Chương 2: Sắp hàng hai chuỗi

Phần mềm

Bioedit

Megax

Minor variant finder

Tại sao cần so sánh trình tự

Ứng dụng so sánh trình tự :

* Giải trình tự bộ gen
* Tìm các chuỗi con chung trong hai bộ gen
* Xác định đột biến gen

String edit :

2 chuỗi A, B , chỉnh sửa chuỗi A thành B

A: source , cột

B: target , hàng

3 thao tác chỉnh sửa :

* Thay thế 1 kí tự trong chuỗi A bằng 1 kí tự trong chuỗi B
* Chèn thêm 1 chữ cái
* Xoá 1 chữ cái nếu trong chuỗi gốc có kí tự đó nhưng trong target không có chuỗi đó

Khoảng cách = số lượng thao tác tối thiểu cần thực hiện để chuyển từ chuỗi nguồn A sang chuỗi đích B

Ví dụ:

A = interestingly

B = bioinformatics

\_i\_\_nterestingly

bioinformatics\_\_

1011011011001111

Vấn đề liên kết chuỗi:

Sử dụng hàm tương tự thay vì hàm cost

Bảng hàm tương tự

Table

Description automatically generated

**An alignment of two sequency:**

Tại 1 vị trí không được phép thêm khoảng trống ở cả hai chuỗi

Cùng độ dài

String alignment

VD:

A\_CAATCC

AGCA\_TGC

5 cặp match, 1 cặp mismatch(C G), 1 chèn (\_ G) (chèn : nguồn là \_ -> đích), 1 xoá (A \_) (xoá : nguồn -> đích là \_)-> 2 indel , 1 mis-match , 5 match-> trọng số = 7

Giải thuật tìm ra alignment tối ưu về trọng số:

Maximum score: điểm tối đa

Optimal alignment  
 bài toán căn chỉnh chuỗi

Thuật toán Needleman-Wunsch

Chuỗi s: nguồn , chuỗi t: đích .

Kẻ bảng gồm các hàng ứng với chuỗi đích , cột ứng với chuỗi nguồn:

* Global alignment

Giới hạn số thao tác chèn xoá

* Local alignment

Chuỗi S[1…n]

Chuỗi T[1..m]

* Tìm xem đoạn con nào của chuỗi S và chuỗi T có liên kết toàn cục với điểm cao nhất

Thuật toán Brute force

Giải thuật tốt hơn: Giải thuật Smith-waterman -> độ phức tạp nm

* Tiền tố , hâu tố của chuỗi
* Semi-global alignment

Khởi tạo cột thứ nhất = 0

Khởi tạo hàng thứ nhất giống như bình thường

Kết quả : Chọn giá trị lớn nhất trên cột cuối cùng , tìm được đường đi tối ưu -> alignment