Sequence Alignment

Xiaohui Xie

University of California, Irvine

Pairwise sequence alignment

- Example: Given two sequences: $S = \texttt{ACCTGA}$ and $T = \texttt{AGCTA}$, find the minimal number of edit operations to transform S to T.
- Edit operations:
  - Insertion
  - Deletion
  - Substitution

Biological Motivation

- Comparing or retrieving DNA/protein sequences in databases
- Comparing two or more sequences for similarities
- Finding patterns within a protein or DNA sequence
- Tracking the evolution of sequences
- ...

Pairwise alignment

- Definition: An alignment of two sequences S and T is obtained by first inserting spaces ($'-'$) either into, before or at the ends of S and T to obtain S' and T' such that $|S'| = |T'|$, and then placing S' on top of T' such that every character in S' is uniquely aligned with a charater in T'.
- Example: two aligned sequences:

```
S: GTAGTACAGCT-CAGTTGGGATCACAGGCTTCT
   |||| || ||| ||||||   ||||||  |||
T: GTAGAACGGCTTCAGTTG---TCACAGCGTTC-
```

Similarity measure

- $\sigma(a, b)$ - the score (weight) of the alignment of character a with character b, where $a, b \in \Sigma \cup \{'-'\}$ wher $\Sigma = \{'\mathtt{A}', '\mathtt{C}', '\mathtt{G}', '\mathtt{T}'\}$. For example

$$\sigma(a,b) = \begin{cases} 2 & \text{if } a = b \text{ and } a, b \in \Sigma \\ 0 & \text{if } a \neq b \text{ and } a, b \in \Sigma \\ -1 & \text{if } a \neq b \text{ and } a =' -' \text{ or } b =' -' \end{cases}$$

- Similarity between S and T given the alignment (S', T')

$$V(S,T) = \sum_{i=1}^{n} \sigma(S'_i, T'_i)$$

Global alignment

- INPUT: Two sequences S and T of roughly the same length
  Q: What's the maximum similarity between the two. Find abest alignment.

Nomenclature

- Σ - an alphabet, a non-empty finite set. For example, $\Sigma = \{A, C, G, T\}$.
- A **string** over Σ is any finite sequence of characters from Σ.
- Σ^n - the set of all strings over Σ of length n. Note that $\Sigma^0 = \{\epsilon\}$.
- The set of all strings over Σ of any length is denoted $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$
- **a substring** of a string $T = t_1 \cdots t_n$ is a string $\hat{T} = t_{1+i} \cdots t_{m+i}$, where $0 \leq i$ and $m + i \leq n$.
- **a prefix** of a string $T = t_1 \cdots t_n$ is a string $\hat{T} = t_1 \cdots t_m$, where $m \leq n$.
- **a suffix** of a string $T = t_1 \cdots t_n$ is a string $\hat{T} = t_{n-m+1} \cdots t_n$, where $m \leq n$.
- **a subsequence** of a string $T = t_1 \cdots t_n$ is a string $\hat{T} = t_{i_1} \cdots t_{i_m}$ such that $i_1 < \cdots < i_m$, where $m \leq n$.

Nomenclature

Biology	Computer Science
Sequence	String,word
Subsequence	Substring (contiguous)
N/A	Subsequence
N/A	Exact matching
Alignment	Inexact matching

Pairwise global alignment

- Example: one possible alignment between ACGCTTTG and CATGTAT is

  ```
  S: AC--GCTTTG
  T: -CATG-TAT-
  ```

- Global alignment
  Input: Two sequences $S = s_1 \cdots s_n$ and $T = t_1 \cdots t_m$ (n and m are approximately the same).
  Question: Find an optimal alignment $S \to S'$ and $T \to T'$ such that $V = \sum_{i=1}^{d} \sigma(S'_i, T'_i)$ is maximal.

Dynamic programming

Let $V(i,j)$ be the optimal alignment score of $S_{1\ldots i}$ and $T_{1\ldots j}$ ($0 \leq i \leq n$, $0 \leq j \leq m$). V has the following properties:
Base conditions:

$$V(i,0) = \sum_{k=0}^{i} \sigma(S_k, '-') \tag{1}$$

$$V(0,j) = \sum_{k=0}^{j} \sigma('-', T_k) \tag{2}$$

(3)

Recurrence relationship:

$$V(i,j) = \max \begin{cases} V(i-1, j-1) + \sigma(S_i, T_j) \\ V(i-1, j) + \sigma(S_i, '-') \\ V(i, j-1) + \sigma('-', T_j) \end{cases} \tag{4}$$

Tabular computation of optimal alignmen

- pseudo code:

```
for i=0 to n do
begin
  for j=0 to m do
  begin
     Calculate V(i,j) using
     V(i-1,j-1), V(i,j-1) and V(i-1,j)
  end
end
```

Tabular computation

i	j	0	1 C	2 A	3 T	4 G	5 T
0		0	-1	-2	-3	-4	-5
1	A	-1	-1	1	0	-1	-2
2	C	-2	1	0	0	-1	-2
3	G	-3	0	0	-1	2	1
4	C	-4	-1	-1	-1	1	1
5	T	-5	-2	-2	1	0	3
6	G	-6	-3	-3	0	3	2

Score: match=+2, mismatch=-1.

Pairwise alignment

- Reconstruction of the alignment: Traceback
  Establish pointers in the cells of the table as the values are computed.
- The time complexity of the algorithm is $O(nm)$. The space complexity of the algorithm is $O(n+m)$ if only $V(S,T)$ is required and $O(nm)$ for the reconstruction of the alignment.

Global alignment in linear space

- Let $V^r(i,j)$ denote the optimal alignment value of the last i characters in sequence S against the last j characters in sequence T.

$$V(n,m) = \max_{k\in[0,m]} \left\{ V(\frac{n}{2},k) + V^r(\frac{n}{2},m-k) \right\} \quad (5)$$

Global alignment in linear space

Hirschberg's algorithm:

1. Compute $V(i,j)$. Save the values of $\frac{n}{2}$-th row. Denote $V(i,j)$ the forward matrix F
2. Compute $V^r(i,j)$. Save the values of $\frac{n}{2}$-th row. Denote $V^r(i,j)$ the forward matrix B
3. Find the column k^* such that

$$F(\frac{n}{2}k^*) + B(\frac{n}{2}, m-k^*)$$

   is maximal
4. Now that k^* is found, recursively partition the problem into two sub problems: i) Find the path from $(0,0)$ to $(n/2,k^*)$
   ii) Find the path from $(n/2, m-k^*)$ to (n,m).

Hirschberg's algorithm

The time complexity of Hirschberg's algorithm is $O(nm)$. The space complexity of Hirschberg's algorithm is $O(\min(m, n))$.

Local alignment problem

- Input: Given two sequences S and T.
  Question: Find the subsequece α of S and β of T, whose simililarity (optimal global alignment) is maximal (over all such pairs of subsequences).
- Example: S=GGTCTGAG and T=AAACGA
  Score: match = 2; indel/substitution=-1
  The optimal local alignment is $\alpha =$CTGA and $\beta =$CGA:
  CTGA ($\alpha \in S$)
  C-GA ($\beta \in T$)

Local Suffix Alignment Problem

- Input: Given two sequences S and T and two indices i and j.
  Question: Find a (possibly empty) suffix α of $S_{1...i}$ and a (possibliy empty) suffix β of $T_{1...j}$ such that the value of the alignment between α and β is maximal over all alignments of suffixes of $S_{1...i}$ and $T_{1...j}$.
- Terminology and Restriction
  $V(i,j)$: denote the value of the optimal local suffix alignment for a given pair i, j of indices.
  Limit the pair-wise scores by:

$$\sigma(x,y) = \begin{cases} \geq 0 & \text{if } x,y \text{ match} \\ \leq 0 & \text{if } x,y \text{ do not match, or one of them is a space} \end{cases} \quad (6)$$

Local Suffix Alignment Problem

- Recursive Definitions

  Base conditions:

  $V(i,0) = 0, V(0,j) = 0$ for all i and j.

  Recurrence relation:

$$V(i,j) = \max \begin{cases} 0 \\ V(i-1,j-1) + \sigma(S_i, T_j) \\ V(i-1,j) + \sigma(S_i, '-') \\ V(i,j-1) + \sigma('-', T_j) \end{cases} \quad (7)$$

  Compute i^* and j^*:

$$V(i^*, j^*) = \max_{i \in [1,n], j \in [1,m]} V(i,j)$$

Local Suffix Alignment Problem

i	j	0	1 x	2 x	3 x	4 c	5 d	6 e
0		0	0	0	0	0	0	0
1	a	0	0	0	0	0	0	0
2	b	0	0	0	0	0	0	0
3	c	0	0	0	2	1	0	0
4	x	0	2	2	2	1	1	0
5	d	0	1	1	1	1	3	2
6	e	0	0	0	0	0	2	5
7	x	0	2	2	2	1	1	4

Score: match=+2, mismatch=-1.

Gap Penalty

- Definition: A *gap* is any maximal, consecutive run of spaces in a single sequece of a given alignment.
  Definition: The *length* of a gap is the number of indel operations in it.
  Example:

  ```
  S: attc--ga-tggacc
  T: a--cgtgatt---cc
  ```

  7 matches, $N_{gaps} = 4$ gaps, $N_{spaces} = 8$ spaces, 0 mismatch.

Affine Gap Penalty Model

- A total penalty for a gap of length q is:

$$W_{total} = W_g + qW_s$$

where
W_g: the weight for "openning the gap"
W_s: the weight for "extending the gap" with one more space
Under this model, the score for a particular alignment $S \rightarrow S'$ and $T \rightarrow T'$ is:

$$\sum_{i \in \{k : S'_i \neq' - ' \ \& \ T'_k \neq' - '\}} \sigma(S'_i, T'_i) + W_g N_{gaps} + W_s N_{spaces}$$

Global alignment with affine gap penality

To align sequence S and T, consider the prefixes $S_{1...i}$ of S and $T_{1...j}$ of T. Any alignment of these two prefixes is one of the following three types:

- Type 1 ($A(i,j)$): Characters S_i and T_j are aligned opposite each other.

```
S: ***********i
T: ***********j
```

- Type 2 ($L(i,j)$): Character S_i is aligned to a chracter to the *left* of T_j.

```
S: ***********i------
T: ******************j
```

- Type 3 ($R(i,j)$): Character S_i is aligned to a chracter to the *right* of T_j.

```
S: ******************i
T: ***********j-----
```

Global alignment with affine gap penality

- $A(i,j)$ – the maximum value of any alignment of Type 1
- $L(i,j)$ – the maximum value of any alignment of Type 2
- $R(i,j)$ – the maximum value of any alignment of Type 3
- $V(i,j)$ – the maximum value of any alignment

Recursive Definition

- Recursive Definition

  Base conditions:

$$V(0,0) = 0 \tag{8}$$

$$V(i,0) = R(i,0) = W_g + iW_s \tag{9}$$

$$V(0,j) = L(0,j) = W_g + jW_s \tag{10}$$

Recurrence relation:

$$V(i,j) = \max\{A(i,j), L(i,j), R(i,j)\} \tag{11}$$

$$A(i,j) = V(i-1,j-1) + \sigma(S_i, T_j) \tag{12}$$

$$L(i,j) = \max\{L(i,j-1) + W_s, V(i,j-1) + W_g + W_s\} \tag{13}$$

$$R(i,j) = \max\{R(i-1,j) + W_s, V(i-1,j) + W_g + W_s\} \tag{14}$$

Local alignment problem

- Local alignment problem
  Input: Given two sequences S and T.
  Question: Find the subsequece α of S and β of T, whose similarity (optimal global alignment) is maximal (over all such pairs of subsequences).
- Example: S=GGTCTGAG and T=AAACGA
  Score: match = 2; indel/substitution=-1
  The optimal local alignment is $\alpha =$CTGA and $\beta =$CGA:
  CTGA ($\alpha \in S$)
  C-GA ($\beta \in T$)
- Suppose the maximal local alignment score between S and T is S.
  How to measure the significane of S?

Measure statistical significance

- One possible solution:

  1. Generate many random sequences $T_1, T_2, \cdots, T_N$, (e.g. $N > 10,000$).
  2. Find the optimal alignment score S_i between S and T_i for all i.
  3. *p*-value $= \sum_{i=1}^{N} I(S_i \geq S)/N$.

  However, the solution is not practical.

Extreme value distribution (EVD)

- Suppose that $X_1, X_2, \cdots, X_n$ are iid random variables. Denote the maximum of these r.v. by $X_{\max} = \max\{X_1, X_2, \cdots, X_n\}$
- Suppose that $X_1, \cdots X_n$ are continuous r.v. with density function $f_X(x)$ and cumulative distribution function $F_X(x)$.
  Question: what is the distribution of $X_{\max}$?

Extreme value distribution (EVD)

- Note that $\mathrm{Prob}(X_{\max} \leq x) = [\mathrm{Prob}(X \leq x)]^n$. Hence

$$F_{X_{\max}}(x) = (F_X(x))^n$$

- Density function of $X_{\max}$

$$f_{X_{\max}}(x) = n f_X(x)(F_X(n))^{n-1}$$

Example: the exponential distribution

- the exponential distribution

$$f_X(x) = \lambda e^{-\lambda x}, \quad x \geq 0 \tag{15}$$

$$F_X(x) = 1 - e^{-\lambda x}, \quad x \geq 0 \tag{16}$$

Mean: $1/\lambda$; Variance: $1/\lambda^2$.

EVD of the exponential distribution

- The EVD:

$$f_X(x) = n\lambda e^{-\lambda x}(1 - e^{-\lambda x})^{n-1} \tag{17}$$

$$F_{X_{\max}}(x) = (1 - e^{-\lambda x})^n \tag{18}$$

EVD of the exponential distribution

- Mean and variance of $X_{\max}$:

$$\mathrm{E}[X_{\max}] = \frac{1}{\lambda}(1 + \frac{1}{2} + \cdots + \frac{1}{n}) \stackrel{n\to\infty}{\longrightarrow} \frac{1}{\lambda}(\gamma + \log n) \tag{19}$$

$$\mathrm{Var}[X_{\max}] = \frac{1}{\lambda^2}(1 + \frac{1}{2^2} + \cdots + \frac{1}{n^2}) \stackrel{n\to\infty}{\longrightarrow} \frac{\pi^2}{6\lambda^2} \tag{20}$$

where $\gamma = 0.5772\ldots$ is Euler's constant.

Asymptotic distribution

- Asymptotic formula for the distribution of $X_{\max}$.
  Define a rescaled $X_{\max}$:

$$U = \frac{X_{\max} - \log(n)/\lambda}{1/\lambda} = \lambda X_{\max} - \log n$$

  As $n \to \infty$, the mean of U approaches γ and the variance of U approaches $\pi^2/6$.

Gumbel distribution

- The cumulative distribution:

$$\begin{aligned} \text{Prob}(U \leq u) =& \text{Prob})(X_{\max} \leq (u + \log n)/\lambda) && (21) \\ =& (1 - e^{-u}/n)^n && (22) \\ =& e^{-e^{-u}} \quad \text{as } n \to \infty && (23) \end{aligned}$$

Or equivalently

$$\text{Prob}(U \geq u) = 1 - e^{-e^{-u}} \quad \text{as } n \to \infty$$

which is called Gumbel distribution.

EVD of the exponential distribution

- EVD for large u The density function

$$f_U(u) = e^{-u}e^{-e^{-u}} \approx e^{-u}(1 - e^{-u} + \frac{e^{-2u}}{2!} - \dots) \approx e^{-u}$$

which decays much slower than the Gaussian distribution.

Karlin & Altschul statistics

- Karlin & Altschul statistics

  For local ungapped alignments between two sequences of length m and n, the probability that there is a match of a score greater than S is:

$$P(x \geq S) = 1 - e^{-Kmne^{-\lambda S}}$$

  Denote $E(S) = Kmne^{-\lambda S}$ - the expected number of unrelated matches with score greather than S.

  Significane requirement: $E(S)$ should be significantly less than 1, that is

$$S < \frac{\log(mn)}{\lambda} + \frac{\log K}{\lambda}$$