MO640/MC668

Guilherme P. Telles

IC-Unicamp

Avisado está

- Estes slides são incompletos.
- Estes slides contêm erros.

Parte I

Alinhamento múltiplo de cadeias

Alinhamento múltiplo de seqüências

- Generalização do alinhamento para duas seqüências.
- Um alinhamento múltiplo entre as cadeias $s_1, \ldots, s_k$ é obtido inserindo espaços nas cadeias de tal forma que elas tenham o mesmo tamanho e nenhuma coluna do alinhamento tenha apenas espaços.

```
M-QPIALLLG
M-LR-ALL-G
M-K-IALLLG
MWPPVA--LG
```

Motivação

- Identificar blocos conservados e padrões em um conjunto de cadeias que são biologicamente relacionadas.
- Analisar história evolutiva.
- Caracterizar famílias de moléculas.

Pontuação

- Vamos considerar que pontuação do alinhamento é dada pela soma das pontuações das colunas.
- Para cada coluna, uma função de pontuação teria que mapear colunas nos reais. O domínio é grande $((|\Sigma| - 1)^k/(|\Sigma| - 1)!)$, logo não é simples definir.

Sum-of-pairs

- Um esquema bastante usado é o sum-of-pairs, SP.
- Simples.
- Satisfaz
  - Independência da ordem dos aminoácidos nas colunas.
  - Valoriza a presença de aminoácidos iguais ou parecidos.
  - Penaliza a presença de buracos e aminoácidos pouco relacionados.
- Usa outros esquemas de pontuação (PAM, BLOSUM, etc) como base.

Alinhamento induzido

- Seja $\mathcal{A}$ um alinhamento múltiplo.
- Um alinhamento $\mathcal{A}_{ij}$ induzido por $\mathcal{A}$ é o alinhamento entre s_i e s_j que resulta da remoção das linhas correspondetes às cadeias $s_k \neq s_i, s_j$ de $\mathcal{A}$.
- $\mathcal{A}_{ij}$ também é chamado de projeção do alinhamento múltiplo.

SP

- A pontuação *SP* de uma coluna k de um alinhamento múltiplo é a soma das pontuações de todos os pares de símbolos na coluna k.
- Por exemplo:

$$\begin{aligned} SP(I,-,I,V) &= p(I,-)+p(I,I)+p(I,V)+ \\ & \quad p(-,I)+p(-,V)+p(I,V) \end{aligned}$$

- A pontuação *SP* de um alinhamento é a soma das pontuações *SP* das colunas dele.
- Se $p(-,-)=0$, então a pontuação do alinhamento $\mathcal{A}$ satisfaz

$$SP(\mathcal{A}) = \sum_{i<j} score(\mathcal{A}_{ij}).$$

Programação dinâmica

- A solução por PD para o problema com duas cadeias pode ser naturalmente generalizada para k cadeias. Supondo que $|s_1| = |s_2| = \ldots = |s_k| = n$,
  - A matriz tem tamanho $O(n^k)$.
  - Cada entrada depende de $2^k - 1$ outras, uma para cada composição possível da coluna, considerando espaço ou caractere de cada cadeia e excluindo o caso com k buracos.
  - Para computar o SP faz-se uma soma de $O(k^2)$ elementos.
  - É preciso calcular o máximo para chegar ao valor de $A[i_1, \ldots, i_k]$.
  - $O(n^k 2^k k^2)$, NP-completo.

Programação dinâmica

- Fazendo $\mathbf{s} = s_1, \ldots, s_k$, $\mathbf{i} = i_1, \ldots, i_k$, e sendo $\mathbf{b}$ um vetor binário de tamanho k, a expressão geral para preenchimento da matriz é

$$A[\mathbf{i}] = \max_{b \neq \mathbf{0}} \{A[\mathbf{i} - \mathbf{b}] + SP(col(\mathbf{s}, \mathbf{i}, \mathbf{b})\}$$

onde $col(\mathbf{s}, \mathbf{i}, \mathbf{b}) = (c_1, \ldots, c_k)$ e

$$c_j = \begin{cases} s_j[i_j] & \text{se} \quad b_j = 1 \\ - & \text{se} \quad b_j = 0 \end{cases}$$

para $1 \leq j \leq k$.

Heurística de projeções

- Dado um alinhamento múltiplo $\mathcal{A}$, a projeção $\mathcal{A}_{ij}$ é o alinhamento entre s_i e s_j induzido por $\mathcal{A}$ com as colunas $(-,-)$ removidas.
- Se conhecemos o alinhamento múltiplo ótimo entre várias cadeias e olhamos para a projeção das cadeias 1 e 2 então
  - o alinhamento ótimo entre 1 e 2 e o alinhamento projetado entre elas não precisa ser igual, mas
  - a pontuação do alinhamento ótimo entre 1 e 2 é maior ou igual à pontuação da projeção.

Heurística de projeções

- Usa um limite inferior L da pontuação do alinhamento ótimo para preencher apenas células relevanes da matriz.
- O limite L pode ser a pontuação de um alinhamento qualquer (não-ótimo).
- Células relevantes são células que melhoram a pontuação do alinhamento em relação a L.
- Funciona apenas para pontuação SP.
- É uma heurística porque no pior caso preenche toda a matriz.

Projeção vs. alinhamento ótimo

```
1 ...AT...
2 ...A-...
3 ...-T...
4 ...AT...
5 ...AT...

2 ...A-...
3 ...-T...

2 ...A...
3 ...T...
```

Heurística de projeções

- A heurística de projeções limita o preenchimento da matriz a uma região do hipercubo que pode conter o alinhamento ótimo.
- As células preenchidas são aquelas que são percorridas por pelo menos um caminho que tem, em todas as projeções dele, custo menor que um limite.

Matriz de cortes

- Usando PD podemos construir a matriz de similaridades de prefixos e também a de sufixos.
- Somando as duas matrizes obtemos a matriz C.
- Cada célula de C guarda a melhor pontuação de um alinhamento que contém um $corte(i, j)$.
- Um $corte(i, j)$ em um alinhamento ótimo $\mathcal{A}$ é a divisão de $\mathcal{A}$ em dois sub-alinhamentos,

$$\mathcal{A}_1 : s_1[1..i], s_2[1..j]$$
$$\mathcal{A}_2 : s_1[i+1..|s_1|], s_2[j+1..|s_2|]$$

- $\mathcal{A}_1$ é o melhor alinhamento possível dos prefixos e $\mathcal{A}_2$ é o melhor alinhamento possível dos sufixos definidos pelo corte.

Matriz de prefixos

		G	A	T	T	C
	0	-2	-4	-6	-8	-10
A	-2	-1	-1	-3	-5	-7
T	-4	-3	-2	0	-2	-4
T	-6	-5	-4	-1	1	-1
C	-8	-7	-6	-3	-1	2
G	-10	-7	-8	-5	-3	0
G	-12	-9	-8	-7	-5	-2

Matriz de sufixos

	G	A	T	T	C	
A	-2	0	-3	-6	-9	-12
T	-5	-3	-1	-4	-7	-10
T	-6	-4	-3	-2	-5	-8
C	-7	-5	-3	-4	-3	-6
G	-6	-6	-4	-2	3	-4
G	-7	-7	-5	-3	-1	-2
	-10	-8	-6	-4	-2	0

Matriz de cortes

		G	A	T	T	C
	-2	-2	-7	-12	-17	-22
A	-7	-4	-2	-7	-12	-17
T	-10	-7	-5	-2	-7	-12
T	-13	-10	-7	-5	-2	-7
C	-14	-13	-10	-5	-4	-2
G	-17	-14	-13	-8	-4	-2
G	-22	-17	-14	-11	-7	-2

Teste de relevância

- Vamos supor que conhecemos um limite inferior L para a pontuação de um alinhamento múltiplo ótimo.
- Teorema: Seja $\mathcal{A}$ um alinhamento ótimo entre $s_1, \ldots, s_k$. Se $SP(\mathcal{A}) \geq L$ então

$$score(\mathcal{A}_{ij}) \geq L - \sum_{x<y,(x,y)\neq(i,j)} sim(s_x, s_y).$$

Teste de relevância

- Vamos denotar

$$L - \sum_{x<y,(x,y)\neq(i,j)} sim(s_x, s_y) = L_{ij}.$$

- L_{ij} é um limite inferior para a pontuação da projeção.
- A célula $\mathbf{i} = (i_1, \ldots, i_k)$ é relevante para o alinhamento ótimo com relação ao limite L se todas as suas projeções satisfazem ao teorema.
- Em outras palavras, $\mathbf{i}$ é relevante se

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

para todo x e y tais que $1 \leq x < y \leq k$.

Implementação

- A implementação eficiente da heurística começa da célula $\mathbf{0}$ e preenche as células relevantes por propagação a partir de $\mathbf{0}$.
- Uma célula $\mathbf{i}$ influencia uma célula $\mathbf{j}$ se $\mathbf{i}$ é usada para computar o valor de $a[\mathbf{j}]$. Nesse caso dizemos também que $\mathbf{j}$ depende de $\mathbf{i}$.
- As células são mantidas em uma fila de prioridades pela ordem lexicográfica das células.
  - Uma célula entra na fila quando depende de uma que já está na file e é inicializada nesse momento.
  - Quando uma célula sai da fila seu valor ótimo já foi calculado e todas as que dependem dela diretamente são atualiadas.

Heurística

$\textsc{Align}(s_1, \ldots, s_k, L)$

```
1  for every x and y, 1 ≤ x < y ≤ k
2        Compute C_xy, the total score array for s_x and s_y
3  for every x and y, 1 ≤ x < y ≤ k
4        L_xy = L − Σ_{i<j,(x,y)≠(i,j)} sim(s_i, s_j)
5  pool = {0}
6  while pool ≠ ∅
7        i = remove the lexicographically smallest cell in pool
8        if C_xy[i_x, i_y] ≥ L_xy, ∀x, y, 1 ≤ x < y ≤ k
9              for every j dependent on i
10                   if j ∉ pool
11                         pool = pool ∪ j
12                         a[j] = a[i] + SP(col(s, i, j − i))
13                   else
14                         a[j] = max(a[j], a[i] + SP(col(s, i, j − i)))
15 return a[n_1, ..., n_k]
```

Heurística

- Pára quando $A[n_1, \ldots, n_k]$ for retirada do pool.
- A complexidade de tempo e espaço são proporcionais ao número de células relevantes. Quanto melhor for L, menor o número de células.
- Uma possibilidade para obter L é a pontuação de um alinhamento qualquer entre as cadeias.
- Para obter o alinhamento é necessário registrar as atualizações das células. Uma maneira possível é usando um grafo.

Alinhamento por heurística em árvore

- Seja $S = \{s_1, \ldots, S_k\}$ um conjunto de cadeias e seja T uma árvore com k nós em que cada nó está rotulado por uma cadeia distinta de S.
- Um alinhamento múltiplo de S, $\mathcal{M}$, é consistente com T se a projeção de todo par de cadeias s_i, s_j tem pontuação igual a $sim(s_i, s_j)$.

Alinhamento por heurística em árvore

- A heurística constrói o alinhamento entre duas seqüências adjacentes e depois acrescenta uma outra seqüência adjacente, fazendo um alinhamento de duas sequencias e incorporando no bloco
- Um alinhamento múltiplo de S será consistente com T.

Alinhamento em árvore

- Encontrar a seqüência que maximiza a pontuação é NP-completo.

Heurística: alinhamento estrela

- Se não tivermos a árvore, podemos usar o alinhamento estrela: a árvore é uma estrela.
- O centro é escolhido como a cadeia que maximiza a soma das similaridades com as demais.
- (Ou usamos cada uma como centro e ficamos com o melhor alinhamento.)
- Não é difícil mostrar que o alinhamento estrela para pontuação que é uma distância (métrica) obtém uma pontuação que é no máximo 2 vezes maior que a pontuação do alinhamento ótimo.

Soluções práticas

- mátodos progressivos.
- métodos iterativos.

Alinhamento progressivo

- Simples, requer poucos recursos, guloso e sensível ao esquema de pontuação.
- Três passos principais: calcular alinhamentos par-a-par, construir uma árvore para as cadeias e depois usa a árvore para alinhar cadeias ou alinhamentos.
- Os algoritmos diferem na implementação de cada um desses passos.

Alinhamento de alinhamentos

- Pode ser resolvido por uma aplicação do algoritmo de programação dinâmica.
- A pontuação entre duas colunas pode ser calculada pela soma-de-pares para a nova coluna.
- Para k colunas a complexidade é $O(nmk^2)$.

Alinhamento iterativo

- Consiste em refinar um alinhamento existente em várias iterações.
- Uma parte dos algoritmos nessa classe retira cada cadeia do conjunto, refaz os alinhamentos de depois combina todos eles.
- Requer muito tempo e depende de outros métodos para produzir alinhamentos múltiplos.

Heurística: CLUSTAL W

- Heurística para alinhamento em árvore.
- Constrói a árvore usando neighbor-joining a partir de distâncias dadas por alinhamentos de pares de cadeias.
- Varia o esquema de pontuação e penalidades de buracos para lidar melhor com cadeias com divergência variada.

Heurística: CLUSTAL W

- Três passos principais:
  1. Construção da matriz de distâncias entre as cadeias.
  2. Construção da árvore.
  3. Construção do alinhamento usando a árvore como guia. Cadeias muito divergentes (menos de 40% de similaridade) são deixadas por último.

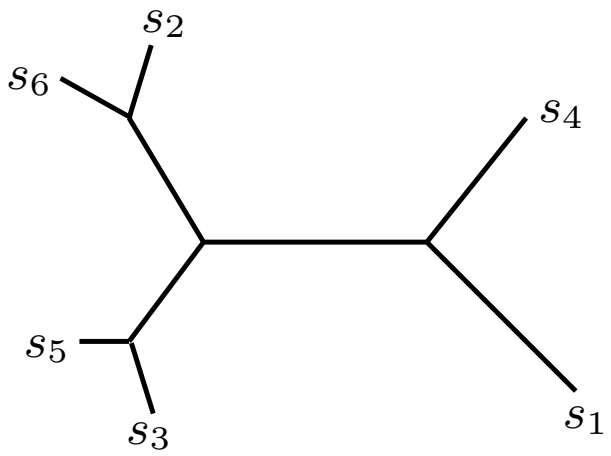
Matriz de distâncias

- Cálculo das pontuações entre as cadeias. A pontuação é o número de identidades no melhor alinhamento dividido pela soma dos tamanhos das cadeias. Dois métodos de contagem:
  1. Pontuação da PD usando uma matriz de substituições, penalidade para abrir um novo buraco e penalidade para estender um buraco ou
  2. contagem de cadeias de colunas com letras iguais (1 ou 2 para proteínas, 2 a 4 para DNA) subtraindo uma penalidade para cada buraco (mais rápido).
- A pontuação é dividida por 100 e subtraída de 1, se tornando uma distância.

Construção da árvore

- A árvore é construída usando neighbor-joining.
- A idéia básica é
  - Começar com uma estrela com todas as cadeias como folhas.
  - Acrescentar um ancestral entre as folhas s_i e s_j mais próximas entre si que das demais e estimar o tamanho dos ramos.
  - Remover s_i e s_j e acrescentar s_{ij}.
  - Repetir até que restem apenas três grupos.
  - Para três grupos, construir a árvore diretamente.
- A árvore não tem raiz. Uma raiz é colocada no ponto em que a média dos tamanhos dos ramos de ambos os lados é igual.
- O alinhamento é construído das folhas em direção à raiz, nós mais próximos primeiro.

Árvore



Produção do alinhamento múltiplo

- As cadeias ou alinhamentos já existentes são alinhadas usando PD com uma matriz de substituições, uma penalidade para abrir buracos e uma para estender buracos.
- As matrizes são reescaladas para ter como menor pontuação o valor 0.
- A pontuação entre duas posições que são colunas de alinhamentos é calculada pela somas dos scores de cada par ponderado pelo peso das cadeias dividida pelo número de cadeias.
- Todo par com buraco recebe pontuação 0.

Pesos

- Os pesos são calculados normalizando as distâncias na árvore de tal forma que a maior distância seja 1.0.
- O peso altera o esquema de pontuação de tal forma que cadeias que são muito próximas de alguma outra recebam peso pequeno e cadeias que são pouco relacionadas com outras recebem peso maior.

Penalidades iniciais de buracos

- Os valores de penalidades (GOP e GEP) para buracos iniciais são escolhidos pelo usuário.
- Os valores iniciais são alterados em função:
  - da matriz de pontuação (valor de mismatch médio p)
  - da similaridade entre as cadeias (o percentual de identidade entre as cadeias ou grupos s)
  - do tamanho das cadeias (logaritmo do tamanho da menor cadeia ou da diferença entre os tamanhos)

$$GOP = (GOP_i + \log(\min(m, n))) * p * s.$$

$$GEP = (GEP_i + (1.0 + \log(\min(m/n)))).$$

Penalidades de buracos por posição

- Os valores de penalidades GOP_i e GEP_i são usados para gerar uma matriz de penalidades de buracos modificadas em função da posição em que o buraco aparece.
- Se já existe um buraco na posição, a penalidade é reduzida:

$$GOP = GOP * 0.3 * \frac{\text{no de seqs sem buraco}}{\text{no de seqs}}$$
$$GEP = \frac{GEP}{2}$$

Penalidades de buracos por posição

- Se não existem buraco próximos a uma posição, a penalidade é aumentada:

$$GOP = GOP * (2 + (8 - \text{distância de algum buraco} * 2)/8)$$

- Se não houver buracos e houver uma cadeia de aminoácidos hidrofílicos de tamanho pelo menos cinco em alguma seqüencia, a penalidade é reduzida.

$$GOP = \frac{GOP}{3}$$

Penalidades de buracos por posição

- Se não houver buracos e não houver uma cadeia de aminoácidos hidrofílicos (D, E, G, K, N, Q, P, R e S)de tamanho pelo menos cinco em alguma seqüencia, a penalidade é modificada em função de um valor específico por aminoácido ou pela média dos valores na coluna.

Matrizes de substituição

- As matrizes de substituição podem ser PAM ou BLOSUM. A matriz é escolhida em função da distância entre as cadeias ou grupos de cadeias.

80-100%	PAM20
60-80%	PAM60
40-60%	PAM120
0-40%	PAM350

80-100%	BLOSUM80
60-80%	BLOSUM62
30-60%	BLOSUM45
0-30%	BLOSUM30