PHÁT TRIỂN THUẬT TOÁN TÌM ĐOẠN TƯƠNG TỰ TRONG CÁC TRÌNH TỰ SINH HỌC ADN

Hoàng Kiếm - Đỗ Phúc
Trường Đại học Khoa học Tự nhiên
(Bài nhận ngày 06/04/2000,
hoàn chỉnh sửa chữa ngày 05/09/2000)

TÓM TẮT : *Bài toán tìm các đoạn tương tự trong các trình tự sinh học ADN là một bài toán quan trọng trong công nghệ sinh học. Từ vị trí của các đoạn tương tự, chúng ta có thể suy diễn các qui luật đặc trưng cho nhóm gen hoặc protein có cùng tính trạng hoặc xác định vị trí cho các đột biến điểm trong các gen. Trong bài báo này, chúng tôi phân tích hai bài toán tìm đoạn giống hệt và bài toán tìm đoạn tương tự và sử dụng thuật toán khám phá tập chỉ báo lớn để tìm các đoạn giống nhau và thuật toán di truyền để tìm các đoạn xấp xỉ trong tập các trình tự sinh học ADN. Trong bài toán khám phá chuỗi con lặp, chúng tôi xem chuỗi con lặp như một tập chĩ báo lớn và dùng thuật toán Apriori-TID (Agrawal 94) để tìm các tập chỉ báo lớn tối đại. Trong thuật toán khám phá các đoạn xấp xỉ, chúng tôi xem nhiễm sắc thể như là một lời giải tiềm năng và dùng thuật toán di truyền để chọn đúng lời giải. Cả hai thuật toán đề xuất đều làm việc tốt trên các tập dữ liệu lớn nên rất phù hợp với khối lượng dữ liệu lớn của các dữ liệu gen. Ngoài ra, chúng tôi cũng còn đề xuất một heuristic cho phép đẩy nhanh tiến trình tìm ra lời giải. Chúng tôi đã tiến hành thử nghiệm thuật toán đề xuất trên tập dữ liệu promoter của Đại học Irvine (Hoa kỳ) và trình bày kết quả thử nghiệm.*

1. GIỚI THIỆU

Bài toán tìm các đoạn tương tự trong các trình tự sinh học ADN là một bài toán quan trọng trong công nghệ sinh học. Từ vị trí của các đoạn tương tự trong các trình tự sinh học ADN, chúng ta có thể suy diễn các qui luật đặc trưng cho nhóm gen hoặc protein có cùng tính trạng hoặc xác định vị trí cho các đột biến điểm [1][3][4]. Trong bài báo này, chúng tôi sử dụng hai thuật toán quan trọng là thuật toán khám phá tập chỉ báo lớn để tìm các đoạn giống nhau và thuật toán di truyền để tìm các đoạn xấp xỉ trong tập các trình tự sinh học ADN và ứng dụng vào tập dữ liệu DNA promoter của Đại học Irvine (Hoa kỳ). Bài báo được tổ chức như sau: 1) Giới thiệu 2) Phát biểu bài toán và độ phức tạp 3) Bài toán tìm đoạn lặp giống hệt 4) Bài toán tìm các đoạn lặp xấp xỉ 5) Kết quả thử nghiệm 6) Một heuristic để cải tiến tốc độ tìm kiếm 7) Kết luận và hướng phát triển.

2. PHÁT BIỂU BÀI TOÁN VÀ ĐỘ PHỨC TẠP

Cho một tập $S = \{ s_1, s_2, ..., s_m \}$ gồm m trình tự sinh học ADN, mỗi trình tự sinh học ADN là một chuỗi các acid nucleotid A,C,G,T . Bài toán đặt ra là:

a) Tìm các đoạn giống nhau trong m trình tự sinh học ADN

b) Tìm các đoạn xấp xỉ trong m trình tự sinh học ADN.

Ví dụ 1: Giả sử S = {s_1,s_2,s_3,s_4,s_5} là tập 5 trình tự sinh học ADN như sau:

s_1 = 'A**CGTAAAA**GTCACACGTAGCCCCACGTACAGT'

s_2 = 'CGCGTCGAAGTCGAC**CGTAAAA**GTCACACAGT'

s_3 = 'GGTCGATGCA**CGTAAAA**TCAGTCGCACACAGT'

s_4 =' A**CGTAAAA**GTAGCTACCCGTACGTCACACAGT'

s_5 = 'A**CGTAAAA**GTCAACGACGTACGTCGTCACAGT'

Với bài toán 1, các lời giải khả dĩ là: TCA, GTC, CACA, ACAGT, CGTAAAA

Với bài toán 2, một lời giải tiêu biểu là CGTAAAAG.

Với m là số lượng các trình tự sinh học ADN có chiều dài L và W là chiều dài của đoạn cần tìm, rõ ràng với một trình tự ADN,chúng ta có L-W+1 vị trí có thể có cho đoạn lặp lại, với m trình tự ADN chúng ta có $(L-W+1)^m$ khả năng có thể cho một lời giải của bài toán 1 hoặc bài toán 2. Khi L,W,m lớn thì bài toán sẽ có độ phức tạp rất lớn. Mục tiêu của bài báo là phát triển các thuật toán để giải hai bài toán trên. Ngoài ra, chúng tôi cũng đề xuất một heuristic nhằm tăng tốc độ tìm kiếm lời giải.

3. BÀI TOÁN TÌM ĐOẠN LẶP GIỐNG HỆT

Chúng tôi sử dụng thuật toán tìm tập chỉ báo lớn rất phổ biến trong lĩnh vực khai thác dữ liệu để phát hiện tập chỉ báo lớn tối đại. Một số định nghĩa liên quan đến bài toán và thuật toán đề nghị được liệt kê sau đây:

3.1. Trình tự sinh học ADN

Gọi B = {'A','C','T','G'}, B được gọi là tập các base cấu tạo nên trình tự sinh học ADN. Ta gọi |s| là chiều dài của trình tự sinh học ADN, đây là số acid nucleotid có mặt trong trình tự s.

3.2. Tập các chuỗi con

Cho s là một trình tự sinh học ADN, gọi P(s) là tập các chuỗi con của chuỗi s. Trên P(s) ta định nghĩa quan hệ thứ tự $<$ như sau:$\forall s_a, s_b \in P(s)$, $s_a < s_b$ nếu s_a là chuỗi con của s_b.

Ta gọi s_d là trội trực tiếp của s_a nếu $s_a < s_d$ và không tồn tại một $s_b \in P(s)$ sao cho $s_b \neq s_d$ nhưng $s_a < s_b$ và $s_b < s_d$.

3.3. Đoạn con lặp phổ biến

Cho S là một tập m trình tự sinh học ADN, $P(s_i)$ là tập các chuỗi con của chuỗi s_i và PU là hợp của tất cả các $P(s_i)$, i=1,...,m.

Gọi $N(s_t)$ là số trình tự sinh học ADN chứa s_i. Cho $s_t \in PU$ và một ngưỡng $\tau \in [0,1]$, s_t được gọi là một đoạn lặp phổ biến (motif) trong S theo ngưỡng τ nếu $N(s_t)/m >= \tau$.

Ta định nghĩa $F(S, \tau) = \{ s \in PU \mid N(s)/m >= \tau \}$.

Dễ thấy nếu $s_a \in F(S, \tau)$ và $s_t \in P(s_a) \Rightarrow s_t \in F(s, \tau)$.

3.4. Đoạn con lặp phổ biến tối đại

Cho S là một tập có m trình tự sinh học ADN và ngưỡng τ, $F(S, \tau)$ là tập các chuỗi con phổ biến theo ngưỡng τ. Cho $a \in F(S, \tau)$, a là một chuỗi con phổ biến tối đại của S nếu

$a \in F(S, \tau)$ và $\sim \exists b \in F(S, \tau), a \neq b, a < b$.

3.5. Một thuật toán để tìm chuỗi con phổ biến

Chúng tôi sử dụng ý tưởng của thuật toán Apriotid được R. Agrawal 94 đề xuất [5] để phát triển thuật toán tìm đoạn con lặp phổ biến và đoạn con lặp phổ biến tối đại. Gọi $L(S,k,\tau)$ là tập các chuỗi con phổ biến theo ngưỡng τ có chiều dài bằng k. Do các trình tự sinh học ADN chỉ chứa các ký tự 'A', 'C', 'T', 'G'. Nên các chuỗi con có chiều dài bằng 1 chỉ có thể là các chuỗi "A", "C", "T", "G". Thuật toán như sau:

```
Answer =∅
Generate L(S,1, τ) từ { "A", "C", "T", "G"}
For ( k=2; L(S,k, τ) <> {} ; k++) do
  Tạo L(S,k, τ) từ L(S,k-1, τ) và L(S,1, τ)
Answer = ∪k L(S,k, τ)
Return Answer;
```

a) Tạo L(S,1, τ)

Motifs có chiều dài là 1 chỉ có thể là các chuỗi "A", "C", 'T', 'G' trong chuỗi s, nếu chúng thõa định nghĩa thì đưa chứng vào $L(S,1, \tau)$.

b) Tạo $L(S,k,\tau)$ từ $L(S,k-1, \tau)$ và $L(S,1, \tau)$

Chúng ta sẽ tạo $L(S,k,\tau)$ từ $L(S,k-1,\tau)$ và $L(S,1,\tau)$. Thuật toán như sau:

Tạo ma trận có dòng và cột là các phần tử của $L(S,1, \tau)$..

```
L(S,k, τ) = ∅
For(each s_y ∈ L(S,k-1, τ)) do
    For (each s_x ∈ L(S,1, τ) ) do
      begin
        s_t = s_y + s_x // Toán tử nối chuỗi
        If (N(s_t, τ)/m >= τ ) and |s_t| = k ) then
            SaveFreqMotif(s_t, L(S,k, τ));
      end;
Answer = L(S,k, τ)
  Return Answer
```

Trong thuật toán trên hàm SaveFreqMotif(s_t, L(S,k, τ)) là hàm để lưu chuỗi con s_t vào L(S,k, τ).

4. BÀI TOÁN TÌM ĐOẠN CON LẶP XẤP XỈ

Gọi W là chiều dài của đoạn con lặp xấp xỉ, L là chiều dài của các trình tự sinh học ADN, các yếu tố của thuật toán di truyền được tóm tắt như sau:

4.1. Biểu diễn di truyền

Một nhiễm sắc thể là một ma trận có 4 dòng ứng với 4 acid nucleotid và có W cột. Phần tử M_{bj} là tần suất xuất hiện của ký tự b (A,C,G,T) ở cột j.

4.2. Phép toán lai ghép: Chọn hai nhiễm sắc thể cha để lai ghép, chọn ngẫu nhiên một cột cần lai ghép và hoán chuyển bộ phận cho nhau.

Các nhiễm sắc thể cha ↓

0.6	0.4
0.2	0.4
0.1	0.1
0.1	0.1

0.3	0.7
0.2	0.1
0.3	0.1
0.2	0.1

Các nhiễm sắc thể con

0.6	0.7
0.2	0.1
0.1	0.1
0.1	0.1

0.3	0.4
0.2	0.4
0.3	0.1
0.2	0.1

4.3. Phép toán đột biến

Dựa trên xác suất đột biến để chọn nhiễm sắc thể đột biến, chọn ngẫu nhiễn cột cần đột biến của nhiễm sắc thể và tạo giá trị ngẫu nhiên cho các phần tử trên cột nhưng vẫn đảm bảo tổng gía trị trong cột là 1.0.

Nhiễm sắc thể cha

0.6	0.4
0.2	0.4
0.1	0.2
0.1	0.2

Nhiễm sắc thể bị đột biến

0.6	0.2
0.2	0.2
0.1	0.1
0.1	0.5

Cột bị Đột biến

5.4. Hàm phù hợp

Điểm của một đoạn con S_i (Score((S_i)) trong trình tự sinh học ADN được tính bằng công thức:

$$\text{Score}(S_i) = \prod_{i=1}^{w} M_{bi}$$

Trong đó M_{bi} là phần tử ứng với chữ cái b tại vị trí i của đoạn lặp lại S_i trong nhiễm sắc thể. Với một nguỡng T cho trước, thuật toán sẽ đi tìm đoạn con S_i trong trình tự sinh học ADN X_i. Điểm của nhiễm sắc thể được tính bằng tỉ số giữa số lượng đoạn con vượt ngưỡng với tổng số trình tự ADN khảo sát.

5. KẾT QUẢ THỬ NGHIỆM

5.1. Thử nghiệm tìm đoạn con lặp giống hệt

Thử nghiệm trên tập dữ liệu chứa 106 đoạn promoter do Đại học Irvine(Hoa kỳ) cung cấp. Một số trình từ thuộc lớp promoter(class+) và không phải promoter (class -) được liệt kê trong bảng 1.

Bảng 1: Các đoạn promoter trong các GENE của Đại học Irvine(Hoa kỳ)

#	DỮ LIỆU	CLASS
1	TACTAGCAATACGCTTGCGTTCGGTGGTTAAGTATGTATAATGCGCGGGCTTGTCGT	+
2	TGCTATCCTGACAGTTGTCACGCTGATTGGTGTCGTTACAATCTAACGCATCGCCAA	+
3	GTACTAGAGAACTAGTGCATTAGCTTATTTTTTTGTTATCATGCTAACCACCCGGCG	+
4	AATTGTGATGTGTATCGAAGTGTGTTGCGGAGTAGATGTTAGAATACTAACAAACTC	+
5	ATATGAACGTTGAGACTGCCGCTGAGTTATCAGCTGTGAACGACATTCTGGCGTCTA	-
6	CGAACGAGTCAATCAGACCGCTTTGACTCTGGTATTACTGTGAACATTATTCGTCTC	-
7	CAATGGCCTCTAAACGGGTCTTGAGGGGTTTTTTGCTGAAAGGAGGAACTATATGCG	-
8	TTGACCTACTACGCCAGCATTTTGGCGGTGTAAGCTAACCATTCCGGTTGACTCAAT	-

Kết quả tìm một số đoạn con lặp giống hệt tối đại:

Các đoạn giống hệt tối đại trong 53 đoạn promoter + là: G, AT, CA, TA

Các đoạn giống hệt tối đại trong 53 đoạn promoter - là : AC, AT, AG, CA, CT, TC, TG

Với ngưỡng τ =0.3, chúng tôi phát hiện 97 đoạn con lặp phổ biến như sau:

A; C; T; G; AA; AC; AT; AG; CA; CC; CT; CG; TA; TC; TT; TG; GA;
GC; GT; GG; AAA; AAC; AAT; AAG; ACA; ACC; ACT; ACG; ATA;
ATC; ATT; ATG; AGA; AGC; AGT; AGG; CAA; CAC; CAT
CAG; CCA; CCT; CCG; CTA; CTC; CTT; CTG; CGA; CGC
CGT; CGG; TAA; TAC; TAT; TAG; TCA; TCC; TCT; TCG; TTA
TTC; TTT; TTG; TGA; TGC; TGT; TGG; GAA; GAC; GAT; GAG
GCA; GCC; GCT; GCG; GTA; GTC; GTT; GTG; GGA; GGC; GGT
AACT; AATG; ATGC; CAAT; CTTT; CTTG; TAAC; TACT; TACG
TTTT; TTGA; TTGT; GCAT; GCCT; GCTT

và 58 đoạn con lặp phổ biến tối đại:

AAA; AAG; ACA; ACC; ATA; ATC; ATT; AGA; AGC; AGT; AGG
CAC; CAG; CCA; CCG; CTA; CTC; CTG; CGA; CGC; CGT; CGG
TAT; TAG; TCA; TCC; TCT; TCG; TTA; TTC; TGG; GAA; GAC
GAT; GAG; GCG; GTA; GTC; GTT; GTG; GGA; GGC; GGT; AACT
AATG; ATGC; CAAT; CTTT; CTTG; TAAC; TACT; TACG; TTTT;
TTGA; TTGT; GCAT; GCCT; GCTT

5.2. Thử nghiệm tìm đoạn con lặp xấp xỉ

Với dữ liệu là là 53 đoạn promoter trong GEN và chiều dài của đoạn xấp xỉ làW= 3, chúng tôi thu được vị trí và giá trị của các đoạn xấp xỉ với các tham số cho thuật toán di truyền là: pop_size = 60 nhiễm sắc thể, xác suất lai ghép 0.1, xác suất đột biến là 0.1. Kết quả chúng tôi phát hiện được một đoạn xấp xỉ và vị trí của chúng trong trình tự :

12 : CGC	21 : CGC	24 : CTT	55 : CTC	44 : CGC	29 : CGT
35 : CGC	14 : CTT	42 : CGC	44 : CGC	51 : CGC	44 : CGC
44 : CGC	44 : CGC	12 : CGC	49 : CGC	12 : CGC	25 : CTC
25 : CGC	51 : CGC	16 : CGC	12 : CGC	48 : CGC	52 : CGC
19 : CTT	14 : CTT	7 : CGC	54 : CCC	17 : CCT	24 : CGC
33 : CTC	48 : CGC	4 : CGC	29 : CGC	26 : CGC	19 : CGC
33 : CGC	46 : CGC	3 : CGC	19 : CGT	4 : CGC	3 : CGC
19 : CGC	6 : CTC	27 : CTC	28 : CGC	26 : CGT	48 : CGC
42 : CTC	35 : CTC	50 : CGC	21 : CGC	24 : TGT	

6. MỘT HEURISTIC ĐỂ CẢI TIẾN TỐC ĐỘ TÌM KIẾM

Trong tiến trình tìm các đoạn con lặp giống hệt, chúng tôi lưu lại vị trí của các đoạn con lặp giống hệt và sử dụng các vị trí này để để hỗ trợ tìm kiếm nhanh các đoạn con trong trình tự có điểm cao và khởi tạo ma trận ban đầu cho các nhiễm sắc thể. Kết quả phát hiện các vị trí có mặt các đoạn giống hệt trong các trình tự ADN ứng với promoter của 5 trình tự đầu tiên được liệt kê trong bảng 2.

Bảng 2: Vị trí của các đoạn giống hệt trong các trình tự ADN

Đoạn lặp	Trình tự	Vị trí	Đoạn lặp	Trình tự	Vị trí
TCA	1	10	CACA	4	26,28
TCA	2	24	CACA	5	27
TCA	3	18	ACAGT	1	28
TCA	4	25	ACAGT	2	28
TCA	5	10,26	ACAGT	3	28
GTC	1	9	ACAGT	4	29
GTC	2	4,10,23	ACAGT	5	28
GTC	3	2,21	CGTAAAA	1	2
GTC	4	24	CGTAAAA	2	16
GTC	5	9,22,25	CGTAAAA	3	11
CACA	1	11	CGTAAAA	4	3

7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi phát triển các thuật toán khám phá các trình tự giống hệt và các trình tự xấp xỉ trong tập các trình tự sinh học ADN. Kết quả thử nghiệm trên tập các trình tự ADN

của promoter là rất khả quan. Chúng tôi tiếp tục nghiên cứu việc ứng dụng các đoạn xấp xỉ vào các bài toán nhận dạng GEN để có thể nhận dạng các vùng tín hiệu khác trong GEN như vùng code và non code, exon, intron và xây dựng mô hình toán học cho bài toán nhận dạng GEN.

DEVELOPING ALGORITHMS FOR FINDING THE SIMILAR MOTIF IN DNA SEQUENCES

Hoang Kiem - Do Phuc

ABSTRACT : *The problem of discovering the similar sub-sequences in a set of DNA biological sequences is an important problem of bio-technology. From the positions of similar sub-sequences, we can discover the features of a group of similar function genes or the position for point mutation. In this paper, we analyze two problems of repeated and approximate sub-sequence discovery and employ the large set discovery algorithm for discovering the repeated sub-sequence and the genetic algorithm for discovering the approximate sub-sequence in a set of DNA biological sequence. In the repeated sub-sequence discovery algorithm, we consider the repeated sub-sequence as a large set and employed the Apriori-TID algorithm (Agrawal , 1994) for discovering the maximal large set. In the approximate sub-sequence discovery algorithm, we consider the chromosome of genetic algorithm as a potential solution and employ the genetic algorithm for selecting the right solution. Two proposed algorithms work very well with large data set, therefore they satisfy the demand of the large data set of genes. Besides, we also propose a heuristic for improving the speed of solution discovery. We apply our proposed algorithms to the data of the promoters from the University of Irvine, USA and show the experiment results.*

TÀI LIỆU THAM KHẢO

[1] Anders Krogh: An introduction to Hidden Markov Models for Biological Sequences, Computer Methods in Molecular Bilogogy, Elservier, 1998

[2] Hoang Kiem, Do Phuc: Discovering the binary and fuzzy association rules from database: In the proceedings of the AFSS2000 international conference, Tsukuba, Japan, 2000

[3] Timothy L. Bailey, William Hart, Learning Consensus Patterns in unaligned DNA sequences using Genetic Algorithms, 1993

[4] Timothy L. Bailey, Charles Elkan: Unsupervised Learnng on Multiple Motifs in Biopolymers using Expectation Maximazation, Machine Learning Journal, Kluwer Academic Publisher, 1995

[5] R. Agrawal , R. Srikant Fast Algorithm for mining association rules in large database, Research report RJ, IBM Almaden Research Center, San Jose, CA (1994)