Lecture 2 Sequencing

Date / / No. 1

A sequence of DNA is denoted by $x_1 x_2 \cdots x_k$ where $x_i \in \{A, T, G, C\}$. If $S = x_1 x_2 \cdots x_k$, we denote $x_i = S(i)$.

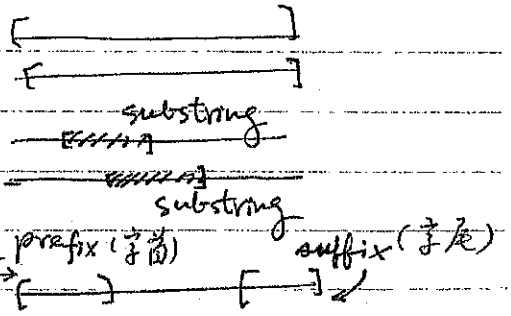
The first problem we study about DNA sequences is the caparison of two sequences and determine how "similar" they are.

In general, the sequences we compare may not of the same length, i.e. two sequences may have different number of terms, but not too "different" from each other. So, we may use the so-called alignment to adjust their length (the number of terms).

e.g. A G A C C T A G

G C A C C T G C A G

Types
① Global alignment
② Local alignment
③ Semi-global alignment



(只在字首或字尾作調整長度的動作。)

Date / /

No. 1

(*) Definition (Alignment)

The alignment is "the comparison of two or more nucleotide or protein sequences (strings) to determine the degree of similarity." → $x_i \in$ {a set of twenty letters}.

(•) Commonly used to deduct functional or evolutionary relationships between genes and proteins.

(*) Global alignments attempt to align every base (or amino acid) in each aligned sequence (string).

(*) Local alignments will align only similar regions between sequences (strings), and leave regions with too many differences unaligned. (相差太多的部分不用調整的意思!)

(*) Multiple Sequence alignment refers to the process of aligning three or more nucleotide or protein sequences to identify similarities between the sesequences. (among)

(•) Many algorithms are known so far!

More details

Definition (Alignment)

An alignment of two sequences S_1 and S_2 is a pair of sequences (S_1', S_2') obtained by insertion of spaces in S_1 and S_2 (respectively) such that

(1) $|S_1'| = |S_2'|$, and

(2) $\forall i$, $S_1'(i)$ is aligned with $S_2'(i)$ and either $S_1'(i)$ or $S_2'(i)$ is not a space. (不可以同時為 space!)

(Note) S_1 (S_2) can be viewed as a subsequence of S_1' (S_2') if we consider the insertions as elements.

Example $S_1 = AGAC \leftarrow \langle A, G, A, C\rangle$

$S_2 = TACCC \leftarrow \langle T, A, C, C, C\rangle$

$S_1' = *AGAC$ | or | $S_1' = *AGAC*$

$S_2' = TACCC$ | | $S_2' = TACC*C$

Score of Alignment

$\frac{x_1 x_2 \cdots x_k}{y_1 y_2 \cdots y_k}$ (After alignment)

For i, j, if
- $x_i = y_i \longrightarrow p > 0$
- $x_i \neq y_j \longrightarrow q < 0$
- $x_i \sim * \longrightarrow r < 0$
- $* \sim y_i \longrightarrow r < 0$

負值

two sequences

Similarity of S_1 and S_2

$$Sim(S_1, S_2) = \max_{\substack{(S_1', S_2') \\ \text{all alignments}}} \left\{ \alpha(S_1', S_2') = \sum_{i=1}^{k} \alpha(S_1'(i), S_2'(i)) \right\}, \quad |S_1'| = |S_2'|$$

$$\alpha(x_i, y_i) = \begin{cases} p > 0 & \text{if } x_i = y_i, \\ q < 0 & \text{if } x_i \neq y_i \text{ and} \\ r < 0 & \text{if one of } x_i, y_i \text{ is a space } *. \end{cases}$$

Definition (Optimal Alignment)

The pair (S_1', S_2') with largest score is called an optimal alignment.

For example: (For AGC and AAAC) Let $p=1$, $q=-1$ and $r=-2$.

*AGC	A*GC	AG*C
AAAC	AAAC	AAAC
(-1)	(-1)	(-1)

are optimal alignments with $Sim(AGC, AAAC) = -1$.

(*) 在 "Alignment", 原來序列的前後順序不咋改變!

Algorithm

目標: Find $Sim(S_1, S_2)$ for input two sequences S_1 and S_2.

1. Consider all possible alignments (S_1', S_2').
2. Find $\alpha(S_1', S_2')$.

e.g. One insertion of AGC $\longrightarrow$ S_1: $S_{1,1}'$ *AGC, $S_{1,2}'$ A*GC, $S_{1,3}'$ AG*C, $S_{1,4}'$ AGC*

$S_2 = S_2' \Rightarrow \alpha(S_{1,1}', S_2') = -1$, $\alpha(S_{1,2}', S_2') = -1$

$\alpha(S_{1,3}', S_2') = -1$, $\alpha(S_{1,4}', S_2') = -3$.

Theorem The time to accomplish "finding optimal alignment" is $O(|S_1||S_2|)$ and the space we need is also $O(|S_1||S_2|)$.

Proof. Since constant work is required per entry in the matrix, the proof follows. ∎

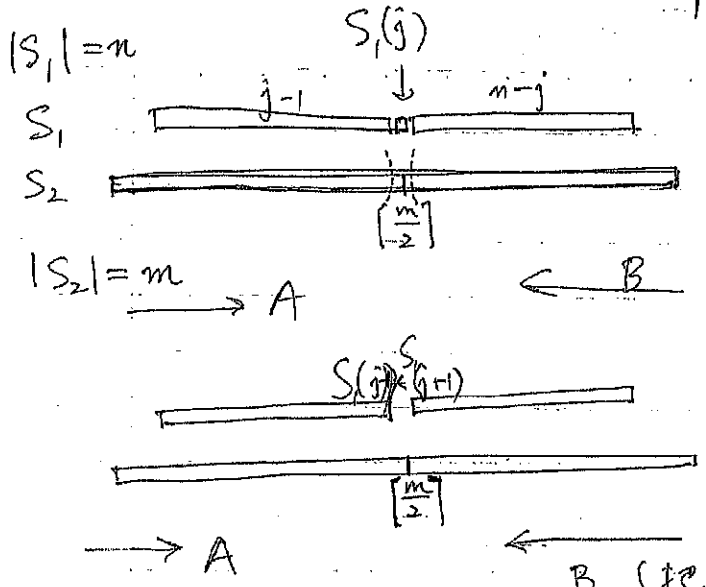
Note No algorithm is known that uses asymptotically less time and has the same generality. Although, there are algorithms for more specific problems.

Note 儲存兩字ce對的 sequences 並不難，但是，如果每字 sequence 都有 10,000 字母，則要出矩陣就要填入 $(10,000)^2 = 100,000,000$ 個格點數，是龐大的空間。

(*) We are aiming to find the "optimal alignments" instead of "presenting the matrix". (不斷地調整長度 5 ce 較當然很 花時間！)

Bonnie Berger's Idea

(*) $A(i,j)$: Score of optimal alignment of $S_1[1,i]$ and $S_2[1,j]$.



$Sim(S_1, S_2) = A(|S_1|, |S_2|)$.

固定 $\lceil \frac{m}{2} \rceil$，變動 "j" (in S_1).

B (把兩 Sequences 都倒過來看).

Step 1 選擇中間的字母（從 S_2 中）。

（這個字母所對的位置不外乎是某一 $S_1(j-1)$ 與 $S_1(j)$ 或 $S_1(j)$ 與 $S_1(j+1)$ 中間的空格。）(For some j)

Step 2 計算 ($\forall j$)

$$\max \begin{cases} A(\lceil \frac{m}{2} \rceil - 1, j-1) \pm 1) + B(\lfloor \frac{m}{2} \rfloor, n-j) \text{ ; and} \\ (A(\lceil \frac{m}{2} \rceil - 1, j) - 2) + B(\lfloor \frac{m}{2} \rfloor, n-j). \end{cases}$$

Step 3 The maximum tell you the best score for aligning the $\frac{m}{2}$(th) character of S_2 with some character of S_1 or gap. (2n values!)

Time we need

$T(m,n) = 2cmn$.

$T(m,1) = m$

$T(1,n) = n$

$$T(m,n) \le cmn + \max_{j=1}^{n} \left(2c\left(\left\lceil \tfrac{m}{2} \right\rceil - 1\right) \cdot j + 2c\left\lfloor \tfrac{m}{2} \right\rfloor (n-j) \right)$$

By induction

$$\le cmn + \max_{j=1}^{n} \left(cmj + cm(n-j) \right)$$

$$\le cmn + cmn$$

$$\le 2cmn$$

⇑ Global Alignments

Semiglobal Comparison :

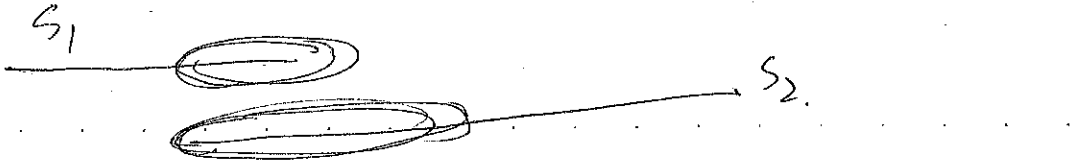
To find the optimal alignment between suffix (prefix) of S_1 with prefix (suffix) of S_2.

Example

S_1: CAGCA*CTTGGATTCTCGG

S_2: CAGCGTGG

prefix ? suffix ? sub-string



Multiple Sequences Alignment (MSA)

Definition (Sum-of-pairs, SP-score)

Let α be a collection of alignments of $S_1, S_2, \cdots, S_n$: $S_1', S_2', \cdots, S_n'$. Then, the SP-score of α, denoted by $SP(\alpha) = \sum_{1 \le i < j \le n} \sigma(S_i', S_j')$, $\sigma(S_i', S_j')$ is called the score of $\alpha_{i,j}$, denoted by $s(\alpha_{i,j})$

(*) Since MSA's are applied in finding the similarity of protein sequences, we use characters of proteins for examples.

S_1 : RCTLEE S_1' : RCTLEE

S_2 : RCLEE $\xRightarrow{\alpha}$ S_2' : RC*LEE

S_3 : CTLEE S_3' : *CTLEE

S_4 : CTEE S_4' : *CT*EE

(With $p = 1$, $q = -1$, $r = -2$) $\quad * \leftrightarrow *$ score 0

$$SP(\alpha) = s(\alpha_{1,2}) + s(\alpha_{1,3}) + s(\alpha_{1,4}) + s(\alpha_{2,3}) + s(\alpha_{2,4}) + s(\alpha_{3,4})$$

$$= 3 + 3 + 0 + 0 + (-3) + 2 = 5.$$

蛋白质的代表符号（20种）

		简写
1.	A : Alanine	Ala
2.	C : Cysteine	Cys
3.	D : Aspartic Acid	Asp
4.	E : Glutamic Acid	Glu
5.	F : phenylalanine	phe
6.	G : Glycine	Gly
7.	H : Histidine	His
8.	I : Isoleucine	Ile
9.	K : Lysine	Lys
10.	L : Leucine	Leu
11.	M : Methionine	Met
12.	N : Asparagine	Asn
13.	P : Proline	Pro
14.	Q : Glutamine	Gln
15.	R : Arginine	Arg
16	S : Sreine	Ser
17.	T : Threonine	Thr
18.	V : Valine	Val
19.	W : Trptaphan	Trp
20.	Y : Tyrasine	Tyr

MSA Problem

Given k sequences, find the optimal alignment of these k sequences.

Dynamic Programming

Time-complexity $= O(k^2 2^k n^k)$

(Exponential in k!)

觀察：每一次調整都要和其它的 Sequences 比對一次！

研究進度

(1.) MSA problem is NP-complete!

Wang and Jiang (1994) & Bonizzoni and Vedova (2001)

2. Gusfield (1993) : $(2-\frac{2}{k})$-approximation algorithm

(If $k=2$, then the algorithm is working.)

3. Pevzner (1992) : $(2-\frac{3}{k})$-approx. algorithm. (Works for $k=3$!)

(*) 4. Bafna, Lawler and Pevzner (1997) : Approximation algorithms for MSA, Theoretical Computer Science, 182, 233-244.

$(2-\frac{l}{k})$-approx. algorithm for $l<k$.

 10

For references (参考资料)

Home

Sequence Alignment Software

Sequence Alignment Web Resources

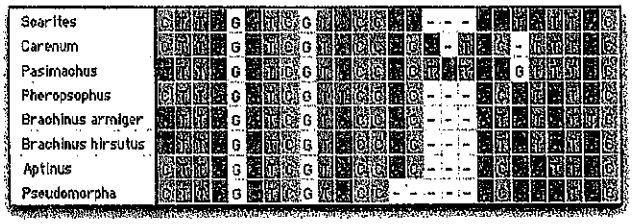
Glossary

About

Contact

DNA Sequence Alignment

Sequence alignment describes the way of aligning DNA, RNA, or protein sequences to highlight or identify similarities between DNA sequences. Typically, gaps have to be inserted into sequences so that identical or similar nucleotides or amino acids are aligned in columns. Here is an example:



Sequence alignment is often used in several biomedical research fields, including phylogenetics, biogeography, and evolution research.

Here is an example of a phylogenetics experiment that includes DNA sequence alignments. Our researcher works for a museum of natural history, and wants to study the evolutionary relation between several animal species he studies. A typical approach would be:

- Isolate DNA from each of the species studied
- Generate DNA sequences for a gene region, an entire gene, or several genes
- Assemble the DNA sequences for each species into separate "contigs"
- Edit the sequence assembly results to remove any errors
- Align the contig sequences to each other using a multiple sequence alignment program
- Verify any observed differences by going back to the original DNA sequences
- Correct the placement of gaps in the aligned sequences, if necessary
- Export the sequence alignment for further analysis with phylogenetics software, for example to generate phylogenetic trees

This web site provides links to commonly used programs and web resources for DNA sequence alignments. Since hundreds of different programs and relevant web sites exist, the goal is *not* to provide lists, but rather to concentrate on the most commonly used and the most useful sequence alignment software.

Sequence Alignment Terms

Alignment

The comparison of two or more nucleotide or protein sequences to determine the degree of similarity. Commonly used to deduct functional or evolutionary relationships between genes and proteins.

Assembly

The process of combining short DNA sequence fragments into larger units by looking for overlaps between different fragments. Often required because the length of the genes studied exceeds the length of the sequence fragments produced by DNA sequencing machines. Also used to combine several fragments that cover the same region, for example in forward and reverse direction, with the goal to reduce errors in the consensus sequence.

Consensus Sequence

A single sequence generated from an alignment or assembly of sequence fragments that is the "best fit" for the given sequences. Historically, majority ("vote based") and inclusive methods were most commonly used to determine consensus sequence. For sequence assemblies, these methods have often been replaced by quality-based consensus methods. Quality-based consensus sequences are typically more accurate that majority-based sequences, and can reduce the need for manual editing of sequence assemblies drastically.

Contig

The result of a sequence assembly or alignment that shows the arrangement of the fragments to form a contiguous large sequence.

Dynamic Programming

A computer-science based method to find the optimal alignment between sequences. For two sequences, this algorithm creates a two-dimensional matrix based on identityl or similarity of bases (or amino acids) in both sequences, and then finds the highest-scoring path to obtain the alignment. A commonly used dynamic programming method is the Needleman-Wunsch algorithm. A nice graphical display of the dynamic programming methods for sequence alignments can be found here.

Global Alignments

Global alignments attempt to align every base (or amino acid) in each aligned sequence.

Local Alignments

Local alignments will align only similar regions between sequences, and leave regions with too many differences unaligned. Local alignments can be better suited for the alignment of very dissimilar sequences. In sequence assembly, the program Phrap demonstrate that local alignments can be used to reduce or eliminate the need to remove low-quality sequence (end clipping) before assembly.

Multiple Sequence Alignment

Multiple sequence alignment refers to the process of aligning three or more nucleotide or protein sequences to identify similarities between the sequences. Alignments that include many sequences can be computational intensive, and require more sophistated algorithms than pairwise alignments.

Pairwise Alignment

In pairwise sequence alignment, exactly two nucleotide or protein sequences are aligned to each other to determine the similarity between the two sequences.

Word-based Alignment Methods

Word-based alignment methods are an optimization often used in sequence alignment and assemblies. Instead of examining every single nucleotide or amino acid, "words" of a fixed length are analyzed. This can lead to substantial reductions in memory use and alignment times. One common application is to used the number of shared words between two sequences to estimate the similarity in early phases of sequence alignments, or to identify sequences that share overlaps in sequence assembly.