Algorithms in Bioinformatics

生物信息学算法原理

Chaochun Wei (韦朝春)

ccwei@sjtu.edu.cn

http://cbb.sjtu.edu.cn/~ccwei

Jing Li （李婧）

Fall 2014

Course organization

- Introduction (Week 1-2)
  - Course introduction
  - A brief introduction to molecular biology
  - A brief introduction to sequence comparison
- Part I: Algorithms for Sequence Analysis (Week 3 - 11)
  - Chapter 1-3, Models and theories
    - Probability theory and Statistics (Week 4)
    - Algorithm complexity analysis (Week 5)
    - Classic algorithms (Week 6)
    - Lab: Linux and Perl
  - Chapter 4, Sequence alignment (week 7)
  - Chapter 5, Hidden Markov Models (week 8)
  - Chapter 6. Multiple sequence alignment (week 10)
  - Chapter 7. Motif finding (week 11)
  - Chapter 8. Sequence binning (week 11)
- Part II: Algorithms for Network Biology (Week 12 - 16)

Grading

- Homework 40%
- Projects 10%
- Exam 50%

Smith/Waterman local alignment (1981)

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- **Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to x_i and $Y_{1\ldots j}$ of Y up to Y_j ($0 \le i \le n$, $0 \le j \le m$),** *then we have*

$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Probability theory

for biological sequence analysis

- Applications
  - BLAST significance tests
  - The derivation of BLOSUM and PAM scoring matrices
  - Position Weight Matrix (PWM or PSSM)
  - Hidden Markov Models (HMM)
  - Maximum likelihood methods for phylogenetic trees

- Definition
  - $P_i \geq 0; \sum_i P_i = 1$
  - $f(x) \geq 0; \int_{-\infty}^{+\infty} f(x)dx = 1$
- Examples:
  - A fair dice: $P_i = 1/6, i = 1,2,\ldots,6.$
  - A random nucleotide sequence: $P_A = P_C = P_G = P_T = 1/4$
- "i.i.d.": independent, identically distributed

Chapter 2:
Algorithm Complexity Analysis

Chaochun Wei

Fall 2014

Order of growth

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3,4, 5,6

Insertion sort:

Algorithm run time complexity: $O(N^2)$
Order of growth: 2

O-notation (big-O notation): Asymptotic upper bound

$O(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \le f(n) \le c\, g(n)$ for all $n \ge n_0\}$

Note about O-notation operations:
$O(k_1 * N^2 + k_2 * N^3) = O(N^3)$ for constants k_1, k_2

O-notation (big-O notation):
Asymptotic upper bound

Example: Sort an array of numbers
5, 2, 4, 6, 1, 3 → 1, 2, 3,4, 5,6

Insertion sort:
algorithm time complexity: $O(N^2)$

Sorting with time complexity of O(N*logN)

Example: Sort an array of numbers

5, 2, 4, 6, 1, 3 → 1, 2, 3,4, 5,6

```
Sort (A)
for j = 2 to length(A)
      do key = A[j]
            /*Use binary search to insert A[j]
            /*into the sorted sequence A[1…j-1]
            i=j-1

            Binary_search(A[j], A[1…j-1],)
```

Sorting

Example: Sort an array of numbers

5, 2, 4, 6, 1, 3 → 1, 2, 3,4, 5,6

There are a lot of sorting algorithms:

- Heap sort (O(N*logN))
- Merge sort (O(N*logN))
- *Quick sort (worst-case $O(N^2)$, average O(N*logN))

Merge sort

```
Merge-Sort (A, p, r)
        if p<r
                then q=[(p+r)/2]
                        Merge-Sort(A, p, q)
                        Merge-Sort(A,q+1,r)
                        Merge(A, p, q, r)
```

Time Complexity: $$T(N) = \begin{cases} O(1); if\ N = 1 \\ 2T(N/2) + O(N); if\ N > 1 \end{cases}$$

Solve it: T(N) = O(N*logN)

Space complexity

Example: Sort an array of numbers

5, 2, 4, 6, 1, 3 → 1, 2, 3,4, 5,6

Need an array of size N: A[1…N], and 3 temporary variables
O(N)

Example: Sequence alignment

Need a two-dimension array of size N*M, and a constant number of temporary variables
O(N*M) or O(max(N, M))

Chapter 3:
Dynamic Programming

Chaochun Wei

Fall 2014

Smith/Waterman local alignment (1981)

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- **Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to x_i and $Y_{1\ldots j}$ of Y up to Y_j ($0 \leq i \leq n$, $0 \leq j \leq m$)**, *then we have*

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} 0 \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Local alignment

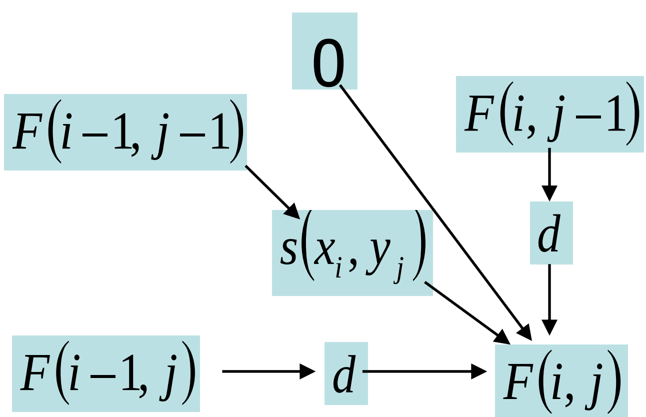
- Two differences with respect to global alignment:
  - No score is negative.
  - Traceback begins at the highest score in the matrix and continues until you reach 0.
- Global alignment algorithm: *Needleman-Wunsch*.
- Local alignment algorithm: *Smith-Waterman*.

A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
A				
G				
C				

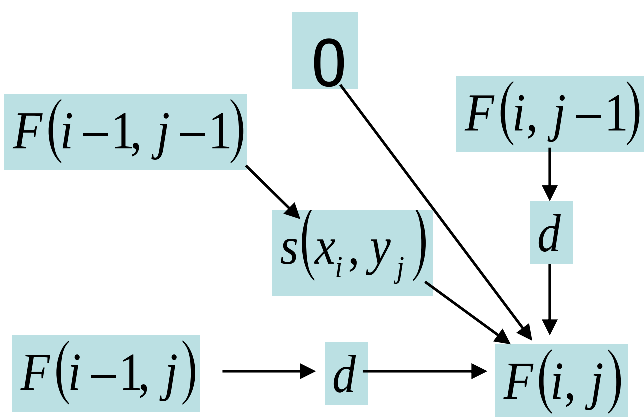


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	0	0	0
A	0			
G	0			
C	0			

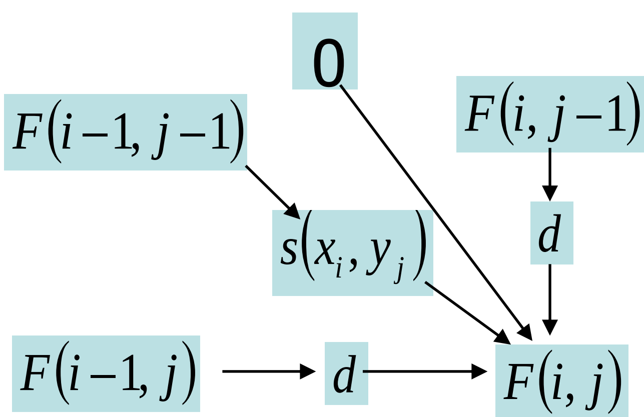


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

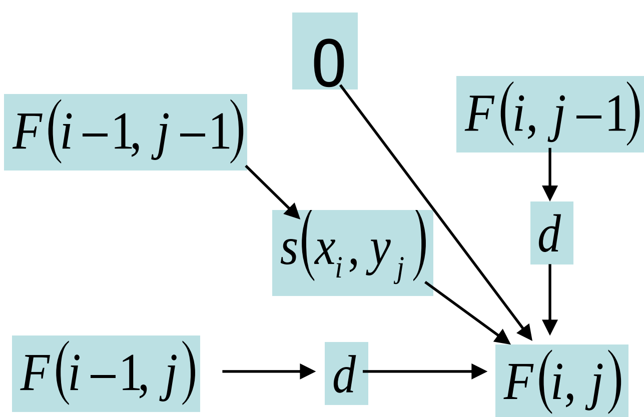


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0



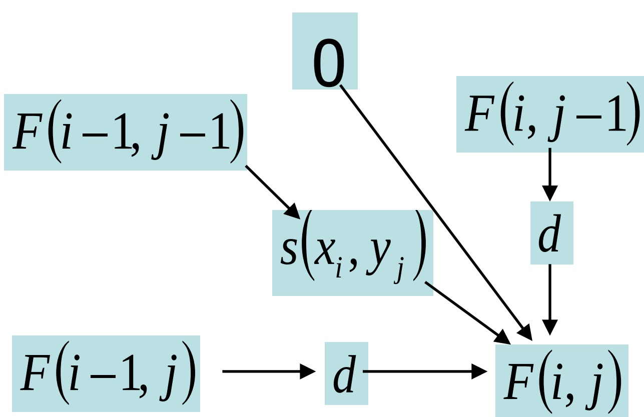
AG
AG

Local alignment

Find the optimal local alignment of AAG and GAAGGC.
Use a gap penalty of d=-5.

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

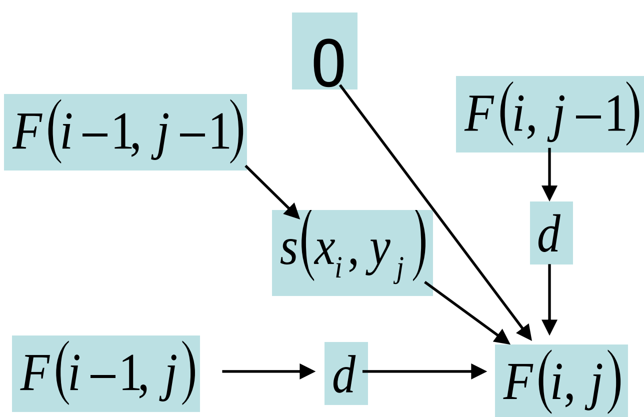
		A	A	G
	0	0	0	0
G	0			
A	0			
A	0			
G	0			
G	0			
C	0			



Local alignment

Find the optimal local alignment of AAG and GAAGGC.
Use a gap penalty of d=-5.

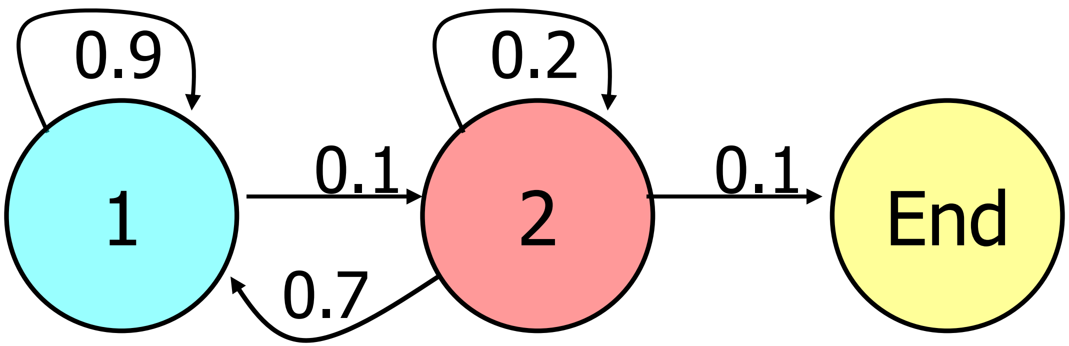
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



		A	A	G
	0	0	0	0
G	0	0	0	2
A	0	2	2	0
A	0	2	4	0
G	0	0	0	6
G	0	0	0	2
C	0	0	0	0

Hidden Markov Model

HMM for two biased coins flipping



$$e_1(H)=0.8, e_1(T)=0.2, e_2(H)=0.3, e_2(T)=0.7$$

TTHHTTHTTTTTHTHHHHHTHTH **Observed sequence x**

\- -

11221111122222111111112222 **Hidden state sequence** π

$$\pi^* = \arg\max_{\pi} P(x,\pi)$$

Hidden Markov Model

- Elements of an HMM (N, M, A, B, Init)
  1. N: number of states in the model
     - $S=\{S_1, S_2, \ldots, S_N\}$, and the state at time t is q_t.
  2. M: alphabet size (the number of observation symbols)
     - $V=\{v_1, v_2, \ldots, v_M\}$
  3. A: state transition probability distribution
     - $A=\{a_{ij}\}$ where $a_{ij}=P[q_{t+1}=S_j|q_t=S_i]$, $1 \leq i,j \leq N$
  4. E: emission probability
     - $E=\{e_j(k)\}$ (observation symbols probability distribution in state j), where $e_j(k)=P[v_k \text{ at } t \mid q_t = S_j\}$, $1 \leq j \leq N$, $1 \leq k \leq M$
  5. Init: initial state probability
     - $Init=\{I_i\}$, where $I_i=P[q_1=S_i]$, $1 \leq i \leq N$.

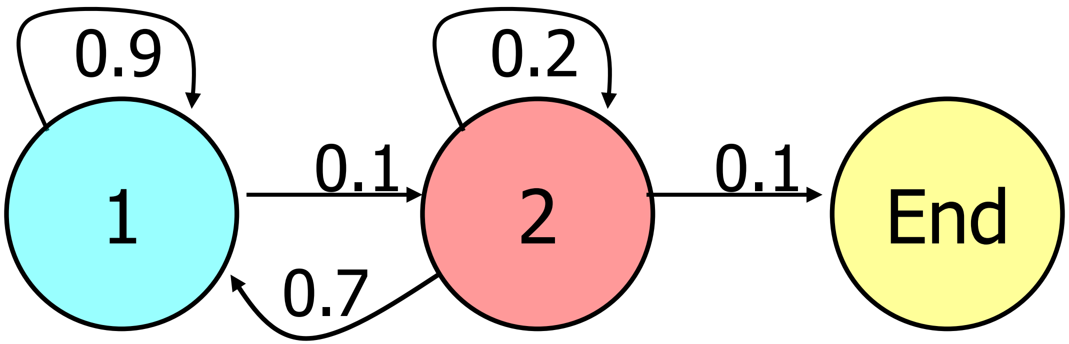
HMM is a generative model

- HMM can be used as a generator to produce an observation sequence $O=O_1O_2\ldots O_T$, where each O_t is one of the symbols from V, and T is the number of observations in the sequence.

1. Choose an initial state $q_1=S_i$ according to Init;
2. Set t=1;
3. Choose $O_t=v_k$ according to $e_i(k)$ (the symbol probability distribution in state S_i);
4. Transit to a new state $q_{t+1}=S_j$ according to a_{ij};
5. Set t=t+1; return to step 3 if t<T; otherwise terminate the procedure.

HMM is a generative model

HMM for two biased coins flipping



$$e_1(H) = 0.8, e_1(T) = 0.2, e_2(H) = 0.3, e_2(T) = 0.7$$

TTHHTTHTTTTTHTHHHHTHTH **Observed sequence x**

1122111112222111111112222 **Hidden state sequence π**

$$P(x, \pi \mid \lambda) = Init_{\pi_0} * e_{\pi_0}(x(0)) * \prod_{0 \le i \le T} (a_{\pi_i \pi_{i+1}} e_{\pi_{i+1}}(x(i)))$$

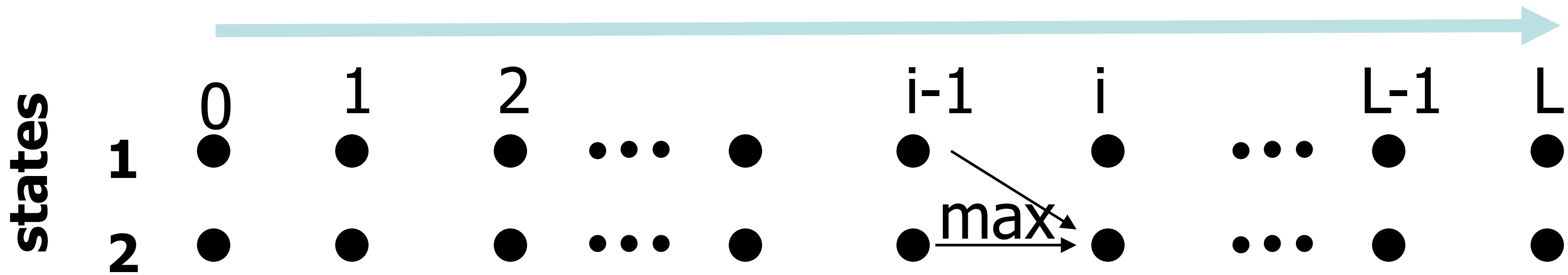
Hidden Markov Model

- HMM： $\lambda=\{A, B, Init\}$

Three basic problems for HMMs

- Problem 1: Given the observation sequence $O=O_1O_2\ldots O_T$, and a model $\lambda=\{A, B, Init\}$, how to compute $P(O|\lambda)$, the probability of the observation sequence given the model?
- Problem 2: Given the observation sequence $O=O_1O_2\ldots O_T$, and a model $\lambda=\{A, B, Init\}$, how to choose a corresponding state sequence $Q=q_1q_2\ldots q_T$, which is optimal in some meaningful sense..
- Problem 3: how to estimate model parameters $\lambda=\{A, B, Init\}$ to maximize $P(O|\lambda)$.

Most Probable Path and Viterbi Algorithm



Let $f_l(i) = \max_{\{\pi_0,\ldots,\pi_{i-1}\}} (\Pr(x_0,\ldots,x_{i-1},x_i,\pi_0,\ldots,\pi_{i-1},\pi_i = l))$

Initialization (i=1…L) $f_0(i) = \pi_i e_i(x_0)$

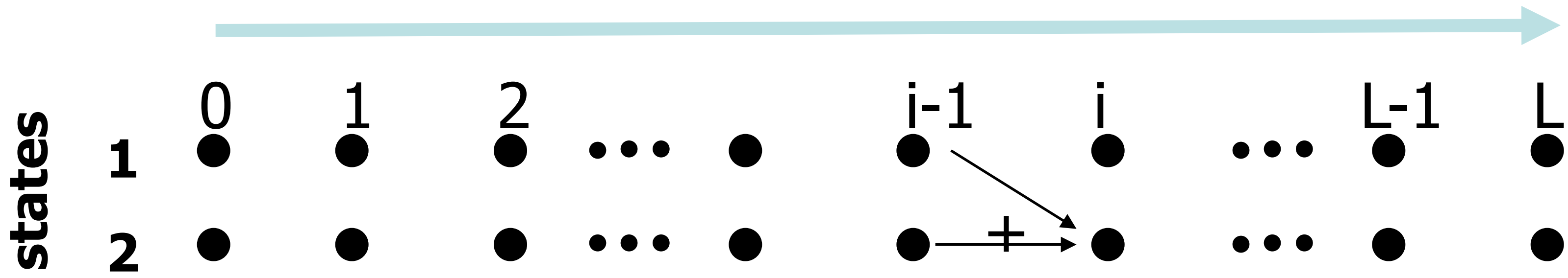
Recursion (i=1…L)

$$f_l(i) = e_l(x_i)\max_k(f_k(i-1)a_{kl});$$
$$ptr_i(l) = \arg\max_k(f_k(i-1)a_{kl}).$$

Time complexity $O(N^2L)$ space complexity $O(NL)$

Solution to problem 2

Probability of All the Possible Paths and Forward Algorithm



Let $f_l(i) = \Pr(x_0, \ldots, x_i, \pi_i = l)$

Initialization (i=1…L) $f_0(i) = \pi_i e_i(x_0)$

Recursion (i=1…L) $f_l(i) = e_l(x_i) \sum_k (f_k(i-1) a_{kl})$

Probability of all the probable paths $P(x) = \sum_\pi P(x, \pi) = \sum_k f_k(L)$

Solution to problem 1

Questions？

- Email： ccwei@sjtu.edu.cn
- Office Time： 12:30-14:30
  - Sunday(January 4, 2015)
  - Thursday (January 8, 2015)