 563—598, 1992.

   Embora escrito em 1992, o artigo é excelente para iniciar os estudos dos problemas de alinhamentos multiplos de seqüências utilizando os critérios de pontuação-SP e para alinhamentos por árvores para uma dada filogenia. Os autores descrevem os principais algoritmos e as principais heurísticas utilizadas até aquele momento para esses problemas.

7. L. R. Founds and R. L. Graham. *The Steiner problem in phylogeny is NP-complete.* Advances in Applied Mathematics, 3, pp 43—49, 1982.

   Os autores provaram que uma variante do Problema 2, onde a função v é definida como distância de Hamming, é NP-completa.

8. D. Gusfield. *Efficient methods for multiple sequence alignment with guaranteed error bounds.* Bulletin of Mathematical Biology, 55, No. 1, pp 141—154, 1993.

   Gusfield propõe pela primeira vez algoritmos de aproximação para os Problemas 1 e 3 que possuem um fator de aproximação $2 - \frac{2}{k}$ para k seqüências de entrada.

   Além disso, para o Problema 1, Gusfield descreve outro algoritmo, muito mais rápido que o algoritmo de aproximação descrito, que devolve, com alta probabilidade, um alinhamento de custo bem próximo do ótimo.

9. T. Jiang and E. L. Lawler and L. Wang. *Aligning sequences via an evolutionary tree: complexity and approximation.* STOC'94, 1, 760–769, 1994.

   Neste artigo, os autores mostram que o Problema 3 é NP-difícil mesmo que a árvore dada seja uma árvore binária e que o Problema 2 é MAXSNP-difícil.

   Eles também mostram um algoritmo de aproximação com fator 2 para o Problema 3 que gasta tempo $O(k^3 + k^2n^2)$, onde k é o número de seqüências a serem alinhadas e n é o tamanho máximo de uma seqüência. Estendendo esse algoritmo, os autores obtém o primeiro PTAS descrito que, para qualquer $t > 1$, tem um fator de aproximação $1 + \frac{3}{t}$ gastando tempo $O(k^{d^{t-1}+2}M(d, t-1, n))$, onde $M(d, t-1, n)$ é o tempo necessário para alinhar de forma ótima uma árvore de profundidade $t - 1$ e é limitada superiormente pelo tempo de $O(n^{(d^{t-1}-1)/(d-1)})$.

10. W. Just. *Computational complexity of multiple sequence aligment with SP-score.* http://www.math.ohiou.edu/~just/PAPERS/dev21www.ps

    Neste artigo, Just mostra os seguintes resultados.

    (a) O Problema 1 é NP-difícil mesmo para uma pontuação-SP, representada por qualquer matriz $\mathcal{M}$ que possui a seguinte propriedade: uma submatriz $\mathbb{M}$ pode ser obtida a partir da matriz de pontuação $\mathcal{M}$, considerando somente dois elementos quaisquer a

e b de Σ, tais que

$$\mathbb{M} = \begin{array}{cccc} & a & b & \Delta \\ a & v_a & u & y \\ b & u & v_b & z \\ \Delta & y & z & x \end{array}$$

onde x, y e z são números não-negativos e $u > \max\{0, v_a, v_b\}$.

(b) O Problema 1 é MAXSNP-difícil, para uma matriz de pontuação que não é uma métrica.

(c) Os Problemas 6 e 7 são NP-difíceis.

11. B. Ma and L. Wang and M. Li. *Near optimal multiple alignment within a band in Polynomial Time* Proc. 32nd ACM Symp. Theory Computing, 2000.

    Os autores mostram os seguintes resultados.

    (a) Descrevem um algoritmo que é um esquema polinomial de aproximação (PTAS) para o Problema 8 e um PTAS para o Problema 5 com a saída restrita a um alinhamento c-diagonal.

    (b) Mostram que o Problema 6 é NP-difícil, mesmo que a matriz de pontuação seja

    $$M(a,b) = \begin{cases} 0 & \text{se } a = b \\ 1 & \text{caso contrário.} \end{cases}$$

    Para isto, reduzem o Problema "Maximum Cut-3" para a versão de decisão do Problema 6. Esse resultado tem como corolário que o Problema 8 também é NP-difícil.

12. P. Pevzner. *Multiple alignment, communication cost, and graph matching.* SIAM Journal on Applied Math, 52, No. 6, pp 1763—1779, 1992.

    O autor descreve um algoritmo de aproximação para o Problema 1 que possui um fator de aproximação $2 - \frac{3}{k} + \frac{2}{k(k-1)}$ e gastando tempo $O(n^3k^3 + k^4)$.

13. R. Ravi and J. D. Kececioglu. *Approximation algorithms for multiple sequence alignment under a fixed evolutionaty tree.* Discrete Applied Mathematics, 88, pp 355—366, 1998.

    Neste artigo, os autores descrevem um algoritmo de aproximação para o Problema 3 e analisam a sua complexidade de tempo e o seu fator de aproximação para árvores d-árias regulares, onde d é uma constante. Em tais árvores, o algoritmo tem um fator de aproximação $\frac{d+1}{d-1}$, consumindo tempo $O(k^dT(d,n) + k^{2d}n^2)$, onde $T(d,n) = O(d2^dn^d + d|\Sigma|^{d+1})$ e n é o número de caracteres de seqüência de maior comprimento em S.

    Para o Problema 4, com custo unitário nas operações de edição (ou seja $M(a,b) = 0$ se $a = b$, e $M(a,b) = 1$, caso contrário), descrevem um algoritmo exato que gasta tempo $O(|\Sigma|kn^{k+1})$ e um algoritmo de aproximação que gasta tempo $O(n)$ e tem um fator de aproximação de $2\rho + 1$, onde ρ é o número máximo de aresta em um caminho entre dois vértices internos da árvore dada.

14. D. Sankoff. *Minimal Mutation trees of sequences.* SIAM J. Appl. Math, 28, pp 35—42, 1975.

Neste artigo, Sankoff descreve um algoritmo exato para o seguinte problema:
Dada uma árvore $T = (U, E)$ com $U = \{v_1, \ldots, v_k, v_{k+1}, \ldots, v_n\}$, e um conjunto de seqüências finitas $S = \{s_1, \ldots, s_k\}$, $k < n$, s_i rotulando o vértice v_i, determinar os rótulos $s_{k+1}, \ldots, s_n$ para os vértices $v_{k+1}, \ldots, v_n$, respectivamente, tal que, o valor de um alinhamento A,

$$V(A_{\{s_1,\ldots,s_n\}}) = \sum_{(v_i,v_j)\in E} v(A_{\{s_i,s_j\}}),$$

onde

$$v(A\{s_i, s_j\}) = \sum_{h=1}^{l(A)} (1 \text{ se } A[i,h] \neq A[j,h] \text{ e } 0 \text{ caso contrário}),$$

seja mínimo.

15. E. Ukkonen. *Algorithms for approximate string matching.* Information and Control, 64, pp 100—118, 1985.

Nesse artigo, Ukkonen generaliza os resultados obtidos em "On approximations string mathching" (Ukkonen 83) descrevendo, para o Problema 1 restrito a duas seqüências de tamanho m e n, e a uma matriz de pontuação D, que é uma métrica, um algoritmo que gasta tempo e espaço $O(d * \min(m, n))$, onde d é o valor da pontuação-SP do alinhamento ótimo.

Para o caso onde $D[i, j]$ é uma constante para $i \neq j$ e $D[i, i] = 0$, Ukkonen descreve um algoritmo que gasta tempo $O(d * \min(m, n))$ e espaço $O(d * \min(d, m, n))$, e espaço $O(\min(m, n, d))$ se queremos determinar somente a pontuação-SP do alinhamento ótimo.

16. E. Ukkonen. *On approximation string matching.* in Proc. Internat. Conference on Foundation of Computing Theory, Lecture Notes in Computer Science 158, pp 487—496, 1983.

Nesse artigo, Ukkonen descreve um algoritmos que, para uma matriz de pontuação D, onde $D[i, j] = 1$ se $i \neq j$ e $D[i, j] = 0$ caso contrário, resolve o Problema 1 restrito a duas seqüências em tempo $O(d * \min(|s_1|, |s_2|))$ e espaço $O(d^2)$, onde d é a pontuação-SP do alinhamento encontrado. Se estamos interessado apenas em determinar se a pontuação-SP de um alinhamento ótimo é menor ou igual a t, o algoritmo pode ser modificado para responder ao teste em tempo $O(t * \min(|s_1|, |s_2|))$ e espaço $O(t)$.

17. L. Wang and D. Gusfield. *Improved Approximation Algorithms for Tree Alignment.* Journal of Algorithms, 25, pp 255—273, 1997.

Assumindo que um critério de pontuação satisfaz a desigualdade triangular, os autores propõe, para o problema de alinhamento multiplo de seqüências quando é dada uma filogenia onde a árvore dada T é *deg*-regular e de profundidade d, um algoritmo 2-aproximação que é executado em tempo $O(kd + kdn^2)$, e um PTAS que, para um t fixo, possui razão de aproximação de $1 + \frac{2}{t} - \frac{2}{t2^t}$, gastando tempo $O(\min\{2^t, k\}kdM(deg, t-1, n))$.

18. L. Wang and T. Jiang. *On the complexity of multiple sequence alignment.* Journal of Computational Biology, 1(4), pp 337—348, 1994.

Nesse artigo, os autores mostraram os seguintes resultados.

(a) Provam que o Problema 1 é NP-difícil, mostrando ser NP-completa a versão de decisão do problema. Para isto, eles reduzem o Problema "Shortest Common Supersequence" para o Problema 1.

(b) Mostram que o Problema 2 é MAXSNP-difícil, mesmo que a matriz de pontuação seja uma métrica, o que implica que não existe um algoritmo com esquema polinomial de aproximação (PTAS) para o problema, a menos que P = NP. Para isto, eles descrevem uma L-redução do Problema "Vertex Cover-B" para o Problema 2.

(c) Para o Problema 3, os autores mostram que que esse é NP-difícil, mesmo que tenha como entrada uma árvore binária. Para isto, eles reduzem o Problema "Shortest Common Supersequence" para a versão do Problema 3 que possui como entrada uma árvore binária. Entretanto, os autores afirmam mostrar no artigo "Approximation algorithms for tree alignment with a given phylogeny" (Algorithmica, 1993) que essa versão não é MAXSNP-difícil, descrevendo nesse um algoritmo PTAS.

(d) Mostram que o Problema 5 é MAXSNP-difícil. Para isto, fazem uma L-redução do Problema "Max Cut-B" para o Problema 5.