ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN THU TRANG

BÀI TOÁN TÌM KIẾM MOTIF VÀ PHƯƠNG PHÁP TỐI ƯU ĐÀN KIẾN

Ngành : Công nghệ thông tin
Chuyên ngành : Hệ thống thông tin
Mã số : 60480104

TÓM TẮT LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội - 2016

MỤC LỤC

MỞ ĐẦU

Tin sinh học có ứng dụng cao trong cuộc sống, đặc biệt trong lĩnh vực y – dược. Về cơ bản, tin sinh học tập trung vào nghiên cứu và áp dụng các phương pháp cũng như các kĩ thuật trong tin học để giải quyết các bài toán trong sinh học phân tử. Tìm kiếm motif trong các chuỗi gene là một trong những bài toán quan trọng nhất của tin sinh học và thuộc loại NP-khó.

Các thành phần điều hòa gene (gene regulatory elements) được gọi là các DNA motif (về sau gọi là motif cho gọn), chúng chứa nhiều thông tin sinh học quan trọng. Vì vậy việc nhận dạng DNA motif đang là một trong những bài toán quan trọng nhất trong tin sinh học và thuộc loại NP-khó. Chủ yếu, có 2 cách tiếp cận để tìm kiếm motif: các phương pháp thực nghiệm và các phương pháp tính toán. Vì chi phí cao và tốn thời gian nên các phương pháp thực nghiệm ít hiệu quả. Phương pháp tính toán đang được dùng rộng rãi cho dự đoán motif.

Người ta đưa ra nhiều phát biểu cho bài toán tìm kiếm motif, và có nhiều thuật toán nghiên cứu và công bố giải quyết bài toán tìm kiếm motif. Trong luận văn này, tôi trình bày bài toán (ℓ,d) motif. Có nhiều thuật toán đưa ra để giải quyết bài toán (ℓ,d) motif, các thuật toán này có thể chia thành 2 loại đó là thuật toán chính xác và thuật toán xấp xỉ. Các thuật toán chính xác luôn luôn tìm ra những motif trong những chuỗi DNA đầu vào nhưng chỉ hiệu quả với các dữ liệu có kích thước nhỏ và thực hiện mất nhiều thời gian. Các thuật toán xấp xỉ có thể không tìm ra được tất cả các motif nhưng nó chạy hiệu quả với các dữ liệu lớn.

Luận văn đề xuất giải quyết bài toán (ℓ,d) motif theo thuật toán xấp xỉ, bằng việc đề xuất thuật toán tối ưu đàn kiến Ant colony optimization (ACO) để giải quyết bài toán (ℓ,d) motif. Đây là thuật toán mới và lần đầu được đưa vào để giải bài toán (ℓ,d) motif. Thuật toán được đặt tên là F-ACOMotif. Và trong thực nghiệm đã chỉ ra được thuật toán F-ACOMotif tối ưu hơn các thuật toán PairMotif+ và MEME về độ chính xác khi tìm ra (ℓ,d) motif.

Ngoài phần kết luận, cấu trúc nội dung của luận văn bao gồm 4 chương như sau:

Chương 1: Trình bày sơ lược các khái niệm về tin sinh học, bài toán tối ưu tổ hợp và phát biểu bài toán (ℓ,d) motif.

Chương 2: Giới thiệu thuật toán Ant colony optimization (ACO) và một vài thuật toán cập nhật mùi khác nhau trong ACO.

Chương 3: Đề xuất thuật toán, đó là thuật toán Ant colony optimization (ACO) để giải quyết bài toán (ℓ,d) motif.

Chương 4: Đưa ra kết quả thực nghiệm của luận văn, so sánh kết quả của thuật toán ACO với các thuật toán PairMotif+ và thuật toán MEME.

CHƯƠNG 1: TIN SINH HỌC VÀ BÀI TOÁN TÌM KIẾM (L,D) MOTIF

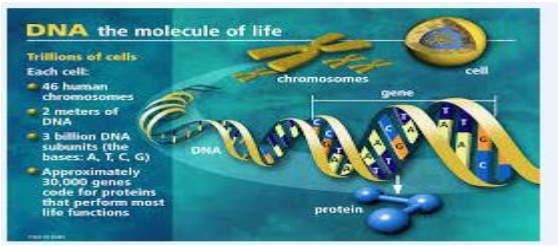
1.1. Tin sinh học

1.1.1 Giới thiệu về tin sinh học

"Tin sinh học là sử dụng toán học, thống kê và khoa học máy tính để giải quyết các vấn đề về sinh học với DNA, chuỗi axit amin và các thông tin có liên quan".

1.1.2 Khái niệm trong sinh học

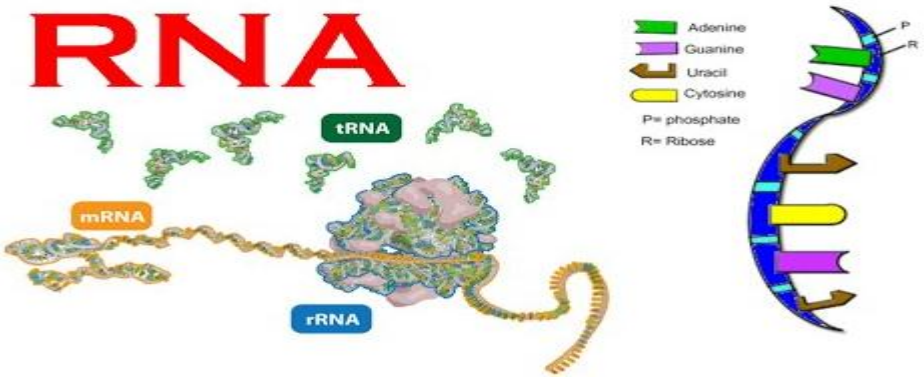
1.1.2.1 DNA



Hình 1.1: DNA phân tử của sự sống

DNA là một phân tử được cấu tạo bởi đường, photphat và bốn nitrogenous bases: adenine, cytosine, guanine và thiamine, được lần lượt viết tắt là A, C, G, và T.

1.1.2.2 RNA

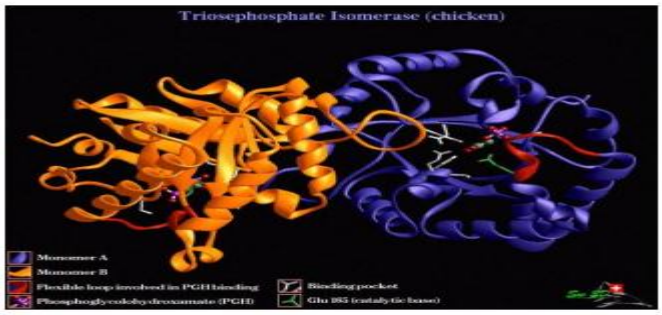


Hình 1.2: Hình ảnh về RNA

RNA (Ribonucleic Acid) là 1 loại acid nucleic (như DNA), RNA cũng có cấu trúc đa phân mà đơn phân là 4 loại nucleotide, tuy nhiên trong RNA nucleotide loại T (pyrimidine thymine) được thay thế bằng U (uracil).

C-G-A-T-T-G-C-A-A-C-G-A-T-G-C DNA
| | | | | | | | | | | | | |
G-C-U-A-A-C-G-U-U-G-C-U-A-C-G RNA

1.1.2.3 Protein

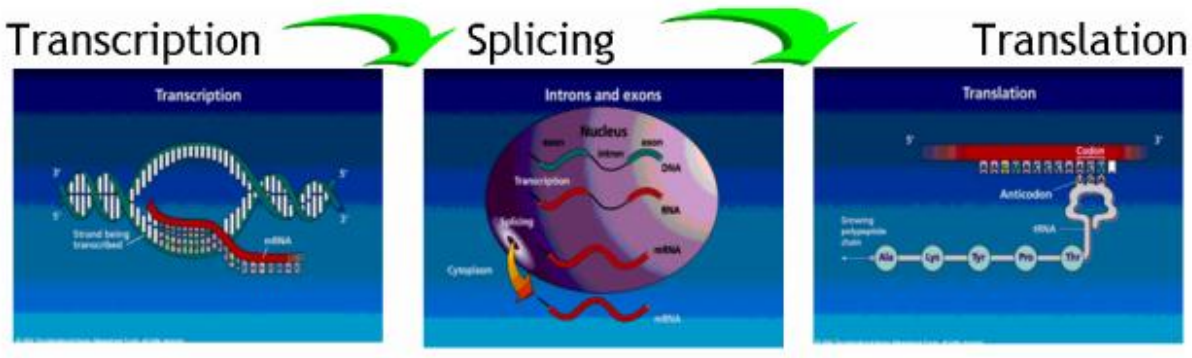


Hình 1.3: Cấu trúc Protein

Các nucleotide trong gene mã hóa cho protein. Các protein cần thiết cho cấu trúc, chức năng và điều chỉnh tế bào, mô và tổ chức, mỗi protein có một vai trò đặc biệt.

1.1.2.4 Quá trình tổng hợp protein

Gồm ba giai đoạn chính : (1) Transcription (phiên mã) (2) Splipcing (ghép mã) (3) Translation (dịch mã) [1] có thể được mô tả như hình dưới:



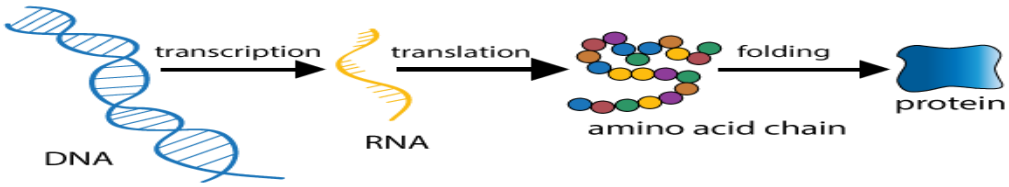
Hình 1.4: Quá trình tổng hợp Protein [1]

1.1.2.5 Một số bài toán trong tin sinh học

Luận văn sẽ tập trung nghiên cứu "Bài toán tìm kiếm motif sử dụng phương pháp tối ưu đàn kiến"

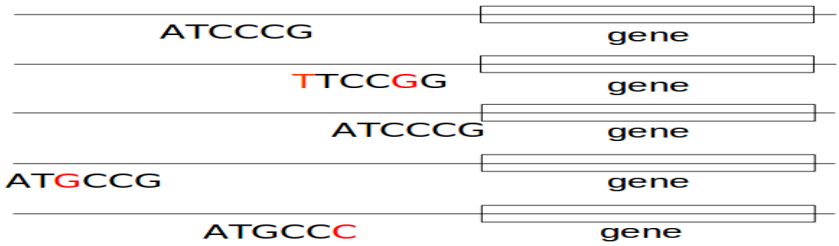
1.1.3 Motif

1.1.3.1 Quá trình điều hòa gen



Hình 1.5: Quá trình tổng hợp Protein

Motif là những đoạn trình tự có kích thước ngắn, lặp đi lặp lại và mang ý nghĩa sinh học.



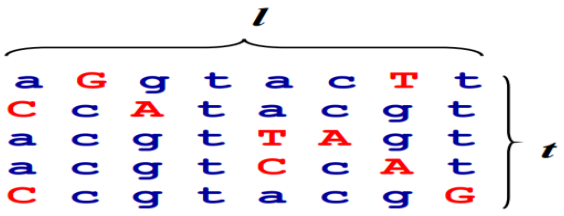
Hình 1.6: Ví dụ về Motif

1.1.3.2 Ý nghĩa của Motif

Có ý nghĩa trong việc kiểm soát sự biểu hiện của gen.

1.1.3.3 Biểu diễn Motif

1.1.3.3.1 Chuỗi hợp nhất và ma trận đặc trưng (Consensus sequence)

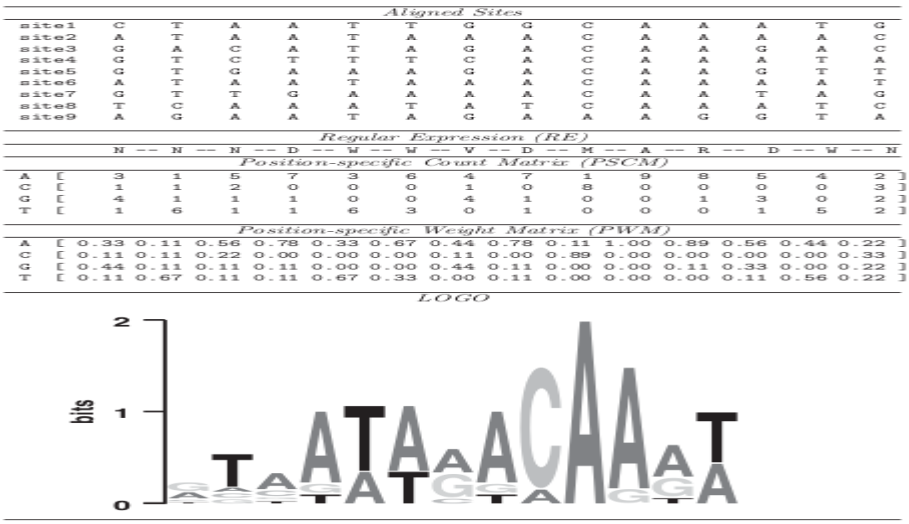


A	3	0	1	0	3	1	1	0
C	2	4	0	0	1	4	0	0
G	0	1	4	0	0	0	3	1
T	0	0	0	5	1	0	1	4

Hình 1.7: Chuỗi hợp nhất

Như ví dụ ở trên ‘ACGTACGT’ là chuỗi hợp nhất.

1.1.3.3.2 Ma trận



Hình 1.8: Biểu diễn Motif

1.1.3.3.3 Biểu tượng

Biểu tượng là cách dùng hình ảnh biểu diễn cho Motif.

Hình 1.9: Biểu diễn Motif dạng sequence

1.2. Bài toán tối ưu tổ hợp và bài toán tìm kiếm (l,d) motif

1.2.1 Bài toán tối ưu tổ hợp

1.2.1.1 Giới thiệu bài toán tối ưu tổ hợp

Mỗi bài toán tối ưu tổ hợp ứng với bộ ba (S, f, Ω), trong đó S là tập hữu hạn các trạng thái (lời giải tiềm năng hay phương án), f là hàm mục tiêu xác định trên S và Ω là tập các ràng buộc.

1.2.1.2 Giới thiệu bài toán người chào hàng

Bài toán được phát biểu như sau:

Có một tập gồm n thành phố (hoặc điểm tiêu thụ) $C = \{c_1, c_2, \dots, c_n\}$ *độ dài đường đi trực tiếp từ* c_i *đến* c_j *là* $d_{i,j}$. *Một người chào hàng muốn tìm một hành trình ngắn nhất từ nơi ở, đi qua mỗi thành phố đúng một lần để giới thiệu sản phẩm cho khách hàng, sau đó trở về thành phố xuất phát.*

1.2.1.3 Các cách tiếp cận giải quyết bài toán tối ưu tổ hợp

1.2.1.3.1 Heuristic cấu trúc

Chúng ta có thể khái quát hóa để mô phỏng dưới dạng thuật toán như sau:

Procedure Heuristic cấu trúc;

Begin

$\boldsymbol{s_p}$ ← chọn thành phần $\boldsymbol{u_0}$ trong $\boldsymbol{C_0}$;

While (chưa xây dựng xong lời giải) **do**

$\boldsymbol{c}$ ←GreedyComponent($\boldsymbol{s_p}$);

$\boldsymbol{s_p} \leftarrow \boldsymbol{s_p}$ ^ $\boldsymbol{c}$;

end-while

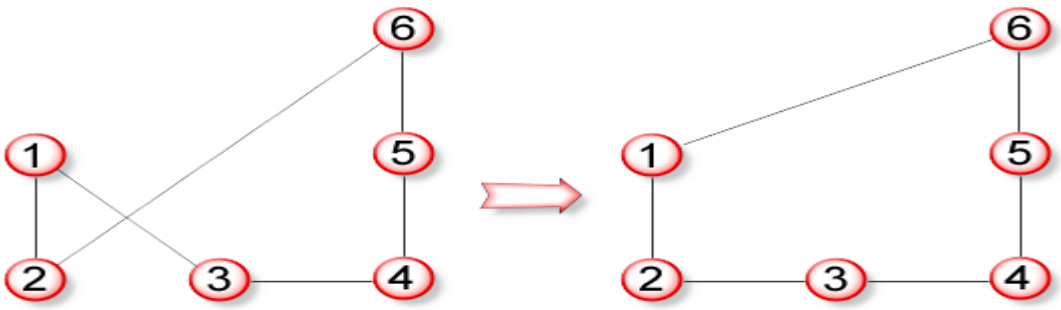
$\boldsymbol{s} \leftarrow \boldsymbol{s_p}$;

Đưa ra lời giải $\boldsymbol{s}$;

End;

Hình 1.10: Phương pháp heuristic cấu trúc

1.2.1.3.2 Tìm kiếm địa phương



Hình 1.11: Lời giải nhận được thông qua tìm kiếm địa phương

1.2.1.3.3 Phương pháp metaheuristic

Phương pháp metaheuristic là một phương pháp heuristic tổng quát được thiết kế, định hướng cho các thuật toán cụ thể (bao gồm cả heuristic cấu trúc và tìm kiếm địa phương).

1.2.1.3.4 Phương pháp Memetic

Proedure Thuật toán memetic-EC;
Begin
Initialize: Tạo ra quần thể đầu tiên;
while điều kiện dừng chưa thỏa mãn **do**
Đánh giá các cá thể trong quần thể;
Thực hiện tiến hóa quần thể nhờ các toán tử cho trước;
Chọn tập con $\boldsymbol{\Omega_{il}}$ để cải tiến nhờ thủ tục tìm kiếm địa phương;
for mỗi cá thể trong $\boldsymbol{\Omega_{il}}$**do**
Thực hiện tìm kiếm địa phương;
end-for
Chọn phần tử tốt nhất;
end-while;
Đưa ra lời giải tốt nhất;
End;

Hình 1.12: Thuật toán memetic sử dụng EC

1.2.2 Phát biểu bài toán tìm kiếm (l,d) motif

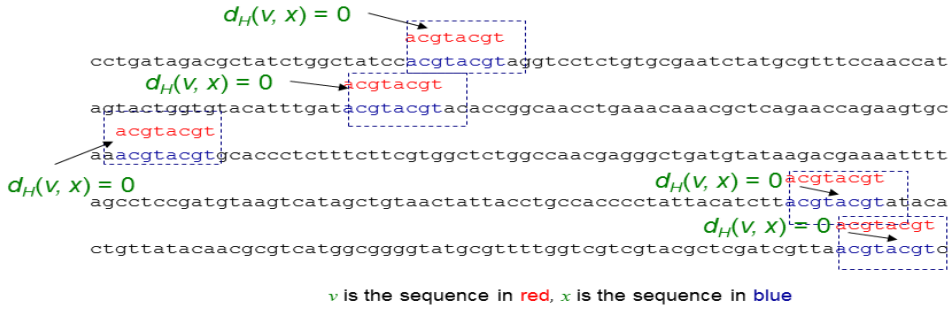
Trước khi đưa ra bài toán, luận văn đưa ra định nghĩa sau:

Định nghĩa: (Haming distance)

Cho x và y tương ứng là hai xâu độ dài l và n, khoảng cách Hamming $d_H(x,y)$ được xác định như sau:

a) $d_H(x,y)$ = số vị trí khác nhau của x và y nếu $l=n$

b) $d_H(x,y)$ = min{$d_H(x,m)$/ m là xâu con độ dài l của y} nếu $l < n$



- $TotalDistance(v, DNA) = 0$

Hình 1.13: Ví dụ khoảng cách hamming

Có nhiều phát biểu cho bài toán tìm kiếm motif. Điển hình có thể kể đến 3 bài toán tìm kiếm motif như sau [14]: Simple Motif Search, (l,d) Motif Search (Planted Motif Search) và Edited Motif Search

Trong luận văn này, chúng tôi sẽ tập trung nghiên cứu bài toán (l,d) Motif Search (LDMS) hay chính là bài toán Planted Motif Search (PMS) từ nay sẽ gọi là bài toán PMS.

Bài toán PMS được phát biểu như sau:

Cho một tập hợp N chuỗi S =$\{S_1, S_2,..,S_N\}$, trong đó mỗi phần tử được lấy ra từ tập $\sum$=$\{A, C, G, T\}$ và hai số nguyên không âm ℓ và d, thỏa mãn $0 \leq d < \ell < n$.

Bài toán (ℓ,d)-motif là tìm chuỗi m độ dài ℓ từ $\sum$ và một tập chuỗi con M=$\{m_1, m_2, .., m_N\}$ trong đó, m_i tương ứng là chuỗi con của S_i có cùng độ dài ℓ sao cho $d_H(m, S_i) \leq d$

Ví dụ:

Mô tả cho việc tìm kiếm (ℓ,d) – motif. Giả sử S là tập gồm 3 chuỗi S_1, S_2, S_3 trong đó:

S_1: GCGCGAT

S_2: CAGGTGA

S_3: CGATGCC

Giả sử cho 2 tham số đầu vào ℓ = 3; và d = 1. Sau khi S được kiểm tra bằng một thuật toán tìm kiếm (ℓ,d) – motif, ta có thể tìm được motif m là: GAT và GTG

Hiện nay có hai phương pháp để tìm kiếm motif:

- Bằng thực nghiệm trong sinh học: Tốn thời gian, chi phí cao, mất nhiều công sức, độ chính xác cao.
- Bằng tính toán trong tin học: Hoàn toàn có thể thực hiện được trong thời gian và chi phí thấp nhưng chỉ đưa ra được các chuỗi có khả năng là motif.

Với hướng tiếp cận bằng tính toán, có hai phương pháp tìm kiếm là chính xác và gần đúng. Các thuật toán chính xác luôn luôn tìm ra những motif trong những chuỗi DNA đầu vào nhưng chỉ hiệu quả với các dữ liệu có kích thước nhỏ và thực hiện mất nhiều thời gian. Một số thuật toán chính xác phổ biến hiện nay: PMS6, PMS5, Pampa, PMSPrune, Voting, RISSOTO, MITRA, PairMotif. Các thuật toán xấp xỉ có thể không tìm ra được tất cả các motif nhưng nó chạy hiệu quả với các dữ liệu lớn, tiêu biểu có: MEME, Gibbs sampler, Genetic Algorithm (GA), PairMotif+.

CHƯƠNG 2. GIỚI THIỆU VỀ THUẬT TOÁN ANT COLONY OPTIMIZATION (ACO)

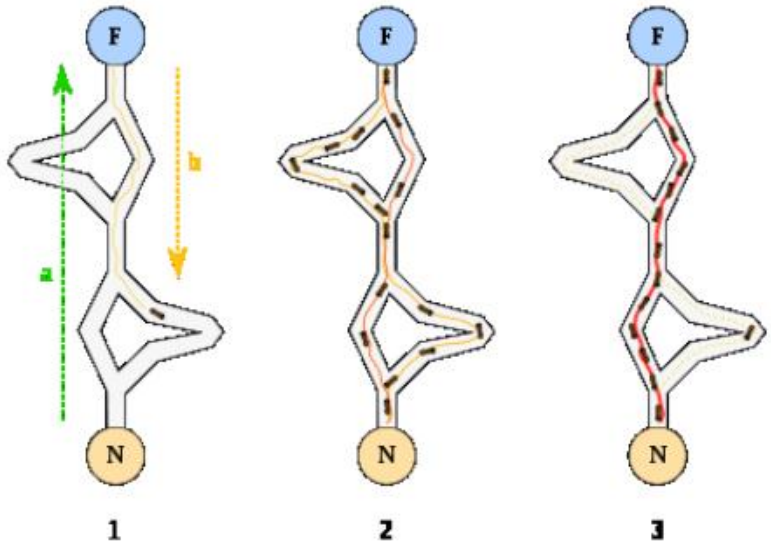
2.1 Giới thiệu về thuật toán ACO

Tối ưu đàn kiến (*Ant Colony Optimization – ACO*) là một phương pháp metaheuristic được đề xuất bởi Dorigo vào năm 1991dựa trên ý tưởng mô phỏng cách tìm đường đi từ tổ tới nguồn thức ăn và ngược lại của các con kiến tự nhiên để giải gần đúng bài toán TƯTH NP-khó.

2.2 Mô hình mô phỏng của thuật toán

2.2.1 Kiến tự nhiên

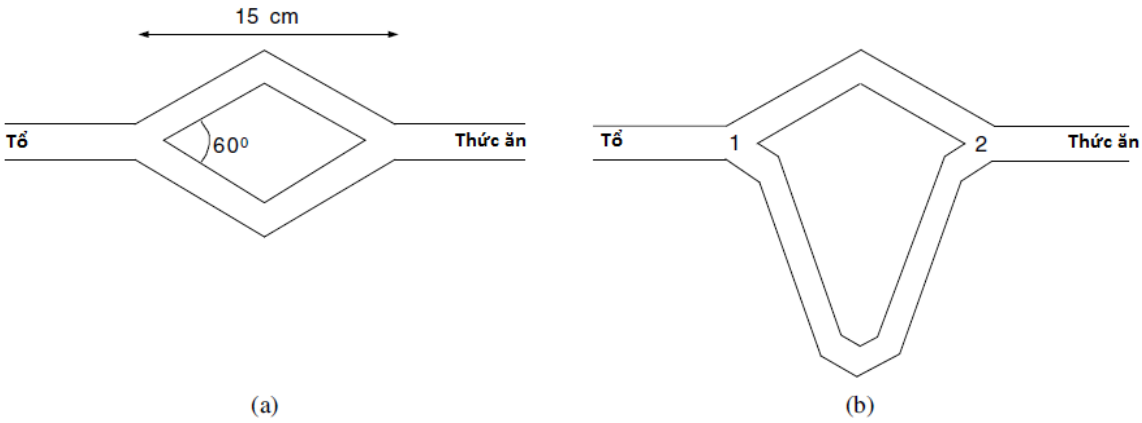
Kiến chịu ảnh hưởng của các vết mùi của các con kiến khác chính là ý tưởng thiết kế thuật toán ACO.



Hình 2.1: Thể hiện hành vi của mỗi con kiến trong tự nhiên

Thí nghiệm trên cây cầu đôi

Thực nghiệm này cho thấylà sự tương tác cục bộ giữa các con kiến với thông tin gián tiếp là vết mùi để lại cho phép điều chỉnh hoạt động vĩ mô của đàn kiến.

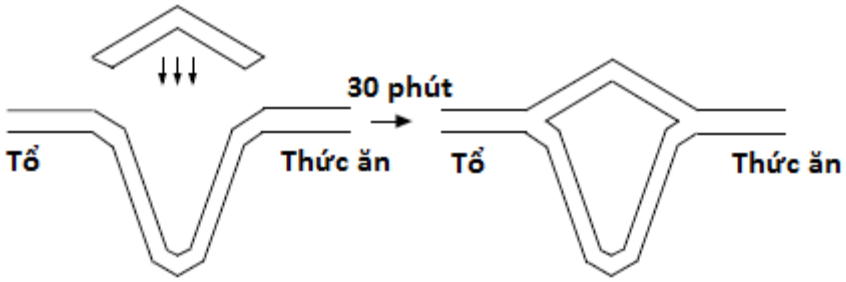


Hình 2.2: Thực nghiệm cây cầu đôi

(a) Hai nhánh có độ dài bằng nhau. (b) Hai nhánh có độ dài khác nhau.

Trong thực nghiệm thứ hai (xem hình 2.2 b), minh chứng bầy kiến đã sử dụng phương thức thăm dò, tìm đường mới.

Việc bay hơi vết mùi là cơ chế tiện lợi cho việc tìm đường mới, nghĩa là việc bay hơi có thể giúp kiến quên đi đường đi tối ưu địa phương đã được tìm thấy trước đây để tìm khám phá đường đi mới, tốt hơn.



Hình 2.3: Thí nghiệm bổ xung

(Ban đầu chỉ có một nhánh và sau 30 phút thêm nhánh ngắn hơn)

2.2.2 Kiến nhân tạo (Artificial Ant)

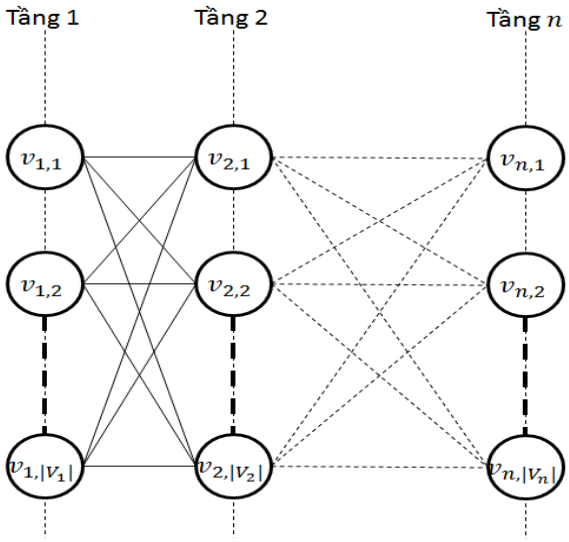
Kiến nhân tạo (về sau trong luận văn ta sẽ gọi đơn giản là kiến) có bộ nhớ riêng, có khả năng ghi nhớ các đỉnh đã thăm trong hành trình và tính được độ dài đường đi nó chọn. Ngoài ra, kiến có thể trao đổi thông tin với nhau, thực hiện tính toán cần thiết, cập nhật mùi…

2.3 Trình bày giải thuật

2.3.1 Đồ thị cấu trúc

Xây dựng đồ thị cấu trúc

Ta gọi đồ thị $G = (V, E, H, \tau)$ là *đồ thị cấu trúc* của bài toán tối ưu tổ hợp, trong đó V là tập đỉnh, E là tập cạnh, H và τ là các thông tin gắn với cạnh.



Hình 2.4: Đồ thị cấu trúc tổng quát cho bài toán cực trị hàm $\mathbf{f(x_1, \dots x_n)}$

2.3.2 Trình bày thuật toán ACO cơ bản

Các bước thực hiện của thuật toán ACO được mô tả trong hình 2.5:

Procedure Thuật toán ACO;

Begin

Khởi tạo tham số, ma trận mùi, khởi tạo $\boldsymbol{m}$ con kiến;

repeat

for $\boldsymbol{k = 1}$ **to** $\boldsymbol{m}$ **do**

Kiến $\boldsymbol{k}$ xây dựng lời giải;

end-for

Cập nhật mùi;

Cập nhật lời giải tốt nhất;

Until (Điều kiện kết thúc);

Đưa ra lời giải tốt nhất;

End;

Hình 2.5: Đặc tả thuật toán ACO

2.3.3 Thông tin Heuristic

Giúp kiến có thể xây dựng được các hành trình tốt ngay trong giai đoạn đầu.

2.3.4 Quy tắc cập nhật vết mùi

2.3.4.1 Thuật toán AS

2.3.4.2 Thuật toán ACS

2.3.4.3 Thuật toán Max-Min

2.3.4.4 Thuật toán Max- Min trơn

2.3.5 ACO kết hợp với tìm kiếm địa phương

Thực nghiệm cho thấy khả năng kết hợp tìm kiếm địa phương cải tiến được lời giải là khá cao.

2.3.6 Số lượng kiến

Nếu sử dụng số lượng kiến ít, trong giai đoạn đầu sẽ không tìm được lời giải tốt và như vậy, việc cập nhật mùi được cập nhật dựa trên các lời giải không tốt.

2.3.7 Tham số bay hơi

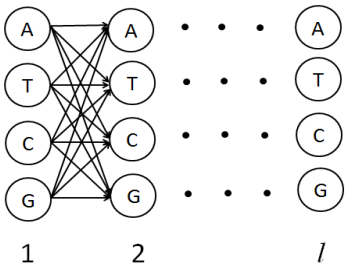
Tham số bay hơi sẽ được xác lập có giá trị lớn, điều này giúp kiến quên đi những lời giải đã xây dựng, tập trung công việc tìm kiếm xung quanh lời giải tốt mới được xây dựng.

CHƯƠNG 3: THUẬT TOÁN ĐỀ XUẤT

3.1 Thuật toán tối ưu đàn kiến

3.2. Xây dựng đồ thị cấu trúc

Để tìm motif có độ dài *l*, đồ thị có 4ℓ đỉnh được xếp thành 4 hàng và ℓ cột. Mỗi đỉnh tại vị trí *(u, j)* được gán nhãn của một loại nucleotide tương ứng như trong hình 2.



Hình 3.1: Đồ thị cấu trúc tìm motif độ dài ℓ

3.3. Thông tin heuristic

- Ở các đỉnh của cột đầu, thông tin heuristics là tần số (frequency) xuất hiện nucleotide tương ứng trong tập dữ liệu *S*.
- Thông tin heuristics ở các cạnh $e_j(u, v)$ là tần số xuất hiện thành phần *uv* trong tập *S*. Chúng chỉ gồm 16 đại lượng $\eta_{u,v}$, (*u*,*v*) ∈∑x∑

3.4. Xây dựng lời giải tuần tự

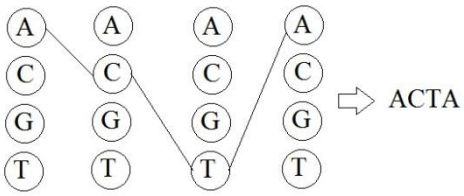
Trong mỗi lần lặp, mỗi con kiến chọn ngẫu nhiên một nút xuất phát *u* ở cột đầu với xác suất P_u^1

$$P_u^1 = \frac{\tau_u^{1*}\,\eta_u}{\sum_{j\,\in\,\{A,C,G,T\}} \tau_v^{1*}\,\eta_v} \qquad (3.1)$$

Trong đó, η_u là thông tin heuristic được tính theo tần số của nucleotide *u* trong dữ liệu và τ_u^1 là vết mùi đã được cập nhật tại đỉnh. Ngoài ra, một con kiến di chuyển từ đỉnh *(u, j)* tới đỉnh *(v, j+1)* theo xác xuất sau:

$$P_{uv}^j = \frac{\tau_{uv}^{j*}\,\eta_{u,v}}{\sum_{r\,\in\,\{A,C,G,T\}} \tau_{ur}^{j*}\,\eta_{u,r}} \qquad (3.2)$$

Trong đó, $\eta_{v,j}$ là thông tin heuristic của canh e_j(u, v).



Hình 3.2: Cách xây dựng đường đi của kiến

3.5. Quy tắc cập nhật mùi (pheromone update rule)

Các vết mùi τ_u^1 trên mỗi đỉnh u ở cột đầu và $\tau_{u,v}^j$ trên các cạnh $e_j(u,\ v)$ ban đầu được khởi tạo bằng τ_{max} cho trước. Sau mỗi vòng lặp, vết mùi τ_u^1 ở mỗi đỉnh u của cột đầu được cập nhật mùi theo Eq (3.3):

$$\tau_u^1 \leftarrow (1-\rho)\tau_u^1 + \Delta_u^1, \qquad (3.3)$$

Trong đó: $\Delta_u^1 = \begin{cases} \rho\tau_{max} & u \in \text{giải pháp tốt nhất} \\ \rho\tau_{min} & \text{giải pháp khác} \end{cases}$,

Trong đó τ_{max}, τ_{min} và ρ là các tham số chọn trước.

Vết mùi ở các cạnh $e_j(u,\ v)$ được cập nhật theo Eq (3.4)

$$\tau_{u,v}^j \leftarrow (1-\rho)\tau_{u,v}^j + \Delta_{u,v}^j, \qquad (3.4)$$

Trong đó: $\Delta_{u,v}^1 = \begin{cases} \rho\tau_{max} & uv \in \text{giải pháp tốt nhất} \\ \rho\tau_{min} & \text{giải pháp khác} \end{cases}$

3.6. Tìm kiếm địa phương (local search)

Sau khi các con kiến tìm được lời giải trong vòng lặp, các lời giải có hàm mục tiêu $\sum_{i=1}^{N} d_H(m, S_i)$ nhỏ nhất được áp dụng tìm kiếm địa phương bởi thủ tục lặp.

Với mỗi motif tiềm năng (potemtial motif) S_m, dùng tập Q(S_m) để chứa kết quả tìm kiếm (), và thủ tục lặp này thực hiện như sau:

Bước 1: khởi tạo Q(S_m) = {S_m};

Bước 2. Thực hiện lặp:

For mỗi *i=1,...,l* thực hiện:

2.1. Thay ký tự (letter) ở vị trí thứ *i* của S_m lần lươt bởi một trong ba ký tự còn lại trong tập $\sum$ để có S_p;

2.2. Tính $H_d(S_p)$;

2.3. Nếu $H_d(S_p) \leq H_d(S_m)$ thì $S_m \leftarrow S_p$ và Q(S_m) = {S_p};

Until khi không thể cải thiện được hàm mục tiêu nữa.

Sau khi áp dụng tìm kiếm địa phương cho các motif tiềm năng trong mỗi lần lặp, các tập Q(S_m) có hàm mục tiêu nhỏ nhất hoặc gần nhỏ nhất được hợp lại thành tập Q các lời giải được xem là tốt nhất sau khi lọc các lời giải có cùng vị trí liên kết (chỉ giữ lại một motif). Dựa trên tập Q, các vết mùi trên đồ thị được cập nhật theo các Eq(3.3) và (3.4) để dùng cho vòng lặp kế tiếp.

Sau khi có tập Q là tập các motif có điểm khoảng cách hamming nhỏ nhất, ta tiến hành kiểm tra các motif có $d_H(m,S_i)$ <=d thì ta in ra motif (ℓ,d).

Thuật toán dừng khi thực hiện xong số vòng lặp chọn trước. Các vị trí liên kết ứng với các motif trong Q cho ta xác định được instance của mottif. $d_H(m, S_i) =$ min{ $d_H(m, m_i)$: m_i độ $dài\ \ell\ là\ một\ chuỗi\ con\ của\ S_i$ }

Các xâu m_i cực tiểu (minimize) sẽ là instance của m và vị trí của nó trong S_i tương ứng sẽ là vị trí liên kết.

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Bộ dữ liệu chuẩn

Để chạy thực nghiệm, luận văn sử dụng 13 bộ dữ liệu: trong đó 4 bộ dữ liệu là dữ liệu sinh học đã được công bố, được lấy từ bài báo [20]. Đây là bộ dữ liệu mà tác giả bài báo [16] sử dụng để chạy chương trình.

4.2 Tiến hành chạy thực nghiệm trên hệ điều hành ubuntu

Chương trình được viết bằng ngôn ngữ Perl chạy trên máy Desktop cấu hình CPU intel core i5 2.5Ghz Ram 8GB, sử dụng hệ điều hành Ubuntu 12.04. Thực nghiệm so sánh hiệu quả thuật toán với Pairmotif+, MEME trên cùng các bộ dữ liệu, số kiến được dùng là 10.

4. 3 Kết quả chạy thực nghiệm và đánh giá

4.3.1 Kết quả thực nghiệm

Thực nghiệm F-ACOMotif trên tập dữ liệu Tompa. Dữ liệu Tompa được tải về theo địa chỉ sau: http://bio.cs.washington.edu/assessment/download.html

Thực nghiệm chạy với 2 tham số ℓ = 21 và d = 8 (các tham số ℓ, d được lựa chọn theo dữ liệu thực), $\tau_{max} = 1.0$; $\tau_{min} = {}^{1}/_{4^{l}}$. Các tham số khác như sau:

n (số kiến)	(vòng lặp)	ρ(tham số bay hơi)
10	500	0.02

Bảng 4. 1: Các tham số chạy F-ACOMotif cho thực nghiệm

Mus 05	Position: 360 281 141 414 Motif : AGAGGTAAAAAAAAAGGAGAG Position: 360 281 141 414 Motif : AGAGGTAAAAAAAAAGGGGAG
Mus07	Position: 1402 1455 1343 336 Motif : CCCCCCCCCCAACACCTGCTG Position: 1239 701 99 647 Motif : TACACACACACACCCACACAC Position: 94 101 891 850 Motif : CTATGAGTCCAAAGCCAGCCT Position: 1239 701 99 647 Motif : TACAGACACACACACACACAC Position: 1402 1455 1343 336 Motif : CCACCCCCCCAACACCTGCTG
hm19	Position: 377 447 358 282 113 Motif : AGGGCGGGGCAGTGTGATGGG Position: 389 234 425 30 142 Motif : TGGGATGGGGCCGGGCGGGGG Position: 423 366 131 71 63

	Motif : CTCTCCTCCCACCACCCACAG Position: 378 448 359 283 114 Motif : GGGCGGGGCACTGTGATGGGA Position: 389 234 425 30 142 Motif : TGGGATGCGGCCGGGTGGGGG Position: 389 234 425 30 142 Motif : TGGGATGCGGCCGGGCGGGGG Position: 389 234 425 30 142 Motif : TGGGATGGGGCGGGGCGGGGG Position: 377 447 358 282 113 Motif : AGGGCGGGGCACTGTGATGGG Position: 423 366 131 71 63 Motif : CTCTCCTCCCCCCACCCACAG Position: 389 234 425 30 142 Motif : TGGGATGCGGCGGGGTGGGGG Position: 389 234 425 30 142 Motif : TGGGATGCGGCGGGGCGGGGG Position: 174 364 129 76 61 Motif : CCCCCTCCTCCCACCACCCAC Position: 174 364 129 76 61 Motif : CCCTCTCCTCCCACCACCCAC Position: 378 448 359 283 114 Motif : GGGCGGGGCAGTGTGATGGGA
hm22	Position: 20 83 306 199 384 131 Motif : GACAGAGGGCGGGTCCCTCCC Position: 370 404 77 473 159 54 Motif : AGGCAGGAAGGAGAAGGGAGG Position: 371 405 78 474 160 55 Motif : GGCAGGAAGGAGAAGGGAGGG Position: 370 404 77 473 159 54 Motif : AGGCAGGAATGAGAAGGGAGG Position: 121 184 124 186 34 122 Motif : GGGACACTGCAGAGCCTGGGG Position: 122 185 125 366 35 123 Motif : GGGCACGGCAGAGCCTGGGGA Position: 371 405 78 474 160 55 Motif : GGCAGGAATGAGAAGGGAGGG Position: 122 185 125 366 35 123 Motif : GGACACGGCAGAGCCTGGGGA Position: 122 185 125 366 35 123 Motif : GGCCACGGCAGAGCCTGGGGA Position: 121 184 124 186 34 122 Motif : TGGACACTGCAGAGCCTGGGG Position: 121 184 124 186 34 122 Motif : AGGACACTGCAGAGCCTGGGG

Bảng 4. 2: Kết quả thực nghiệm trên cơ sở dữ liệu TRANSFAC

Nhận xét:

Từ kết quả thực nghiệm cho thấy, F-ACOMotif cho kết quả là một tập các motif và một tập vị trí các thể hiện của motif. Ở đây luận văn không in ra danh sách các thể

hiện mà chỉ in ra vị trí của các thể hiện, vì quá nhiều thể hiện, nếu in ra các thể hiện sẽ rất rối.

4.3.2 So sánh và đánh giá

4.3.2.1 So sánh với MEME

Các tham số chạy F-ACOMotif lần lượt như sau:

n (số kiến)	(vòng lặp)	ρ(tham số bay hơi)
10	500	0.004

Bảng 4.3: Tham số chạy F-ACOMotif

$\tau_{max} = 1.0;\ \tau_{min} = {}^{1}/_{4^l}$

(ℓ,d)	MEME	F-ACOMotif
(9,2)	GTTCAGCGT	GTTCAGCGT
(15,4)	AGCGAGCCTTTACAA	ATCGAGCTTTGACAA
(18,5)	AGTGAAAGACTTGTACCT	AGTGAAAGACTTGTACCT
(21,6)	GCGCGACGGACTTACGTCTTC	GCGCGACGGACTTACGTCTTC
(24,7)	AATTACTTTTCGATAAAGTGGATC	AATTACTTTCCGATAAAGTGGATC

Bảng 4.4: Kết quả so sánh F-ACOMotif với thuật toán MEME

Nhận xét:

Từ bảng so sánh kết quả, ta nhận thấy rằng với các tham số (l,d) lần lượt là: (9,2); (18,5); (21,6); (24,7) thì F-ACOMotif và MEME kết quả gần giống nhau chỉ khác kết quả duy nhất ở 1 tham số là (15,4) tuy nhiên không lớn lắm. Do đó, ta có thể kết luận F-ACOMotif tìm được motif chính xác tương đương MEME.

4.3.2.2 Kết quả so sánh F-ACOMotif với Pairmotif+ và MEME trên tập dữ liệu thực

Data	(ℓ,d)	Pairmotif+	MEME	F-ACOMotif	Motif công bố
DHFR	(11, 3)	GCGCCAAACTT	-	ATTTCGCGCCA	ATTTCGCGCCA
Preproinsulin	(15, 4)	TGCAACCTCAGCCCC	-	CAGACCCAGCACCAG	CAGCCTCAGCCCCCA
Metallothionein	(15, 4)	CTCTGCACCCGGCCC	-	CTCTGCACCCGGCCC	CTCTGCACRCCGCCC
Yeast ECB	(16, 5)	TTACCCAGTAAGGAAA	TTTCCCGTTTAGGAAA	TTTCCCGTTTAGGAAA	TTTCCCNNTNAGGAAA

Bảng 4.5: Kết quả so sánh F-ACOMotif với MEME và PairMotif+

Nhận xét:

Từ bảng kết quả so sánh F-ACOMotif với MEME và PairMotif+ ta nhận thấy: MEME tìm ra motif với thời gian rất ngắn. Nhưng hạn chế của MEME là với những chuỗi đầu vào có độ dài quá lớn, MEME không tìm được motif.

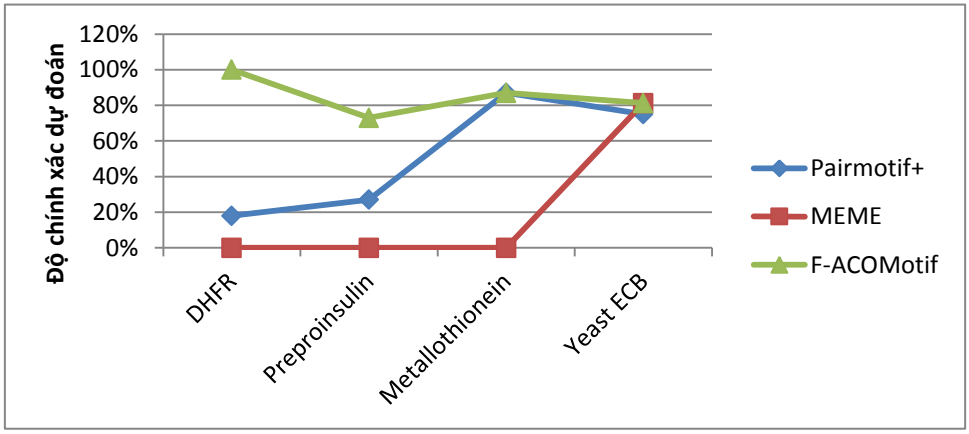
Từ bảng kết quả so sánh F-ACOMotif với MEME và PairMotif+ ta tiến hành lập bảng so sánh độ chính xác của motif dự đoán:

Data	(ℓ,d)	Pairmotif+	MEME	F-ACOMotif
DHFR	(11, 3)	18%	0%	100%
Preproinsulin	(15, 4)	27%	0%	73%
Metallothionein	(15, 4)	87%	0%	87%
Yeast ECB	(16, 5)	75%	81.25%	81.25%

Bảng 4.6: So sánh độ chính xác của motif dự đoán

Nhận xét:

Từ bảng so sánh độ chính xác của motif dự đoán ta nhận thấy rằng F-ACOMotif dự đoán motif chính xác hơn so với MEME và Pairmotif+



Hình 4.1: Đồ thị so sánh độ chính xác của F-ACOMotif so với PairMotif+ và MEME

Nhận xét:

Từ kết quả so sánh F-ACOMotif với MEME và PairMotif+ có thể thấy rằng F-ACOMotif hiệu quả hơn thuật toán MEME và PairMotif+ về độ chính xác khi tìm ra Motif so với motif thực.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

KẾT LUẬN

Bài toán tìm kiếm (ℓ,d) motif là một bài toán có ý nghĩa trong tin sinh học, nó đóng vai trò quan trọng trong việc xác định vị trí liên kết trong quá trình phiên mã trong chuỗi DNA. Xác định được các Motif và các thể hiện tương ứng của nó có ý nghĩa rất quan trọng, từ đó các nhà nghiên cứu sinh học có thể phát hiện ra các tương tác giữa DNA và Protein, điều hòa gen cũng như sự phát triển và tương tác trong một tế bào.

Trong luận văn này, chúng tôi đã dựa trên ý tưởng của thuật toán ACOMotif đề xuất thuật toán mới là F-ACOMotif để giải quyết bài toán (ℓ,d) motif.

So sánh thực nghiệm với thuật toán MEME và PairMotif+, cho thấy thuật toán F-ACOMotif cho kết quả tốt hơn khi tìm ra motif với độ chính xác cao so với motif thực được công bố trong thực nghiệm sinh học.

HƯỚNG PHÁT TRIỂN

Luận văn đề xuất thuật toán ACO để giải quyết bài toán tìm kiếm (ℓ,d) motif và cho lời giải tốt. Tuy nhiên, thời gian chạy thuật toán để cho lời giải tốt còn chậm. Và F-ACOMotif chỉ cho hiệu quả đối với các tập dữ liệu với số chuỗi đầu vào nhỏ hơn 10. Trong tương lai sẽ nghiên cứu cải tiến bài toán tìm kiếm (ℓ,d) motif với thời gian thực hiện ngắn và độ chính xác so với motif thực sẽ cao hơn.