PHÁT TRIỂN THUẬT TOÁN TẠO CÂY PHÂN LOÀI

Đỗ Phúc - Lê Anh Tài
Trường Đại học Khoa học Tự nhiên
(Bài nhận ngày 06/04/2000,
hoàn chỉnh sửa chữa ngày 01/11/2000)

TÓM TẮT *: Trong bài báo này chúng tôi tập trung vào việc xây dựng phần mềm hỗ trợ tạo cây phân loài dựa trên tập các trình tự sinh học DNA. Chúng tôi dựa vào khoảng cách Hamming và qui hoạch động để tính khoảng cách (phản ánh mức độ tương đồng) giữa hai trình tự để xây dựng thuật toán đối sánh hai và nhiều trình tự. Cuối cùng thuật toán tạo nhóm phân cấp (hierachical clustering algorithm) được sử dụng để tạo cây phân loài. Chúng tôi đã dùng thuật toán đề xuất để tạo cây phân loài từ dữ liệu DNA lấy từ WebSite NCBI và cài đặt trình duyệt thông tin di truyền dạng cây trong CSDL DNA trên Web Site Sinh Tin học Việt nam do nhóm đề tài xây dựng.*

1. GIỚI THIỆU

Phân loài là tiến trình phân chia tập các trình tự DNA thành nhiều tập con nhỏ hơn sao cho các trình tự trong một tập con phải có một mức độ giống nhau theo một tiêu chuẩn đánh giá nào đó [6]. Với mỗi tập con các trình tự, chúng ta rút ra một trình tự đặc trưng để phục vụ cho việc gom nhóm các trình tự ở các cấp cao hơn. Kết quả của tiến trình tạo cây phân loài là một cây phân cấp các nhóm trình tự tương đồng. Bài toán tạo cây phân loài là một trong các bài toán quan trọng của sinh học. Từ cây phân loài các nhà sinh học có thể định hướng công tác chọn, lai tạo giống và khảo sát sự tương đồng giữa các trình tự DNA. Mục tiêu của đề tài là xây dựng phần mềm hỗ trợ tiến trình tạo cây phân loài từ tập các trình tự DNA. Chúng tôi sử dụng khoảng cách Hamming và thuật toán qui hoạch động để tính khoảng cách giữa hai trình tự DNA[1,4] và xây dựng thuật toán đối sánh hai trình tự DNA, thuật toán tạo kiến trúc phân cấp nhóm (hierachical clustering algorithm) để tạo cây phân loài [3]. Chúng tôi đã tiến hành thử nghiệm các thuật toán đề xuất trên tập dữ liệu DNA được download từ website NCBI *(http://www.ncbi.nlm.nih.gov)* để tạo cây phân loài cho trình duyệt hỗ trợ tìm kiếm thông tin trên WebSite Công nghệ Sinh Tin học Việt nam *(http://services.citd.edu.vn/bioinfo)* do nhóm đề tài xây dựng. Bài báo được tổ chức như sau: 1) Giới thiệu 2) Thuật toán đối sánh hai trình tự DNA 3) Thuật toán đối sánh nhiều trình tự DNA và rút trích trình tự đặc trưng 4) Thuật toán tạo cây phân loài. 5) Kết quả thử nghiệm 6) Kết luận và hướng phát triển.

2. THUẬT TOÁN ĐỐI SÁNH HAI TRÌNH TỰ DNA

Cho tập B= {A, C, G, T} các nucleotide, trình tự DNA là một chuỗi liên tục s có dạng $s = a_0,a_1,...,a_i,...,a_n$ với $a_i \in B$, i=1,...,n.

Ta ký hiệu 0:s:k là một chuỗi con của s bắt đầu từ vị trí 0 đến vị trí k của s và |s| là chiều dài của trình tự s hay số lượng ký tự trong trình tự s.

2.1. Khoảng cách giữa các trình tự DNA và các thao tác

Khoảng cách giữa hai trình tự DNA là giá trị phản ánh mức độ giống nhau giữa hai trình tự. Khoảng cách Hamming được sử dụng để đo khoảng cách giữa hai trình tự [1]. Khoảng cách Hamming cho biết số các vị trí có trị khác nhau trong hai trình tự DNA, khoảng cách càng nhỏ thì hai trình tự càng giống nhau và ngược lại

Ví dụ1:	Trình tự s =	AAT	AGCAA	AGCACACA
	Trình tự t =	TAA	ACATA	ACACACTA
Khoảng cách Hamming(s,t)		2	3	6

Khoảng cách Hamming rất hữu dụng trong trường hợp hai trình tự có chiều dài bằng nhau nhưng nó không áp dụng được khi hai trình tự có chiều dài khác nhau. Trong cơ chế sao chép DNA, các lỗi như xóa hay chèn thêm một acid nucleotic là phổ biến, do vậy ta có thể xoá hay chèn thêm ký tự trắng (-) vào trình tự để đưa hai trình tự có chiều dài khác nhau thành bằng nhau. Để làm điều này, chúng ta định nghĩa một số thao tác sau:

- (a , a) là một **phép so** (match) (không có sự thay đổi từ s vào t).
- (a , –) là một **phép xóa** kí tự a trong s (deletion).
- (a , b) là một **phép thay** a trong s bởi b trong t (replace).
- (– , b) là một **phép chèn** b vào s (insert).

Do tính đối xứng giữa hai trình tự s và t, nên một phép xóa trong s sẽ tương đương với một phép chèn trong t và ngược lại.

2.2. Đối sánh hai trình tự DNA

Đối sánh hai trình tự s và t là sự sắp xếp vị trí các kí tự giữa s và t bằng cách chèn thêm các kí tự trắng (-) để hai trình tự s, t có cùng chiều dài nhưng lại có khoảng cách nhỏ nhất:

Ví dụ2: s :	AGCACAC–A	hoặc	AG– CACACA
t :	A– CACACTA		ACACACT – A

Các phép toán được sử dụng để s và t có chiều dài bằng nhau:

Trái :	Phải :
Match (A,A)	Match (A,A)
Delete(G, –)	Replace(G,C)
Match (C,C)	Insert (–,A)
Match (A,A)	Match (A,A)
Match (C,C)	Match (A,A)
Match (A,A)	Match (C,C)
Match (C,C)	Replace(A,T)
Insert (–,T)	Delete (C,–)
Match (A,A)	Match (A,A)

Các phép toán ở phía trái bao gồm một phép xóa, một phép chèn , và bảy phép so. Các phép toán ở phía phải bao gồm một phép xóa, một phép chèn, hai phép thay và năm phép so.

2.3.Tính khoảng cách giữa các trình tự DNA

Để tính khoảng cách giữa hai trình tự DNA, chúng ta dùng một hàm chi phí hay trọng số w cho từng phép toán. Cho hai kí tự a, b ∈{A, C, T, G}, hàm tính chi phí w(a,b) được định nghĩa như sau :

$w(a, a) = 0$; $w(a, b) = 1$ với $a \neq b$; $w(a, -) = w(-, b) = 1$

Ví dụ 3: s = AGCACAC-A hoặc AG-CACACA

t= A-CACACTA ACACACT-A

Chi phí là 2 Chi phí là 4

Do vậy sự đối sánh bên trái là tốt hơn sự đối sánh bên phải vì bên trái có chi phí nhỏ hơn [6]. Xét hai chuỗi con dẫn trước (prefix) của s và t được ký hiệu là 0:s :i và 0:t:j với i,j≥1, giả sử chúng ta đã xác lập được các phép toán tối ưu giữa tất cả các chuỗi con dẫn trước có chiều dài ngắn hơn chiều dài của s và t, đặc biệt là các trường hợp của các chuỗi con dẫn trước sau đây:

•0 : s : (i-1) và 0 : t : (j-1)

•0 : s : (i-1) và 0 : t : j

•hoặc 0 : s : i và 0 : t : (j-1)

Một sự đối sánh tối ưu của hai chuỗi con dẫn trước 0 : s : i và 0 : t : j phải là một sự mở rộng của một trong các trường hợp trên bằng cách thực hiện:

•Một phép thay replace(s_i, t_j) hoặc một phép so match(s_i,t_j) tùy thuộc vào s_i có bằng s_j hay không.

•Một phép xóa (s_i, -) hay

•Một phép chèn (-, t_j).

Với s_i và t_j là ký tự tại vị trí i của chuỗi s và ký tự tại vị trí j của chuỗi t.Theo định nghĩa, chúng ta sẽ phải chọn các phép toán có chi phí tối thiểu theo công thức (1):

$$d_w(0 : s : i, 0 : t : j) = \min \{ d_w(0 : s : (i-1), 0 : t : (j-1)) + w(s_i, t_j),$$

$$d_w(0 : s : (i-1), 0 : t : j) + w(s_i, -),$$

$$d_w(0 : s : j, 0 : t : (j-1)) + w(-, t_j) \} \qquad (1)$$

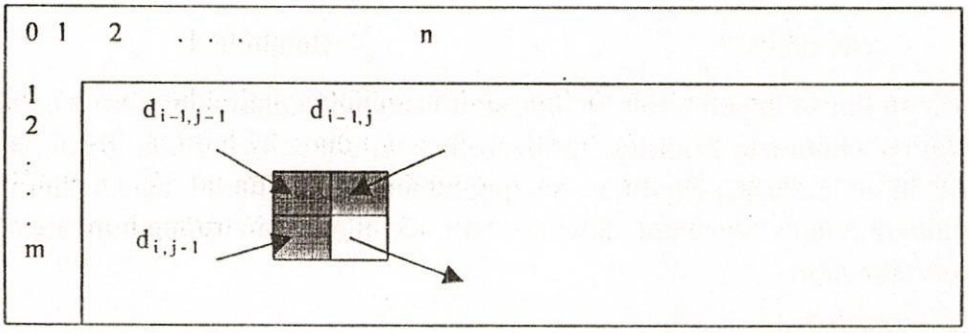
Công thức (1) có dạng đệ qui và là công thức cơ bản để áp dụng qui hoạch động vào việc tính khoảng cách giữa hai trình tự DNA. Từ công thức (1), chúng ta nhận thấy không có sự lựa chọn nào khi một trong các chuỗi con dẫn trước rơi vào một trong ba trường hợp sau đây:

$d_w(0:s:0, 0:t:0) = 0$

$d_w(0:s:i, 0:t:0) = d_w(0:s(i-1), 0:t:0) + w(s_i,-)$ với i=1, ..., m

$d_w(0:s:0, 0:t:j) = d_w(0:s:0, 0:t:(j-1)) + w(-, t_j)$ với j=1, ..., m

Theo phương pháp qui hoạch động [4] với hàm w cho trước, các khoảng cách của tất cả các chuỗi con dẫn trước của s và t sẽ xác định một ma trận khoảng cách D có kích thước là (m+1) × (n+1) với $d_{ij} = d_w(0:s:i, 0:t:j)$. Ba khả năng lựa chọn trong công thức (1) dẫn đến các ràng buộc giữa các phần tử của ma trận khoảng cách như trong hình 1. Trong đó $d_{ij} = \min(d_{i-1,j-1} + w(s_i,t_j), d_{i-1,j} + w(s_i,-), d_{i-1} + w(-,t_j))$. Giá trị nằm ở góc dưới bên phải của ma trận khoảng cách chính là khoảng cách giữa hai trình tự s,t vì $d_{mn} = d_w(0:s:m, 0:t:n) = d_w(s,t)$

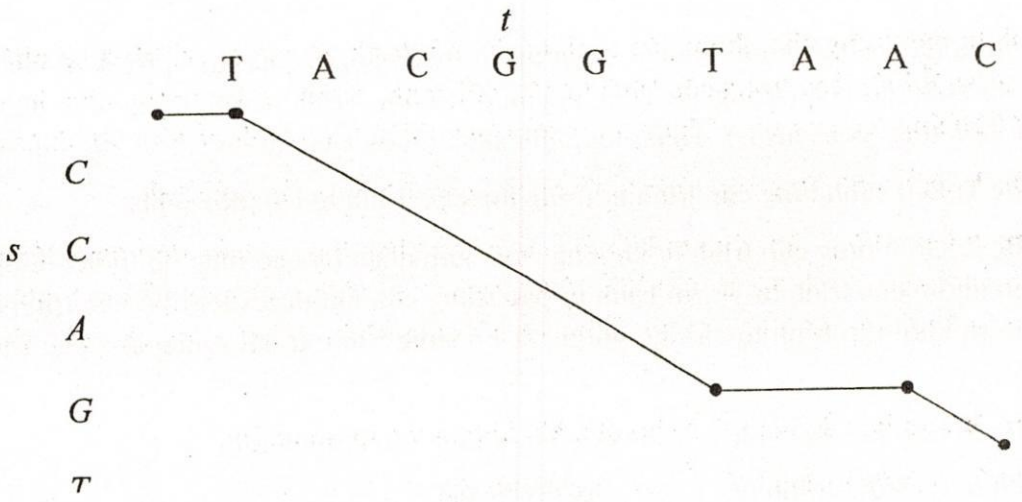


Hình 1: Ràng buộc giữa các phần tử trong ma trận khoảng cách

Ví dụ 4: Với hai trình tự s=CCAGTC và t=TACGGTAAC, ta có ma trận khoảng cách giữa hai trình tự s và t như trong bảng 1 và con đường tối ưu của sự đối sánh của hai trình tự s và t như trong hình 2.

Bảng 1: Ma trận khoảng cách giữa hai trình tự s và t

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



Hình 2: Con đường tối ưu của hai trình tự s và t

2.4. Thuật toán đối sánh hai trình tự DNA

Thuật toán đối sánh hai trình tự có thể tóm tắt như sau:

Vào: Hai trình tự s và t cần tính khoảng cách

Ra: Khoảng cách giữa hai trình tự s và t

Bước 1: Khởi tạo ma trận khoảng cách D

Bước 2: Dùng qui hoạch động để tính giá trị các phần tử của ma trận khoảng cách D.

Bước 3: Trả về giá trị của phần tử nằm ở góc dưới bên phải của ma trận khoảng cách D là khoảng cách giữa hai trình tự s và t.

3. THUẬT TOÁN ĐỐI SÁNH NHIỀU TRÌNH TỰ VÀ RÚT TRÍCH TRÌNH TỰ ĐẶC TRƯNG

Một sự đối sánh của k trình tự là một ma trận hình chữ nhật bao gồm những ký tự được lấy từ tập ký tự mở rộng {A, C, T, G, -} thõa mãn ba điều kiện sau đây:

i) Có đúng k dòng

ii) Bỏ qua những ký tự trắng thì dòng thứ i đúng bằng trình tự s_i

iii) Mỗi cột chứa ít nhất một ký tự khác với ký tự trắng "-"

Ví dụ 5: Sau đây là một sự đối sánh nhiều trình tự

AGTCCTTA–AATTC–A

GTTCCAACT–CTGAA–

ATCCTAGATAT–TATT

ATCGAAGTTCTAA–TT

Trính tự đặc trưng: ATTCCAGATATTTATT

Trình tự đặc trưng cho nhóm các trình tự là một trình tự mà tại vị trí k sẽ nhận ký tự có tần số xuất hiện cao trong các trình tự cần đối sánh. Trình tự đặc trưng còn được gọi là trình tự liên ứng (consensus). Thuật toán đối sánh nhiều trình tự được tóm tắt như sau:

Bước 1: Đối sánh từng cặp trình tự trong tập các trình tự cần đối sánh

Bước 2: Gom từng cặp trình tự khoảng cách gần nhau hay có mức độ tương đồng cao nhất thành nhóm các trình tự và rút trình tự đặc trưng của nhóm rồi loại bỏ các trình tự đã được gom ra khỏi tập trình tự cần go nhóm và bổ sung trình tự đặc trưng cho các trình tự bị loại bỏ.

Bước 3: Lặp lại các bước 1,2 cho đến khi không tạo thêm nhóm.

Ví dụ 6 : Với 4 trình tư ACGTACGT

CGTCGACGTA

CGTACGTAC

ACGACGGTTGGT

Ta kết quả đối sánh nhiều trình tự bằng thuật toán trên như sau:

--ACGTACGT--

-CGTCGACGTA-

---CGTACGTAC

ACGACGGTTGGT

4. THUẬT TOÁN TẠO CÂY PHÂN LOÀI

Cho tập S = { $s_1, s_2, \ldots, s_n$ } gồm n trình tự DNA và một ngưỡng τ. Ta tạo ra ma trận vuông M cấp n với giá trị mỗi phần tử M(i,j) của ma trận là khoảng cách giữa hai trình tự s_i vàs_j được tính theo thuật toán ở mục 2. Sau khi tính ma trận khoảng cách M, ta dựa vào ngưỡng τ để gom các trình tự thành các nhóm. Tùy theo τ, các nhóm này có thể rời nhau hay giao nhau. Thuật toán tạo cây phân lớp các trình tự DNA dựa trên ý tưởng thuật toán tạo cây phân cấp nhóm [3,6] được tóm tắt như sau:

Đầu vào: Tập các trình tự cần tạo cây phân loài

Đầu ra: Cây phân loài (cấu trúc phân cấp các trình tự)

Bước 1: Tính khoảng cách giữa mỗi cặp trình tự DNA

Bước 2: Dựa vào ngưỡng τ để gom các trình tự thành các nhóm và rút đặc trưng cho từng nhóm theo cách trên.

Bước 3: Loại bớt các nhóm đối tượng thừa

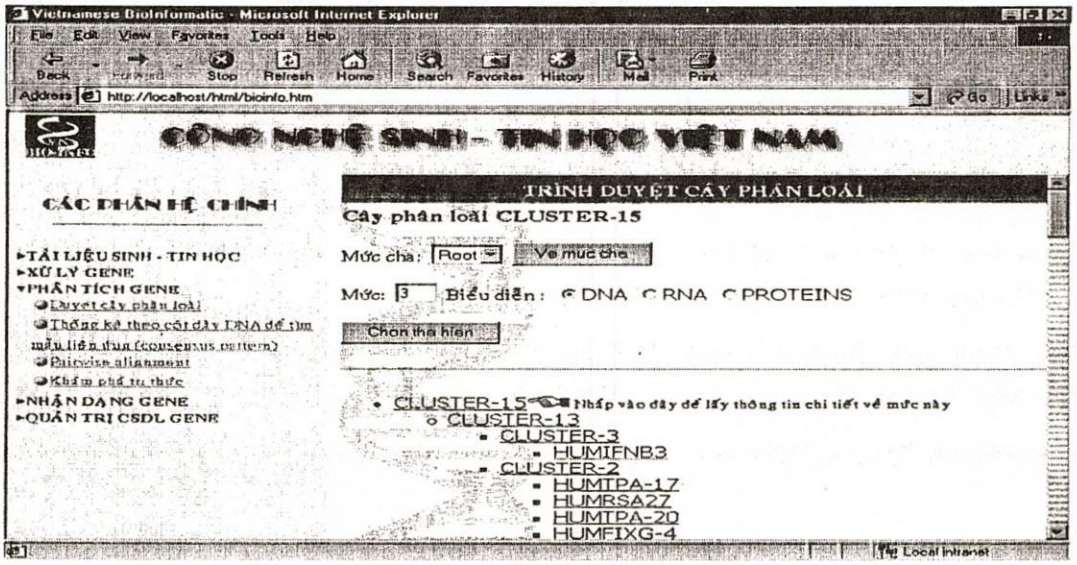
Bước 4: Xem có đưa ra đối tượng mới nào không ? nếu không thì dừng, nếu có thì tiếp tục.

Bước 5: Thiết lập một phân cấp trên các đối tượng mới

Bước 6: Quay lại bước 1 với các đối tượng mới.

5. KẾT QUẢ THỬ NGHIỆM

Chúng tôi đã download dữ liệu từ Website NCBI và tạo cây phân loài trong hình 3. Cây phân loài này được bố trí trong WebSite Công nghệ Sinh Tin học Việt nam(*http://services.citd.edu.vn/bioinfo*) dưới dạng một trình duyệt CSDL.



Hình 3: Trình duyệt CSDL DNA trên Website Sinh tin học Việt nam

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi xây dựng phần mềm tạo cây sinh loài từ tập dữ liệu các trình tự DNA. Dựa trên khoảng cách Hamming và tính chất các trình tự sinh học, chúng tôi phát triển các thuật toán đối sánh hai hay nhiều trình tự sinh học. Thuật toán tạo cây phân loài được phát triển từ thuật toán tạo kiến trúc phân cấp nhóm rất phổ biến trong lĩnh vực khai thác dữ liệu. Kết quả thử nghiệm trên nguồn dữ liệu trình tự DNA lấy từ Internet và có đối sánh với trình duyệt của Website NCBI. Một cây phân loài đã được tạo lập và cài đặt trên Website Sinh Tin học Việt nam do nhóm đề tài xây dựng.

DEVELOPING ALGORITHMS FOR BUILDING PHYLOGENETIC TREES

Đỗ Phúc - Lê Anh Tài

ABSTRACT : In this paper, we focus on developing algorithms for building phylogenetic tree from a set of DNA sequences. We use Hamming distance and dynamic programming to calculate the distance between two DNA sequences (reflecting the sequence similariry). An algorithm for calulcating the distance between two DNA sequences, pairwise aligment and multiple alignment are developed. We use our propsed hierachical clustering algorithm for building the phylogenetic tree and install the discovered phylogenetic tree as a DNA database taxonomy browser in our Bio-Informatic WebSite.

TÀI LIỆU THAM KHẢO

[1]Annette, D.L. M. Griffin and Hugh G. Griffin: Computer Analysis of Sequence Data, Humana Press Inc, Vol.24, (1994)

[2] Boris Mirkin: Mathematical Classification and Clustering: Kluiver Academic Publisher, (1996)

[3] Hoang Kiem, Do Phuc: Developing motif based algorithm for discovering knowledge from a set of DNA sequences, in Proc. Of SCI'2000 conference, Florida, USA, (2000)

[4]Pearson W.R. and Miller: Dynamic Programming Algorithms for Biological Sequence Comparison, Methods in Enzymology, (1992)

[5] R. Durbinans Anders Krogh: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleotide acids, Cambridge University Press, (1998)

[6] Swofford, D.L and Olsen G.J Phylogeny reconstructuion, Molecular Systematics, 1996