USO DE PROGRAMAÇÃO DINÂMICA EM DOBRAMENTO DE RNA

DYNAMIC PROGRAMMING USE IN RNA FOLDING

Luiz Carlos da Silva Rozante

IMES – Universidade Municipal de Ensino Superior de São Caetano do Sul
Instituto de Matemática e Estatística – USP - Universidade de São Paulo

RESUMO

Os métodos laboratoriais para determinação da estrutura do RNA são onerosos. A estrutura secundária do RNA, além de fornecer informações acerca da função da molécula, serve também como importante etapa na definição de sua estrutura terciária. Daí a importância em se desenvolver métodos computacionais, rápidos e precisos de predição da estrutura secundária, a partir da estrutura primária. As duas mais importantes estratégias de resolução do problema estão baseadas em critérios de estabilidade termodinâmica (de energia livre mínima) e na identificação dos dobramentos comuns entre moléculas homólogas. No primeiro caso, os algoritmos mais importantes são baseados em técnicas de programação dinâmica. Situado no contexto da genômica estrutural e da bioinformática, este trabalho apresenta os modelos propostos para o problema, além de descrever formalmente as várias técnicas e métodos envolvidos na sua resolução. Desenvolvemos também implementações eficientes dos algoritmos mais expressivos baseados em cálculo de energia livre mínima.

Palavras-chave: dobramento de RNA, estrutura secundária, programação dinâmica, minimização de energia livre.

ABSTRACT

The experimental methods for deducing RNA structure are costly. The RNA secondary structure besides supplying information about the molecule function, it also serves as an important step in determining its tertiary structure. So it is important to develop fast and accurate computer methods on prediction of secondary structure from primary one. Two most significant strategies for solving the problem are based on thermodynamic stability criterion (of minimum free-energy) and in the search of the common folding among homologous molecules. In the first case, the most important algorithms are based on techniques of dynamic programming. Being situated on structural genomic and bio computer science areas, this work presents the models proposed for the problem and it describes formally the several techniques and methods involved in the solution of this problem. We also developed efficient implementations of the most expressive algorithms based on calculation of free energy minimum.

Keywords: RNA folding, RNA secondary structure, dynamic programming, free energy minimization.

1. INTRODUÇÃO

Uma molécula de RNA consiste em uma cadeia de nucleotídeos conectados por ligações covalentes. Cada nucleotídeo contém um grupo fosfato, um açúcar (ribose) e uma base. Essa molécula de RNA é um polímero e é formado pela ligação de grupos fosfato. Somente as bases diferem e elas são quatro: Adenina *(A)*, Citosina *(C)*, Guanina *(G)* e Uracil *(U)*.

Sob condições naturais, uma cadeia de RNA dobra-se sobre si mesma, através da formação de pontes de hidrogênio entre bases complementares *(A* com *U* e *C* com *G)* e entre bases *wobble (U* com *G)*. As bases complementares formam pares de bases estáveis, através da criação de pontes de hidrogênio entre elas, que são ditos pares de bases de Watson-Crick. Além disso, é possível considerar também (e geralmente o é) o par G-U, cuja ligação é mais fraca e que é denominado par de base *oscilante* ou *instável* (em inglês *wobble*). Os pares de bases de Watson-Crick, juntamente com os oscilantes, são denominados pares de bases *canônicos*.

A *estrutura secundária* de uma molécula de RNA é o conjunto de pares de bases canônicos — ou simplificadamente pares de bases —, que ocorrem na "dobradura" natural da molécula.

1.1. Representação e conceituação matemática

De um modo mais formal, uma molécula de RNA é representada como uma seqüência de n caracteres $R = r_1, r_2, \ldots, r_n$, onde $r_i \in \{A, U, C, G\}$ representa o i-ésimo nucleotídeo. Uma *estrutura secundária* da molécula - cuja noção topológica está ilustrada na Figura 1 - é um conjunto S de pares de inteiros, tal que cada par $(i, j) \in S$, com $1 \leq i < j \leq n$, satisfaz as seguintes restrições:

Restrição 1: r_i e r_j é um par de base canônico;

Restrição 2: $j - i > t$, onde tipicamente $t = 4$ ou $t = 3$;

Restrição 3: se $i < i'$ e $(i', j'), \in S$, então somente um dos casos ocorre:

Caso 2: $i < j < i' < j'$;

Caso 3: $i < i' < j' < j$.

Se $(i, j) \in S$ dizemos que r_i e r_j são bases *pareadas*. A Restrição 2 modela um fato da realidade biológica, observado experimentalmente, que consiste na impossibilidade de uma molécula dobrar-se sobre si mesma - em alguma parte - de forma pontiaguda. A Figura 2 ilustra um exemplo onde temos $t = 4$.

Os Casos 2 e 3 excluem uma configuração natural chamada *pseudo-nó*. Dizemos que ocorre um *pseudo-nó* quando existem pares $(r_i, r_j), (r_{i'}, r_{j'}) \in S$ com $i < i' < j < j'$. Sua exclusão simplifica o problema. Alguns tipos de pseudo-nós foram tratados no algoritmo de Rives e Eddy, cujas complexidades de tempo e espaço são, respectivamente, $O(n^6)$ e $O(n^4)$ para uma molécula com n bases. Entretanto, Lynas e Pedersen[2] mostraram recentemente que o problema geral é NP-difícil.

Poderíamos ser levados a conceber um algoritmo trivial que enumerasse todas as possíveis candidatas a estruturas e depois simplesmente escolhesse, entre aquelas que podem ser estruturas secundárias, aquela que correspondesse à estrutura mais estável (baseada num critério termodinâmico, por exemplo). No entanto, o número possível de candidatas a estruturas é de pelo menos 2^n, para seqüências de n nucleotídeos. Isto, evidentemente, torna tal algoritmo inviável para seqüências de tamanho razoável.

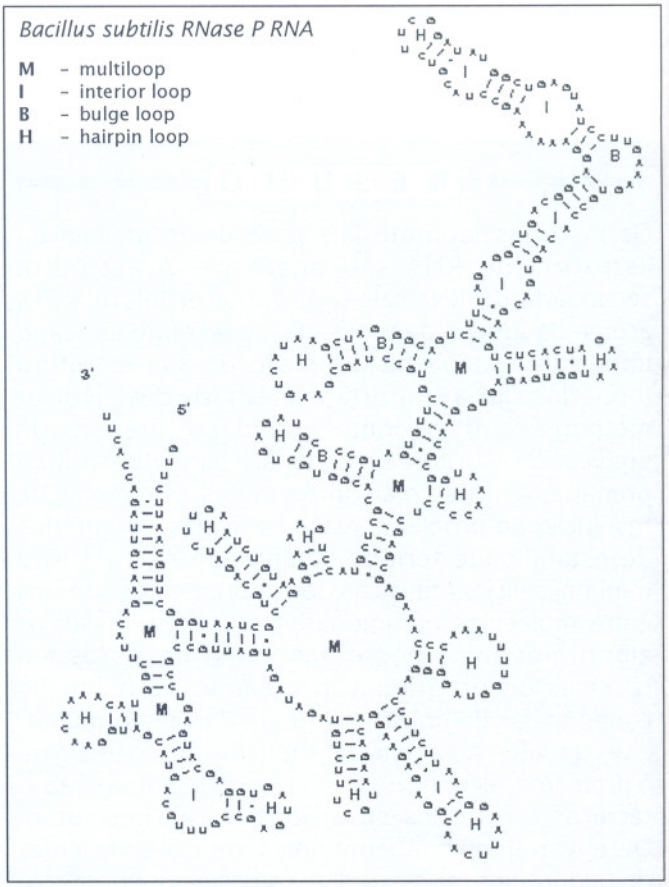

Figura 1: Exemplo de estrutura secundária do RNA na sua representação normal. Obtida em http://www.ibc.wustl.edu/_zuker/Bio-5495/RNAfold-html.

[1] RIVAS E.; EDDY S. A dynamic programming algorithm for RNA structure prediction including pseudoknots. **Journal of Molecular Biology**, (285):2053-2068, 1999.

[2] LYNGS R. B.; PEDERSEN C. N. S. Pseudoknots in RNA secondary structure. In **Proc. 4rd Int. Conf. Computational Molecular Biology** *(RECOMB'00)*. ACM, Apr. 2000.

2. USANDO CÁLCULO DE ENERGIA LIVRE MÍNIMA

A estratégia mais difundida para se predizer a estrutura secundária de uma molécula de RNA é baseada no cálculo da estrutura secundária de energia livre mínima. Tal estratégia utiliza-se da idéia de se atribuir uma energia a cada um de seus pares de bases, ou a seus elementos formadores estruturais, como laços internos, barrigas, arcos, hélices e multilaços[3].

2.1. Algoritmo básico

Um modelo simplificado do problema, cuja idéia inicial foi proposta por Nussinov[4], supõe que as energias de cada um dos pares de bases são independentes entre si, de maneira que a energia total da estrutura S pode ser escrita como

$$Energia(S) = \sum_{(i,j) \in S} \alpha(r_i, r_j), \text{ onde } \acute{a}(r_i, r_j) < 0.$$

Ou seja, pressupõe-se existir uma função $\acute{a}$ tal que $\acute{a}(r_i, r_j)$ é definida como a energia de ligação do par de bases (r_i, r_j).

Esta idéia permite-nos modelar o problema como segue. Seja a seqüência $R=r_1, r_2, \ldots, r_n$ para a qual desejamos encontrar uma estrutura secundária S de energia livre mínima.

Definimos

$$E(R) = \min_{S} \{Energia(S)\},$$

onde S varia em todas as estruturas secundárias de R.

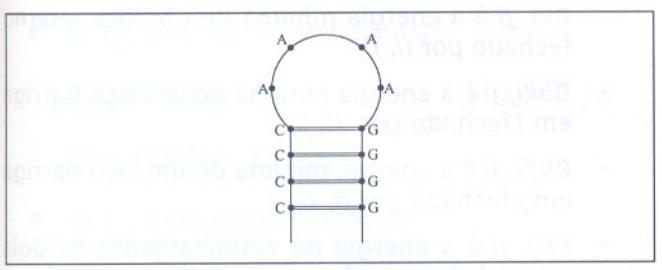

Figura 2: Um exemplo de "arco" formado com quatro (t=4) bases do tipo 'A'.

Tomando a subseqüência $R_{i,j}=r_i, r_{i+1}, \ldots, r_j$, $1 \le i < j \le n$ para a qual desejamos encontrar a estrutura secundária $S_{i,j}$ correspondente de energia livre mínima, há quatro possibilidades a serem tratadas:

1. Se r_i não é base pareada em nenhuma estrutura de energia mínima, então $E(R_{i,j}) = E(R_{i+1,j})$;
2. Se r_j não é base pareada em nenhuma estrutura de energia mínima, então $E(R_{i,j}) = E(R_{i,j-1})$;
3. Se em alguma estrutura de energia mínima, r_i e r_j são bases pareadas, mas não entre si, então $E(R_{i,j}) = \min_k \{E(R_{i,k}) + E(R_{k+1,j})\}$, para $i + t < k < j - t$;
4. Se r_j é pareada com r_i em alguma estrutura de energia mínima, então $E(R_{i,j}) = E(R_{i+1,j-1}) + \alpha(r_i, r_j)$.

De modo mais formal, reescrevemos então as situações (modelo) acima como

$$E(R_{i,j}) = \begin{cases} 0, & se\ j - i \le t \\ \min \begin{cases} E(R_{i+1,j}), \\ E(R_{i,j-1}), \\ \min_{i+t<k<j-t} \{E(R_{i,k}) + E(R_{k+1,j})\}, \\ \alpha(r_i, r_j) + E(R_{i+1,j-1}) \end{cases}, & \text{caso contrário.} \end{cases} \quad (2)$$

A Expressão 2 é resolvida por programação dinâmica. Resolvemos esta recorrência através de um algoritmo interativo, o qual preenche uma matriz de energias E, onde cada célula $E[i][j]$ armazena $E(R_{i,j})$, $1 \le i < j \le n$. Atribuímos $E[i][j] \leftarrow 0$ para valores iniciais $j - i \le t$. Lembremos que o parâmetro t relaciona-se à impossibilidade da molécula dobrar-se, sobre si mesma, de forma demasiado pontiaguda (Figura 2 da Seção 1.1). Este algoritmo é de complexidade $O(n^3)$.

Uma vez calculado $E[1][n]$, a computação (identificação) do dobramento de energia mínima $S_{1,n}$ é feita através de um algoritmo rastreador (do inglês *traceback*).

2.2. Incorporação de laços

Infelizmente, a abordagem citada é insuficiente para capturar e representar algumas situações que concretamente ocorrem na definição das estruturas secundárias, pois não leva em consideração a influência que a energia de um par de bases exerce sobre outro, notadamente os pares adjacentes; tampouco contabiliza as energias associadas a estruturas denominadas laços, definidos a seguir e ilustrados na Figura 3.[5, 6, 7, 8]

[3] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull. Math. Biol.*, (46):591-621, 1984.
[4] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algoritms for loop matchings. *SIAM J. appl. Math.*, (35):68-82, 1978.
[5] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull. Math. Biol.*, (46):591-621, 1984.
[6] D. Sankoff. Simultaneous solution of the mRNA folding, alignment and protosequense problems. *SIAM J. Appl. Math.*, (5):1-35, 1985.
[7] M. S. Waterman and T. F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics*, (7):455-464, 1986.
[8] M. Zuker and C. D. H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. (333):333-344, 1999.a

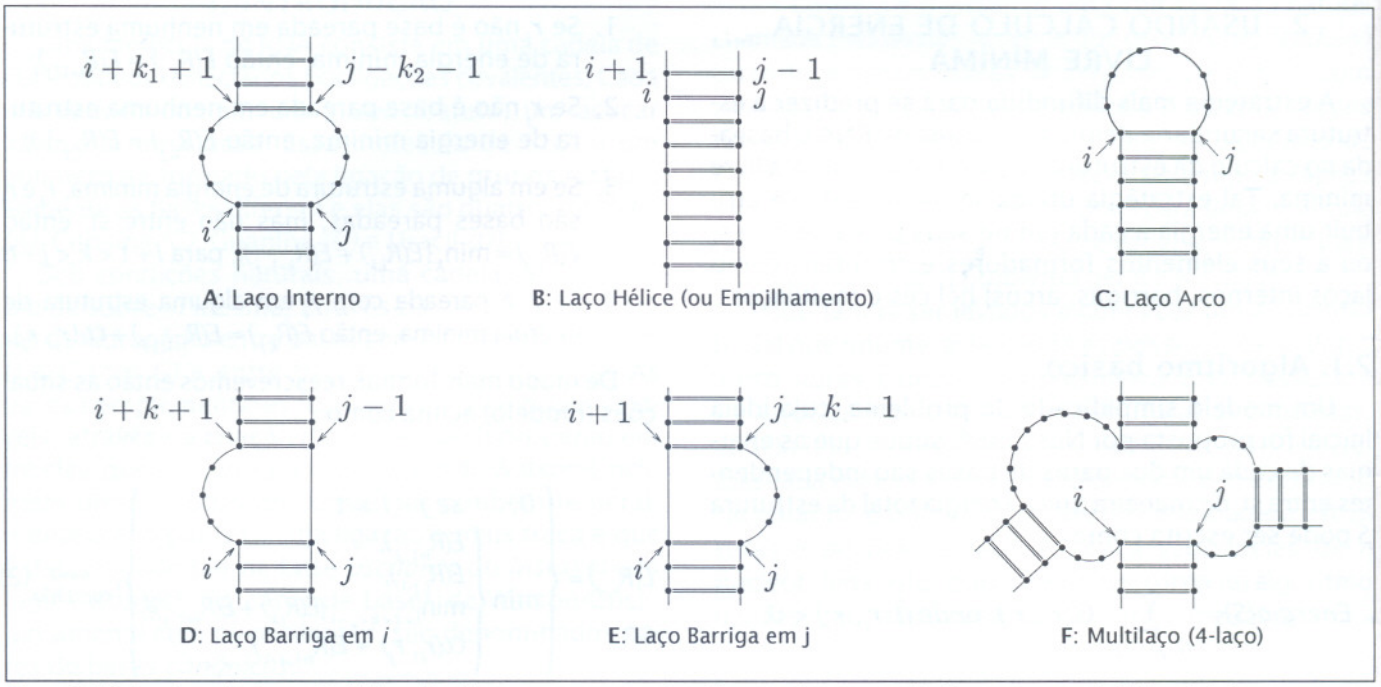

Figura 3: Vários tipos de laços. As linhas simples representam as ligações covalentes e as duplas as pontes de hidrogênio.[9]

Seja $(i,j) \in S$ e sejam i', v e j' posições tais que $i < i' < v < j' < j$. Então dizemos que:

1. v é *acessível* a (i, j) se $(i', j') \notin S$ para todo i' e j';
2. (i', j') é *acessível* a (i, j) se $(i', j') \in S$ e i' e j' são acessíveis a (i, j);
3. o conjunto formado pelas bases dos pares de bases acessíveis a (i, j) e pelas bases não pareadas – também acessíveis a (i, j) – é o *laço fechado* por (i, j), ou simplesmente *laço*.
4. o laço formado por k pares de bases (o par de fechamento (i, j) juntamente com $(k - 1)$ pares de bases acessíveis a (i, j)) e por k' bases não pareadas é chamado *k-laço* (ou *k-ciclo*) de tamanho k' fechado por (i, j).
5. uma base não pareada não pertencente a nenhum laço é uma *base externa*; um par de bases pareadas não pertencente a nenhum laço é denominado *par externo*. A coleção formada pelas bases externas e pares externos é denominado *laço externo*.

Uma estrutura secundária S induz uma decomposição de R em uma coleção de laços disjuntos $Laço_1$, $Laço_2$, ... $Laço_m$, onde m > 0, se e somente se, $S \neq \emptyset$. Energias são atribuídas aos k-laços e a energia da estrutura S passa a ser escrita como

$$Energia(S) = \sum_{i=1}^{m} \varepsilon\,(Laço_i), \qquad (3)$$

onde ε é uma função que fornece a energia de um k-laço $Laço_i$.

Para atribuir energias aos seis tipos de laços, são definidas as seguintes funções:

- $\varepsilon h(i, j)$ é a energia do laço arco fechado pelo par (i, j);
- $\varepsilon i(i, j)$ é a energia mínima de um laço interno fechado por (i, j);
- $\varepsilon bi(i, j)$ é a energia mínima de um laço barriga em i fechado por (i, j);
- $\varepsilon bj(i, j)$ é a energia mínima de um laço barriga em j fechado por (i, j);
- $\varepsilon s(i, j)$ é a energia de empilhamento de dois pares de bases adjacentes (i, j) e $(i+1, j-1)$;
- $\varepsilon m(i, j)$ é a energia mínima de um k-laço de tamanho k', com $k > 2$ (multilaço), fechado por (i, j).

Novamente usamos a estratégia de programação dinâmica para resolver o problema. Seja a seqüência $R = r_1, r_2, ..., r_n$, para a qual desejamos encontrar a estrutura secundária $S_{1,n}$ de energia livre mínima. Consideremos a subseqüência $R_{i,j} = r_i, r_{i+1}, ..., r_j$, $1 \leq i < j \leq n$, para a qual desejamos encontrar a estrutura secun-

[9] J. C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. ICUNICAMP/PWS, 1997.

dária $S_{i,j}$ de energia livre mínima. Como na modelagem anterior, há quatro possibilidades a serem tratadas, diferenciando-se apenas na situação 4:

$$L(R_{i,j}) = \min \left\{ \begin{array}{ll} 0, & se\ j - i \leq t \\ \min \left\{ \begin{array}{l} E(R_{i+1,j}), \\ E(R_{i,j-1}), \\ \min_{i<k<j} \{E(R_{i,k}) + E(R_{k+1,j})\}, \\ L(R_{i,j}) \end{array} \right. & \end{array} \right\}, \quad (4)$$

caso contrário, onde

$$L(R_{i,j}) = \{\varepsilon h(i,j), \varepsilon i(i,j), \varepsilon bi(i,j), \varepsilon bj(i,j), \varepsilon s(i,j), \varepsilon m(i,j)\}. \quad (5)$$

Podemos então caracterizar os cálculos associados a $L(R_{i,j})$ para cada uma das configurações ilustradas na Figura 3, cuja opção correspondente pode ser vista na Expressão 5. Assim sendo,

- se $L_{i,j}$ é um laço arco, então

  $L[i][j] \leftarrow L(R_{i,j}) = \varepsilon h(i,j), \zeta(j - i - 1)$.

- se $L_{i,j}$ é uma região empilhada (ou hélice), então

  $L[i][j] \leftarrow L(R_{i,j}) = \varepsilon s(i,j) = \eta + L[i+1][j-1]$.

- se $L_{i,j}$ é uma barriga em i, então

  $L[i][j] \leftarrow L(R_{i,j}) = \varepsilon bi(i,j) = \min_{k \geq 1} \{\beta(k) + L[i+k+1]\}$.

- se $L_{i,j}$ é uma barriga em j, então

  $L[i][j] \leftarrow L(R_{i,j}) = \varepsilon bj(i,j) = \min_{k \geq 1} + \{\beta(k) + L[i+1][j-k-1]\}$.

- se $L_{i,j}$ é um laço interno, então

  $L[i][j] \leftarrow L(R_{i,j}) = \varepsilon i(i,j) = \min_{k_1, k_2 \geq 1} \{\gamma(k_1 + k_2) + L[i+1+k_1][j-1-k_2]\}$.

- se $L_{i,j}$ é um multilaço, então

  $L[i][j] \leftarrow L(R_{i,j}) = \varepsilon m(i,j) = \min_{i<k<j-1} \{G[i+1][k] + G[k+1][j-1] + a\}$,

onde

$$G[i][j] = \min \left\{ \begin{array}{l} L[i][j] + b \\ \min_{i<h<j} \min \left\{ \begin{array}{l} G[i][h] + (j-h) \times c \\ G[i][h] + G[h+1][j] \\ (h - i + 1) \times c + G[h+1][j], \end{array} \right. \end{array} \right. \quad (6)$$

sendo que a, b e c são constantes onde a representa a contribuição do par de fechamento (i, j) do multilaço, b a contribuição de cada par acessível a (i, j) e c a contribuição de cada base não pareada acessível a (i, j).

As funções ζ, η, β, e γ nas expressões dadas são determinadas experimentalmente. A complexidade deste algoritmo é $O(n^4)$ e a computação (identificação) do dobramento de energia mínima $S_{1,n}$ é feita através de algoritmos recursivos.

2.3. Melhoria da eficiência em laços internos

Os laços internos são as estruturas que dominam assintoticamente no algoritmo geral apresentado na seção anterior. A partir deste fato, esforços foram empreendidos a fim de melhorar a complexidade do algoritmo como um todo, por meio da melhoria da complexidade dos laços internos.

Assumindo que a estabilidade de um laço interno depende apenas do seu tamanho, Waterman e Smith[10] mostraram como reduzir o tempo de computação do algoritmo geral para $O(n^3)$.

No entanto, o cálculo da energia associada ao laço interno fechado por (i, j) e (i', j') é determinado por quatro fatores:

- Contribuição entrópica, que depende do tamanho do laço;
- Contribuição referente ao par terminal não pareado adjacente a (i, j);
- Contribuição referente ao par terminal não pareado adjacente a (i', j');
- Penalidade associada a assimetria do laço.

Lyngs *et al.*[11] propuseram um algoritmo que executa o algoritmo geral em tempo $O(n^3)$ - otimizando o cálculo de laços internos para $O(n)$ - mas com a vantagem de incorporar todos os fatores acima.

A fim de simplificar sua descrição, adotaremos uma matriz denotada por LI para armazenar as energias associadas aos laços internos. Desta forma, a energia do laço interno fechado por (i, j) e (i', j') pode ser escrita como

$$LI[i][j] = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'-2 \geq 2}} \{L[i'][i'] + eL(i, j, i', j'), \quad (7)$$

[10] M. S. Waterman and T. F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics*, (7):455-464, 1986.
[11] R. B. Lyngs, M. Zuker, and C. N. S. Pedersen. Internal loops in RNA secondary structure prediction. In *Proc. 3rd Int. Conf. Computational Molecular Biology (RECOMB'99)*. ACM, Apr. 1999.

onde

$$eL(i, j, i', j') = \begin{cases} tamanho(i' - i + j - j' - 2) + \\ empilha(i, j) + \\ empilha(i', j') + \\ assimetria(i' - i - 1,\ j - j' - 1). \end{cases} \quad (8)$$

Papanicolaou *et al*[12] propuseram uma função para a assimetria, que adotamos na implementação tanto do algoritmo de Lyngs e Zuker quanto na do algoritmo de Waterman e Smith, que é da forma

$$assimetria(n1, n2) = \min\{K, n \times f(m)\},$$

onde $n = |n1 - n2|$ e $m = \min\{n1, n2, c\}$. As constantes K, C e a função f são assim definidas: $c = 5$, $K = 6$, $f(1) = 0.7$; $f(2) = 0.6$; $f(3) = 0.4$; $f(4) = 0.2$ e $f(5) = 0.1$. Peritz *et al.*[13] redefiniram a constante c como sendo $c = 1$. Em nossa implementação utilizamos a versão de Peritz para c.

Lyngs *et al*[14] observaram que a função *assimetria* ( ) pode ser dividida em dois termos:

$$assimetria(n1, n2) = desbalanço(n1, n2) + tamanho'(n1 + n2), \quad (9)$$

onde $desbalanço(n1, n2) = |n1 - n2|$ e *tamanho'* ( ) representa a influência que o tamanho do laço (indicado pelo valor $n1 + n2$) exerce sobre a função *assimetria* ( ).

Se definirmos $g(n1 + n2) = tamanho(n1 + 1 + n2 + 1) - tamanho'(n1 + n2)$ e observarmos que $desbalanço(n1 + 1, n2 + 1) = desbalanço(n1, n2)$, então

$$\begin{aligned} assimetria(n1 + 1, n2 + 1) &= desbalanço(n1 + 1, n2 + 1) + tamanho'(n1 + 1 + n2 + 1) \\ &= desbalanço(n1, n2) + tamanho'(n1 + 1 + n2 + 1) \\ &= desbalanço(n1, n2) + tamanho'(n1 + n2) + g(n1, n2) \\ &= assimetria(n1, n2) + g(n1 + n2). \end{aligned}$$

Esta dependência do tamanho da função *assimetria* ( ), representada por *tamanho'* ( ), pode ser movida para a função *tamanho* ( ). Em outras palavras, a variação na função *assimetria* ( ) quando variamos os valores da função *tamanho'* ( ) e mantemos constante o valor de *desbalanço* ( ) depende apenas do tamanho do laço. Esta dependência do tamanho da função de assimetria pode ser transferida para a função *tamanho* ( ), que ajuda a compor o cálculo geral da energia do laço interno (indicada no primeiro item da Expressão 8).

A partir disto, é possível então fazer a observação chave de que se fixamos o *desbalanço* ( ), a penalidade *assimetria* ( ) não se altera com o tamanho, isto é,

$$assimetria(n1 + 1, n2 + 1) = assimetria(n1, n2).$$

```
Interno (i, j)
  Para a = 0 até 1 faça
    E ← ∞
    Para l =2 - a até min {i - 1, n - j - a} faça
      aux1 ← L [i - l + 1] [j - l + 1] +
        assimetria (0,2l + a - 2) +
        empilhamento (i - l + 1, j - l + 1)
      aux2 ← L [i + a + l - 1] [j + a + l - 1] +
        assimetria (0,2l + a - 2,0) +
        empilhamento (i + a + l -1, j + a + l -1)
      E ← min {E, aux1, aux2}
      aux1 ← LI [i - l][ j + a + l ]
      aux2 ← E + tamanho (2l + a - 2) +
          empilhamento (i - l, + a + l)
      LI [i - l][ j + a + l ] ← min {aux1, aux2}
  L [i ][ j ] ← LI [i ][ j ]
```

Figura 4: Algoritmo para cálculo de laços internos conforme Lyngs e Zuker ($O(n^3)$).

Definimos a matriz *LI' [i][j][l]* como sendo a energia mínima de uma laço interno fechado por *(i, j)* de tamanho *l*.

Se a Equação 9 mantém-se, então para $l \geq 2$

$$LI'[i][j][l] = \begin{cases} LI'[i+1][j-1][l-2] + tamanho(l) - \\ tamanho(l-2) + empilha(i, j) - \\ empilha(i+1, j-1) \end{cases} \quad (10)$$

A Expressão 10 fornece a recursão necessária para computar cada entrada de *LI'* em tempo constante. Observemos que *LI'* contém $O(n^3)$ entradas e que *LI* pode ser calculado a partir de *LI'* como

$$LI[i][j] \leftarrow \min_l \{LI'[i][j][l]\}, \quad (11)$$

onde cada uma das $O(n^2)$ entradas de *LI* são computadas em tempo $O(n)$. A Figura 5 ilustra as entradas da matriz *LI* que são preenchidas pelo algoritmo.

Infelizmente, a matriz *LI'* requer espaço $O(n^3)$, o que praticamente inviabilizaria o método. Entretanto, podemos observar que precisamos de *LI' [i][j][l]* apenas em dois momentos:

[12] C. Papanicolaou, M. Gouy, and J. Ninio. An energy model that predicts the correct folding of both the tRNAs and the 5S RNA molecules. *Nucleic Acids Res.*, (21):31-44, 1984.

[13] A. E. Peritz, R. Kierzek, N. Sugimoto, and D. Turner. Thermodynamic study of internal loops in oligoribonucleotides: symmetric loops are more stable than asymmetric loops. *Biochemistry*, (30):6428-6436, 1991.

[14] R. B. Lyngs, M. Zuker, and C. N. S. Pedersen. Internal loops in RNA secondary structure prediction. In *Proc. 3rd Int. Conf. Computational Molecular Biology (RECOMB'99)*. ACM, Apr 1999.

- Quando queremos determinar se ele é um candidato a $LI[i][j]$;
- No cálculo do valor de $LI'[i-1][j+1][l+2]$.

Isto é utilizado no algoritmo representado pela Figura 4, para evitar a manutenção da matriz LI'.

2.4. Geração de soluções subótimas

Uma consideração importante que deve ser feita com relação às modelagens vistas e aos respectivos algoritmos, é que elas nos fornecem uma única solução, que pode não ser necessariamente a estrutura verdadeira. É desejável, então, que se tenha um conjunto de soluções, onde algumas delas representem valores subótimos, no que se refere à energia livre. Zuker[15] descreve um algoritmo que oferece soluções sub-ótimas.

Wuchty *et al*[16] apresentaram um algoritmo que gera todas as estruturas secundárias subótimas dentro de um intervalo energético definido pela energia livre mínima da estrutura ótima e um limite superior arbitrário.

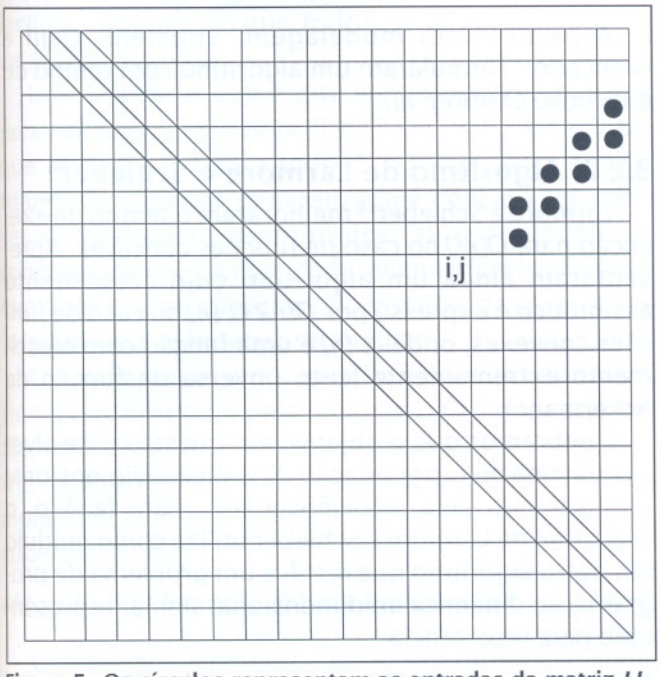

Figura 5: Os círculos representam as entradas da matriz *LI* que são preenchidas no algoritmo de Lyngs e Zuker para um dado par *(i, j)*.

3. MELHORIA DA EFICIÊNCIA PARA CLASSES ESPECIAIS DE FUNÇÕES

Com o intuito de melhorar o desempenho do algoritmo geral, alguns esforços foram empreendidos a partir de suposições acerca do comportamento das funções de desestabilização dos laços. Alguns algoritmos melhoram sensivelmente a complexidade de tempo supondo linearidade, convexidade ou concavidade destas funções. Infelizmente estas suposições não refletem de maneira confiável a realidade biológica; ou seja, estas funções na prática não são lineares, convexas ou côncavas. De qualquer modo, estes algoritmos representam avanços no campo estrito da computação.

3.1. Melhoria da eficiência para funções lineares

Caso façamos a suposição de que as funções de desestabilização g sejam lineares no tamanho k do laço, isto é, quando explicitamente fazemos $g(k) = a + bk$, então é possível reduzir o tempo de computação de barrigas e laços interiores para uma constante.

Isto leva a um algoritmo geral de complexidade de tempo $O(n^2)$[17], caso não consideremos os multilaços. Se nos cálculos levarmos em conta os multilaços, então a complexidade de tempo do algoritmo geral é $O(n^3)$. Lembremos que já estamos supondo a linearidade para os multilaços. Lembremos também que, como vimos na Seção 2.2, as expressões associadas ao cálculo de barrigas em i, barrigas em j e laços internos são

$$\varepsilon bi(i, j) = \min_{k \geq 1} \{\beta(k) + L(R_{i+k+1, j-1})\},$$

$$\varepsilon bj(i, j) = \min_{k \geq 1} \{\beta(k) + L(R_{i+1, j-k-1})\}$$

e

$$\varepsilon i(i, j) = \min_{k_1, k_2 \geq 1} \{\gamma(k_1 + k_2) + L(R_{i+1+k1, j-1-k2})\},$$

respectivamente.

Analisando a expressão acima para $\varepsilon bi(i, j)$, considerando β linear e assumindo, por conveniência, que $\beta(k) = a + b(k-1)$, temos que

$$\begin{aligned}
&\varepsilon bi(i, j) \\
&= \min\{a + L(R_{i+2, j-1}), \min_{k \geq 2} \{\beta(k) + L(R_{i+k+1, j-1})\}\} \\
&= \min\{a + L(R_{i+2, j-1}), \min_{l \geq 1} \{\beta(l+1) + L(R_{i+l+2, j-1})\}\} \\
&= \min\{a + L(R_{i+2, j-1}), \min_{l \geq 1} \{\beta(l) + L(R_{i+l+2, j-1})\} + b\} \\
&= \min\{a + L(R_{i+2, j-1}), \varepsilon bi(i+1, j) + b\}.
\end{aligned}$$

[15] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, (244):48-52, 1989.
[16] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, (49):145-165, 1999.
[17] M. S. Waterman and T. F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics*, (7):455-464, 1986.

Portanto,

$$\varepsilon bi(i, j) = \min \{a + L(R_{i+2, j-1}), \varepsilon bi(i, +1, j) + b\}. \quad (12)$$

Para barriga em j, de forma similar ao tratamento acima, obtemos que

$$\varepsilon bj(i, j) = \min \{a + L(R_{i+1, j-2}), \varepsilon bj(i, j-1) + b\}. \quad (13)$$

No caso de laços internos, a função $\varepsilon i\ (i, j)$ pode ser reescrita da forma

$$\varepsilon i\ (i, j) = \min \begin{cases} min_{k1=1, k2 \geq 1}\{\gamma\ (1 + k_2) + L(R_{i+1+1, j-1-k2})\}, \\ min_{k1 \geq 1, k2=1}\{\gamma\ (k_1 + 1) + L(R_{i+1+k1, j-1-1})\}, \\ min_{k1>1, k2>1}\{\gamma\ (k_1 + k_2) + L(R_{i+1+k1, j-1-k2})\}, \end{cases} \quad (14)$$

Os dois primeiros casos da Expressão 14 são equivalentes aos casos barriga em i (Expressão 12) e barriga em j (Expressão 13), respectivamente.

Assumindo que $\gamma\ (k) = c + d(k - 2)$, temos para o terceiro termo da Expressão 14 que

$$min_{k1>1, k2>1}\{\gamma\ (k_1 + k_2) + L(R_{i+1+k1, j-1-k2})\},$$

$$= min_{l \geq 1, k2>1}\{\gamma\ (1 + l + k_2) + L(R_{i+2+l, j-1-k2})\},$$

$$= d + \varepsilon i\ (i + 1, j),$$

Portanto, se supomos que as funções β e γ são lineares, como indicado acima, então o tempo de computação do algoritmo geral - se excluírmos os multilaços - consome tempo e espaço $O(n^2)$.

3.2. Melhoria da eficiência para funções côncavas e convexas

Dizemos que uma função $w(x, y)$ é *côncava* quando ela satisfaz a desigualdade quadrangular

$$w(i, j) + w(i', j') \leq w(i', j) + w(i, j'),$$

para todo. $i \leq i' \leq j \leq j'$. De forma similar, dizemos que $w(x, y)$ é *convexa* quando $-w(x, y)$ é côncava, ou seja:

$$w(i, j) + w(i', j') \geq w(i', j) + w(i, j').$$

3.2.1. Algoritmo de Eppstein

Proposto por Eppstein, Galil e Giancarlo[18], neste algoritmo faz-se a suposição de que não ocorrem multilaços na estrutura, bem como a função que fornece o custo (contribuição energética) de um laço, denotada por $g(k)$, onde k é o número de bases acessíveis ao laço, é convexa. Assim sendo, a Expressão (5) (excetuando-se os multilaços) pode ser reescrita como

$$V[i, j] \leftarrow E(L_{i,j}) = min\{\varepsilon h\ (i, j), C[i, j]\},$$

onde

$$C[i, j] = \min_{i < i' < j' < j} \{V[i', j'] + g((i' - i) + (j - j'))\}. \quad (15)$$

Para simplificar a apresentação do algoritmo, a recorrência (15) é modificada através da mudança de algumas variáveis. Especificamente, fazemos $\in [i, j] = C[n - i - 1, j]$, $D[i, j] = V[n - i - 1, j]$, e $w(x,y) = g(y - x)$. Daí a recorrência (15) torna-se

$$E[i, j] = \min_{\substack{0 \leq i' < i \\ 0 \leq j' < j}} \{D[i', j'] + w((i' - j', i + j)\}. \quad (16)$$

A restrição $i' < j'$ presente em (15) não está contemplada em (16). Isto pode ser tratado com a atribuição $V[i, j] \leftarrow \infty$ quando $i + j > n + 1$ ou reescrevendo a recorrência (16) da seguinte forma

$$E[i, j] = \min_{\substack{i' < i \\ j' < j \\ i' + j' > n+1}} \{D[i', j'] + w((i' + j', i + j)\}. \quad (17)$$

A partir desta modelagem, Eppstein, Galil e Giancarlo[18] formularam um algoritmo com tempo de execução $O(n^2 \log^2 n)$.

3.2.2. Algoritmo de Larmore e Schieber

Larmore e Schieber[19] melhoraram o tempo de execução para $O(n^2)$ no caso de funções côncavas. Apresentaram ainda um algoritmo cujo crescimento assintótico é expresso por $O(n2\ \alpha\ (n)$ no caso de funções convexas, onde $\alpha\ (n)$ é uma função com crescimento extremamente lento (inversa da função de Ackermann).

Lembremos que o objetivo em questão é resolver o problema de programação dinâmica bidimensional representado pela recorrência (16). Para fazê-lo, o algoritmo de Larmore e Schieber utiliza como módulo um outro algoritmo que resolve um problema de programação dinâmica unidimensional similar representado pela recorrência

$$E[i] = \min_{\substack{1 \leq i \leq n \\ 0 \leq C_1 \leq \ldots \leq C_n \leq n}} \{D[i'] + w(i', i) \mid 0 \leq i' \leq C_i\}. \quad (18)$$

Este algoritmo que resolve o problema de programação dinâmica unidimensional faz uso de duas suposições:

1. Os valores de $D[j]$ para $j = C_{i-1} + 1, \ldots, C_i$ são facilmente computados a partir dos valores de $E[i - 1]$. Por conveniência definimos $C_0 = 0$;

[18] D. Eppstein, Z. Galil, and R. Giancarlo. Speeding up dynamic programming. In *28th Symposium on the Foundations of Computer Science*, pages 488-495, 1988.

[19] L. Lamore and B. Schieber. On-line dynamic programming with applications to the prediction of RNA secondary structure. In *First ACM-SIAM Symposium on Discrete Algorithms*, pages 503-512, 1990.

2. A função w é côncava.

Para o caso de w ser convexa, Larmore e Schieber utilizam-se de outro algoritmo, desenvolvido por Klawe e Kleitman[20], para resolver a recorrência 18.

O problema de programação dinâmica unidimensional, mencionado acima, pode ser visto como um problema de busca em matriz. O algoritmo descrito por Larmore e Schieber não resolve este problema (recorrência 18) diretamente. Eles tratam-no como um problema de busca em matriz. Para que possamos enunciar este problema equivalente é necessário, antes que façamos algumas definições.

Uma matriz $n \times m$ triangular M é dita ser *monotônica totalmente côncava* se para todo $1 \leq i < i' \leq n$ e $0 \leq j \leq j' < m$, a desigualdade $M[i, j] > M[i, j']$ implica que $M[i', j] > M[i', j']$. De forma similar uma matriz $n \times m$ triangular M é dita ser *monotônica totalmente convexa* se para todo $1 \leq i < i' \leq n$ e $0 \leq j \leq j' < m$, a desigualdade $M[i, j] < M[i, j']$ implica que $M[i', j] < M[i', j']$. Para facilitar, assume-se que todos os elementos finitos de M são distintos.

Uma matriz $n \times m$ é dita ser *triangular superior generalizada* se existem $0 \leq C_1 \leq C_2 \leq C_n = m - 1$ tal que $M[i, j] = \infty$ para todo $C_i < j < m$. Uma matriz triangular superior generalizada é côncava (ou convexa) totalmente monotônica, se a condição de concavidade (ou convexidade) acima mantém-se para qualquer quatro entradas não infinitas de M, as quais formam uma submatriz retangular.

O problema de programação dinâmica unidimensional representado pela recorrência 18 pode ser, então, traduzido em um problema de busca em uma matriz triangular superior generalizada totalmente monotônica. Definimos uma matriz $n \times m$ triangular superior M por

$$M[i, i'] = D[i'] + w(i', i) \; 1 \leq i \leq n, 0 \leq i' \leq C_i. \qquad (19)$$

O resto dos elementos de M são definidos como sendo ∞. Então resolver a recorrência 18 é equivalente a encontrar o elemento mínimo em cada linha da matriz M.

É interessante notarmos que a suposição de que w é côncava traduz a condição de que M é côncava totalmente monotônica. Da mesma forma, supor que w é convexa traduz a condição imposta de que M é convexa totalmente monotônica. é interessante observarmos também que a restrição - atribuída aos elementos das colunas $C_{i-1} + 1, \ldots, C_i$ de M (que não estão definidos como ∞) - que determina que estes devem estar disponíveis apenas depois do elemento mínimo na linha $i-1$ ter sido calculado, implica que não necessitamos de todas as entradas da matriz D o tempo todo.

Vamos então definir o problema equivalente de busca em matriz. Seja M uma matriz $n \times m$ triangular superior generalizada totalmente monotônica. As linhas de M são indexadas no intervalo $1, \ldots, n$ e as colunas no intervalo $0, \ldots, m - 1$. Para cada $1 \leq i \leq n$ existe uma coluna $C_{i-1} \leq C_i < m$ tal que $M[i, j] = \infty$, para todo $j > C_i$. Queremos então encontrar o elemento mínimo em cada linha de M considerando a seguinte restrição: para $i > 1$ o valor de c_i dos elementos das colunas $C_{i-1} + 1, \ldots, C_i$ de M (que não estão definidos como ∞) estarão disponíveis apenas depois de calculado o elemento mínimo na linha $i - 1$.

Larmore e Schieber conceberam então um algoritmo linear para resolver o problema definido acima. Isto levou à formulação de um algoritmo para resolver o problema de programação dinâmica bidimensional — representado pela recorrência 16 — que consome tempo $O(n^2)$, no caso de w ser côncava.

Klawe e Kleitman[21] construíram um algoritmo para o problema *off-line* de busca em uma matriz triangular superior generalizada totalmente monotônica que consome tempo $O(n\,\alpha(n))$, onde α é a inversa da função de Ackermann. Larmore e Schieber utilizaram este algoritmo para resolver o respectivo problema *on-line* e, a partir daí, formularam um algoritmo para o problema de programação dinâmica bidimensional (recorrência 16), que consome tempo $O(n^2\alpha(n))$, no caso de w ser convexa.

4. IMPLEMENTAÇÃO E CONCLUSÕES

Desenvolvemos implementações eficientes dos algoritmos mais expressivos baseados em cálculo de energia livre mínima, tanto do ponto de vista da complexidade computacional (de tempo e espaço), quanto da representatividade do modelo termodinâmico:[22, 23, 24, 25].

[20] M.M. Klawe and D. J. Kleitman. An almost linear time algorithmfor generalizedmatrix searching. In *Technical Report RJ 6275, IBM - Research Division, Almaden Research Center*, 1988.
[21] M.M. Klawe and D. J. Kleitman. An almost linear time algorithmfor generalizedmatrix searching. In *Technical Report RJ 6275, IBM - Research Division, Almaden Research Center*, 1988.
[22] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algoritms for loop matchings. *SIAM J. appl. Math.*, (35):68-82, 1978.
[23] M. Zuker and C. D. H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. (333):333-344, 1999.a
[24] M. S. Waterman and T. F. Smith. Rapid dynamic programming algorithms for RNA secondary structure. *Advances in Applied Mathematics*, (7):455-464, 1986.
[25] R. B. Lyngs, M. Zuker, and C. N. S. Pedersen. Internal loops in RNA secondary structure prediction. In *Proc. 3rd Int. Conf. Computational Molecular Biology (RECOMB'99)*. ACM, Apr 1999.

Estas implementações tiveram como propósito possibilitar o entendimento, preciso e detalhado, destes métodos e modelos. O código C gerado, os arquivos de dados e demais arquivos necessários à execução dos programas estão disponíveis e podem ser acessados no endereço *http://www.ime.usp.br/dcc/posgrad/teses/rozante*.

Neste endereço encontra-se também disponível um arquivo no formato dvi/ps, contendo um documento com mais detalhes a respeito dos métodos, modelos e implementação em questão. Este documento foi construído sob a filosofia *literate programming* (ambiente CWEB[26]), de modo que inserimos, de forma "diluída", código C ao longo do texto principal. Ou seja, à medida em que os conceitos e métodos são apresentados, incluímos, no ponto da apresentação, o código correspondente ao conceito ou método.

A precisão dos resultados fornecidos pelos algoritmos baseados em minimização de energia tendem a melhorar, na proporção em que os parâmetros termodinâmicos se tornam mais precisos. Neste contexto, os parâmetros de energia tendem a incorporar cada vez mais casos especiais que surgem à medida em que avança o conhecimento sobre as propriedades físico-químicas dos ácidos nucléicos.

Em relação às soluções fornecidas pelos algoritmos baseados em cálculo de energia livre mínima, podemos dizer que essas soluções podem não descrever adequadamente a situação real. Em outras palavras, o modelo adotado para descrever as interações termodinâmicas pode não capturar a totalidade das situações que efetivamente ocorrem na natureza. Isto ocorre por dois motivos.

Primeiro, os parâmetros de energia com os quais os algoritmos trabalham são inevitavelmente imprecisos. Logo, a estrutura de energia livre mínima pode ser sub-ótima em relação aos parâmetros usados. O mesmo pode ocorrer em função do não conhecimento (ou não tratamento) de alguma restrição biológica que pode alterar as energias relativas, tornando/levando a uma outra estrutura sub-ótima entre as mais favoráveis. Estes fatos justificam o desenvolvimento de algoritmos que forneçam várias soluções.

Os algoritmos de Eppstein e Larmore representam importantes avanços no campo estrito da computação para o problema. No entanto, eles não representam um avanço importante do ponto de vista da contribuição biológica, pois, na natureza, as funções de desestabilização de laços não são convexas nem côncavas. Além disto, a suposição de não existência de multilaços não é razoável.

Como a suposição de linearidade é ainda mais restritiva do que a de convexidade e a de concavidade, não acreditamos que algoritmos baseados nesta suposição representem contribuições significativas para o problema.

Algoritmos baseados em minimização energética conseguem operar sobre uma única seqüência e, em relação à qual, não é necessário conhecer qualquer informação filogenética. Isto representa uma vantagem desta estratégia em relação àquela baseada em análise comparativa, já esta última, exige que se disponha de um conjunto de moléculas homólogas como entrada, o que nem sempre é possível.

No entanto, parece razoável supor que, com o tempo, o acúmulo de informação nas bases de dados de bioseqüências pode levar a um estado onde raramente, para um dada seqüência, não se disponha de um conjunto de homólogos.

Os métodos baseados em análise comparativa conseguem detectar interações terciárias (pseudonós) na estrutura. Isto representa uma vantagem desta estratégia em relação àquela baseada no cálculo de energia livre mínima. Esses métodos, geralmente, são insensíveis a pequenas variações na seqüência de nucleotídeos, enquanto nos métodos baseados em cálculo de energia livre mínima, pode-se chegar a estruturas muito diferentes a partir de seqüências que variam em poucas bases.

[26] D. E. Knuth and S. Levy. *The CWEB System of Structured Documentation*. Reading, Massachusetts: Addison-Wesley, 1993.

REFERÊNCIAS BIBLIOGRÁFICAS

PERITZ, A. E.; KIERZEK, R.; SUGIMOTO, N.; TURNER, D. Thermodynamic study of internal loops in oligoribonucleotides: symmetric loops are more stable than asymmetric loops. **Biochemistry**, (30):6428-6436, 1991.

PAPANICOLAOU, C.; GOUY, M.; NINIO, J. An energy model that predicts the correct folding of both the tRNAs and the 5S RNA molecules. **Nucleic Acids Res.**, (21):31-44, 1984.

KNUTH, D. E.; LEVY, S. **The CWEB System of Structured Documentation**. Reading, Massachusetts: Addison-Wesley, 1993.

EPPSTEIN, D.; GALIL, Z.; GIANCARLO, R. Speeding up dynamic programming. In **28th Symposium on the Foundations of Computer Science**, pages 488-495, 1988.

SANKOFF, D. Simultaneous solution of the mRNA folding, alignment and protosequense problems. **SIAM J. Appl. Math.**, (5):1-35, 1985.

RIVAS, E.; EDDY, S. A dynamic programming algorithm for RNA structure prediction including pseudoknots. **Journal of Molecular Biology**, (285):2053-2068, 1999.

SETUBAL, J. C.; MEIDANIS, J. **Introduction to Computational Molecular Biology**. ICUNICAMP/PWS, 1997.

LAMORE, L.; SCHIEBER, B. On-line dynamic programming with applications to the prediction of RNA secondary structure. In **First ACM-SIAM Symposium on Discrete Algorithms**, pages 503-512, 1990.

KLAWE, M.M.; KLEITMAN, D. J. An almost linear time algorithmfor generalizedmatrix searching. In **Technical Report RJ 6275, IBM - Research Division, Almaden Research Center**, 1988.

ZUKER, M.; TURNER, C. D. H. **Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide**. (333):333-344, 1999.

ZUKER, M. On finding all suboptimal foldings of an RNA molecule. **Science**, (244):48-52, 1989.

ZUKER, M.; SANKOFF, D. RNA secondary structures and their prediction. **Bull. Math. Biol.**, (46):591-621, 1984.

WATERMAN, M. S.; SMITH, T. F. Rapid dynamic programming algorithms for RNA secondary structure. **Advances in Applied Mathematics**, (7):455-464, 1986.

LYNGS, R. B.; PEDERSEN, C. N. S. Pseudoknots in RNA secondary structure. In **Proc. 4rd Int. Conf. Computational Molecular Biology** *(RECOMB'00)*. ACM, Apr 2000.

NUSSINOV, R.; PIECZENIK, G.; GRIGGS, J. R.; KLEITMAN, D. J. Algoritms for loop matchings. **SIAM J. appl. Math.**, (35):68-82, 1978.

LYNGS, R. B.; ZUKER, M.; PEDERSEN, C. N. S. Internal loops in RNA secondary structure prediction. In PROC. **3rd Int. Conf. Computational Molecular Biology** *(RECOMB'99)*. ACM, Apr 1999.

WUCHTY, S.; FONTANA, W.; HOFACKER, I. L.; SCHUSTER, P. Complete suboptimal folding of RNA and the stability of secondary structures. **Biopolymers**, (49):145-165, 1999.