ĐẠI HỌC QUỐC GIA HÀ NỘI

# TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

**BÁO CÁO HỘI NGHỊ SINH VIÊN NGHIÊN CỨU KHOA HỌC**

**NĂM 2021**

**Tên đề tài:**

**Tính toán thông minh cho bài toán tìm motif DNA**

NHÓM SINH VIÊN THỰC HIỆN:

1. Đỗ Hoàng Khánh, 17020028, QH-2017-I/CQ-CAC
2. Nguyễn Tuấn Hoàng, Dartmouth College
3. Đào Quang Thái Dương, 20020006, K65CB

NGƯỜI HƯỚNG DẪN KHOA HỌC:

1. PGS.TS. Hoàng Xuân Huấn
2. TS. Hoàng Thị Điệp
3. TS. Đỗ Đức Đông

Năm 2021

Tính toán thông minh cho bài toán tìm motif DNA

# Tóm tắt

Bài toán tìm kiếm motif DNA, tìm đoạn mẫu ngắn xuất hiện phổ biến trên một tập các chuỗi DNA cho trước, là một trong những bài toán quan trọng trong lĩnh vực tin sinh, đóng vai trò hỗ trợ tìm hiểu rõ cơ chế điều hòa gene và giúp phát triển những phương thuốc mới. Nghiên cứu này đề xuất ACOSite, giải gần đúng bài toán tìm kiếm motif DNA, sử dụng tiếp cận tối ưu hóa đàn kiến. ACOSite được thiết kế và phát triển dựa trên việc phân tích ưu điểm và nhược điểm của ACOMotif, là thuật toán kiến tìm motif trước đây của nhóm. Thực nghiệm trên các bộ dữ liệu sinh học cho thấy ACOSite cho điểm F1 hứa hẹn khi so sánh với thuật toán ACOMotif và thuật toán MEME - một trong các thuật toán hàng đầu trong việc giải bài toán này.

# Giới thiệu

Một motif là một chuỗi nucleotide hoặc amino acid xuất hiện phổ biến trong một tập hợp chuỗi phân tử sinh học. Motif được cho là có một chức năng sinh học nào đó. Vấn đề tìm kiếm motif và xác định vị trí liên kết (binding sites) trên các chuỗi DNA là một trong những bài toán quan trọng trong lĩnh vực tin sinh, và theo Li et al. (1999), đây là một bài toán NP-khó. Tổng quan, có hai hướng tiếp cận chính để tìm kiếm motif: phương pháp thực nghiệm và phương pháp tính toán. Do rào cản về chi phí lớn và tốn thời gian nên phương pháp thực nghiệm gặp nhiều hạn chế, dẫn đến các phương pháp tính toán liên tục được đề xuất để giúp hỗ trợ dự đoán motif. Do tìm motif là một bài toán NP-khó, những nghiên cứu sử dụng phương pháp tính toán cho đến thời điểm hiện tại tập trung vào các cách dự đoán xấp xỉ gần đúng để giải bài toán này.

Hiện nay, một vài thuật toán đã được đề xuất để giải bài toán một cách tối ưu nhất có thể. Một hướng đi phổ biến cho các thuật toán ấy là việc sử dụng phương thức xác suất (probabilistic method), có thể kể đến như MEME (Bailey et al., 2006) hay Gibbs Motif Sampler (Hertz et al., 1999). Những thuật toán sử dụng phương thức xác suất thường có thời gian chạy rất nhanh, nhưng chúng lại dễ bị “mắc kẹt” ở tối ưu cục bộ (local optima), do đó hiệu quả của chúng không đồng đều ở các bộ dữ liệu khác nhau.

Bên cạnh đó, một hướng đi phổ biến khác có thể kể tới việc sử dụng phương pháp tối ưu đàn kiến (Ant Colony Optimization, gọi tắt là ACO). Ví dụ, thuật toán MFACO được đề xuất bởi Bounama et al. (2010) sử dụng Consensus Score (điểm đồng thuận) để tìm kiếm motif và Information Content (nội dung thông tin, sau đây viết tắt là IC) để xác định vị trí liên kết của chúng. Hoang (2015) đề xuất thuật toán ACOMotif với việc sử dụng hàm mục tiêu theo Hamming Score với các thiết kế đồ thị cấu trúc, quy tắc cập nhật mùi, thông tin heuristic, và phương thức tìm kiếm cục bộ phù hợp với hàm Hamming này.

Trong nghiên cứu này, từ việc phân tích thuật toán ACOMotif, chúng tôi đề xuất một thuật toán mới theo tiếp cận tối ưu đàn kiến, gọi tên là ACOSite, sử dụng IC làm hàm mục tiêu, cùng với đồ thị cấu trúc và phương thức tìm kiếm cục bộ mới. Cách tiếp cận mới này cho ra độ chính xác cao khá hiệu quả để tối ưu trên hàm mục tiêu IC.

Phần còn lại của bài này được trình bày như sau. Mục 2 phát biểu bài toán tìm motif DNA, cùng với việc giới thiệu các hàm chấm điểm thông dụng. Mục 3 giới thiệu phương pháp chung cho các thuật toán tìm motif sử dụng tiếp cận ACO, và sau đó, phân tích những hạn chế của thuật toán ACOMotif. Dựa trên những điều đó, thuật toán mới sẽ được đề xuất ở Mục 4, trước khi trình bày kết quả thực nghiệm ở Mục 5. Một số kết luận sẽ được trình bày ở mục cuối cùng.

# Phát biểu bài toán

Bài toán tìm kiếm motif có thể định nghĩa như sau:

Gọi là một tập hợp các chuỗi DNA có cùng độ dài , trong đó . Trong báo cáo này, chúng tôi chỉ xét các chuỗi DNA, do đó có giá trị thuộc bảng chữ cái . Gọi là độ dài của motif cần tìm ().

Bài toán yêu cầu tìm ra tập vị trí liên kết (điểm bắt đầu của chuỗi liên kết) tiềm năng nhất của một motif ký hiệu là , trong đó là số lượng vị trí tìm được, mang ý nghĩa là vị trí trên chuỗi (, ). Từ tập này có thể xác định một tập chuỗi liên kết với . Sự tiềm năng của một lời giải được đánh giá dựa trên một hàm chấm điểm, trong đó nó sẽ đánh giá độ tương đồng giữa các chuỗi con trong lời giải tìm được.

Hàm chấm điểm là một phần quan trọng của việc tìm lời giải tiềm năng. Một hàm chấm điểm không tốt sẽ dẫn đến một kết quả không tốt, cho dù thuật toán tối ưu hàm đó có hiệu quả đến đâu. Đã có nhiều nghiên cứu đề cập tới các hàm chấm điểm đưa ra lời giải khá tốt, và một số hàm thông dụng sẽ được trình bày sau đây.

## Một số hàm mục tiêu thông dụng

Hai hàm mục tiêu sau đây đều hướng đến việc chấm điểm độ giống nhau giữa các chuỗi liên kết. Nói cách khác, các thuật toán sử dụng chúng sẽ hướng tới việc tìm ra tập chuỗi liên kết giống nhau tối đa.

### Hamming Score

Khoảng cách Hamming (Hamming Distance) giữa hai xâu và cùng độ dài được định nghĩa là số lượng các ký tự ở vị trí tương ứng với nhau nhưng có giá trị khác nhau, ký hiệu là .

Hamming Score (ký hiệu là ) của một tập chuỗi liên kết đề xuất được tính bằng tổng khoảng cách Hamming của từng chuỗi trong M với một chuỗi mẫu (pattern) độ dài xác định từ chính tập .

Hàm số trả về giá trị càng nhỏ thì các chuỗi trong tập M càng giống nhau càng, và được coi là càng tiềm năng để xác định motif.

### Information Content

Điểm Information Content (ký hiệu là ) của một tập chuỗi liên kết được tính như sau:

Trong đó, là tần suất ký tự thứ là chữ cái u trên các , là tần suất nền (background frequency) của chữ cái xuất hiện trong toàn thể bộ dữ liệu.

Bản chất của IC chính là tổng của entropy tương đối trên từng vị trí tương ứng giữa các chuỗi liên kết của motif, tương phản với tần suất nền của bộ dữ liệu. Entropy tương đối của một vị trí cực tiểu khi tần suất của các chữ cái bằng tần suất nền, cực đại khi vị trí này bằng nhau ở mọi chuỗi trong tập. Do đó, khác với hàm Hamming Score, điểm IC càng cao thì tập chuỗi liên kết được đề xuất càng giống nhau và càng tiềm năng.

Hàm này được sử dụng trong nhiều nghiên cứu về tìm kiếm motif bởi độ tương quan cao trong việc mô phỏng đặc trưng motif. Bởi vậy, trong báo cáo này chúng tôi đề xuất sử dụng tiếp cận tối ưu đàn kiến với hàm mục tiêu IC thay vì Hamming Score như ACOMotif.

# Phương pháp ACO

## Tổng quan về phương pháp ACO

ACO được đề xuất bởi Dorigo et al. (1996) là một phương pháp metaheuristic để giải các bài toán tối ưu tổ hợp khó, ví dụ như Bài toán Người giao hàng (Traveling Salesman Problem) hay Bài toán Quy hoạch Toàn phương Nhị phân Không ràng buộc (Unconstrained Binary Quadratic Programming). Các thuật toán sử dụng phương pháp ACO để tìm motif thường tuân theo lược đồ Memetic, một phương pháp tìm kiếm dựa trên quần thể, đề xuất lần đầu bởi Moscato (1989) và được áp dụng cho thuật toán di truyền.

Dưới đây là phương pháp chung cho các thuật toán tìm motif sử dụng ACO với các kĩ thuật tìm kiếm cục bộ tuân theo một lược đồ memetic đơn giản. Bài toán gốc sẽ được đưa về bài toán tìm lời giải tối ưu trên một đồ thị cấu trúc trong đó V là tập đỉnh, E là tập cạnh, là tập ràng buộc, là tập thông tin heuristic và là tập vết mùi (có thể được đặt trên đỉnh hoặc cạnh). Một lời giải chấp nhận được là một đường đi thỏa mãn tập ràng buộc , bắt đầu từ một đỉnh trong tập con của V, và được mở rộng nhờ thủ tục bước đi ngẫu nhiên trên đồ thị cấu trúc với xác suất lựa chọn tỉ lệ thuận với thông tin heuristic và vết mùi.

Thuật toán này thực hiện nhiều thế hệ kiến gồm con kiến nhân tạo, và chúng sẽ sinh ra các lời giải dựa vào thông tin heuristic và vết mùi. Với mỗi tập lời giải được sinh ra bởi một thế hệ kiến, một tập con sẽ được chọn để áp dụng kĩ thuật tìm kiếm cục bộ (local search), trước khi chọn ra lời giải tốt nhất của thế hệ đó. Những lời giải tốt nhất của các thế hệ sẽ được bổ sung vào tập kết quả mong muốn cuối cùng. Bản thân cả quá trình này cũng được thực hiện nhiều lần với các vết mùi được khởi tạo lại.

**procedure** Memetic-ACO Algorithm

Khởi tạo kiến và ma trận vết mùi

**while** Điều kiện dừng chưa thỏa mãn **do**

Sinh ra lời giải cho *A* con kiến trong thế hệ

Chọn một tập con từ *A* lời giải của thế hệ

**for** từng lời giải trong tập con **do**

Thực hiện enhanced/local search cho lời giải

**end for**

Cập nhật vết mùi

**end while**

Đưa ra lời giải

**end procedure**

***Hình 1.*** *Mã giả của thuật toán Tối ưu đàn kiến*

## Phân tích thuật toán ACOMotif dùng tiếp cận ACO giải bài toán tìm motif

Thuật toán ACOMotif (Hoang, 2015) đặt bài toán trên giả thiết một chuỗi DNA đầu vào chỉ có đúng một vị trí liên kết, tức là luôn bằng **.** Tận dụng đặc tính của hàm mục tiêu Hamming Score, ACOMotif sử dụng mô hình tối ưu đàn kiến nói trên trên một đồ thị cấu trúc kích thước ( đối với chuỗi DNA) nhằm tìm ra chuỗi mẫu . Sau đó ở mỗi xâu sẽ chọn một chuỗi con có khoảng cách Hamming () nhỏ nhất với chuỗi mẫu này, từ đó tìm tạo được tổng các khoảng cách Hamming tối thiểu. Các quá trình tìm kiếm cục bộ để cải tiến lời giải chủ yếu xoay quanh việc thay thế các ký tự trên chuỗi mẫu này. Điểm mạnh của thuật toán này là chạy khá nhanh nhờ đồ thị cấu trúc kích thước nhỏ.

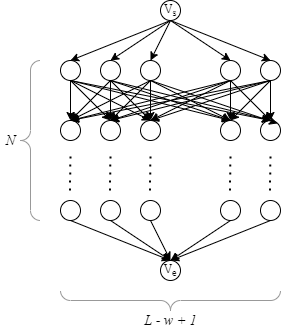
Về cơ bản, các hàm mục tiêu được để xuất cho đến nay, bao gồm cả Hamming Score và IC đều chưa mô phỏng chính xác được đặc trưng xác định motif và chuỗi liên kết. Thuật toán ACOMotif tiếp cận từ việc xây dựng cấu hình chuỗi mẫu, nên chỉ có thể tích hợp với hàm mục tiêu Hamming Score. Đồ thị cấu trúc này khó tích hợp với các hàm chấm điểm khác kiểu Hamming.

# Phương pháp đề xuất

Chúng tôi đề xuất thuật toán ACOSite sử dụng tiếp cận tối ưu đàn kiến trong đó xây dựng trực tiếp cấu hình vị trí liên kết rồi tính điểm. ACOSite chấm điểm độ giống/khác của một tập chuỗi liên kết đề xuất bằng hàm IC. Thuật toán sẽ đặt bài toán trên giả thiết mỗi chuỗi DNA đầu vào sẽ có tối thiểu một vị trí liên kết. Ta sẽ bắt đầu với việc sử dụng thuật toán ACO để tìm ra một tập vị trí liên kết có số lượng đúng bằng N, tức là một vị trí trên mỗi chuỗi đầu vào sao cho hàm mục tiêu của tập này đạt cực đại. Sau đó, phương pháp tìm kiếm cục bộ giúp thay thế các vị trí đã chọn sao có điểm IC tối ưu hơn và tìm thêm các vị trí mới nếu có thể mà không khiến điểm IC này giảm đi. Mô tả chi tiết về các thành phần của thuật toán ACOSite sẽ được trình bày sau đây.

## Đồ thị cấu trúc

Đồ thị cấu trúc sẽ gồm một đỉnh xuất phát , một đỉnh kết thúc và một mạng hàng, mỗi hàng gồm đỉnh. Mạng này tương ứng với việc trên mỗi chuỗi của chuỗi DNA đầu vào độ dài , ta sẽ lựa chọn một vị trí để bắt đầu chuỗi liên kết độ dài . Đỉnh xuất phát sẽ có cạnh nối tới các đỉnh thuộc hàng 1 của mạng. Với , từng đỉnh trên hàng thứ của mạng sẽ có bộ cạnh nối đầy đủ sang từng đỉnh trên hàng thứ . Cuối cùng, từ các đỉnh của hàng thứ sẽ nối về đỉnh kết thúc . Do đồ thị có kích thước khá lớn nên vết mùi sẽ không được lưu trên cạnh mà được đặt trên đỉnh.

****

***Hình 2.*** *Mô hình đồ thị cấu trúc*

## Thông tin heuristic

Thông tin heuristic được tính trực tiếp trong quá trình tìm kiếm đường đi. Khi đến hàng thứ trên mạng, ta đã xác định được tập gồm *i* vị trí liên kết ứng viên và cần lựa chọn vị trí thuộc hàng tiếp theo. Thông tin heuristic của mỗi đỉnh của hàng sẽ được tính trực tiếp bằng điểm IC của tập hiện tại thêm . Cách tính này sẽ trực tiếp cho thấy mức độ ảnh hưởng của mỗi đỉnh đó tới điểm IC đã tích lũy được.

## Vết mùi và quy tắc cập nhật

Thông tin về vết mùi được đặt trên từng đỉnh của đồ thị cấu trúc, và được cập nhật sau khi một thế hệ kiến tìm kiếm xong lời giải.

### Quy tắc SMMAS

Trước tiên, mùi ở tất cả các đỉnh đều giảm đi một lượng theo tỉ lệ (), sau đó, nếu đỉnh nằm trong lời giải tốt nhất của thế hệ sẽ được bù lại vết mùi để tiến gần với một giá trị , nếu không thì xuống gần với . Đây là công thức cập nhật mùi SMMAS (Smooth Max-Min Ant System) (Huan, Linh-Trung, et al. 2012).

Ở mỗi thế hệ, các con kiến sẽ tìm được một đường đi dạng . Trong nhiều lời giải đó, ghi nhận lời giải đạt được điểm IC cao. Công thức cập nhật mùi tuân theo quy tắc sau:

nếu

nếu

Ban đầu, tất cả vết mùi đều được khởi tạo bằng .

### Quy tắc thử nghiệm

Ngoài ra, chúng tôi cũng thử nghiệm một tùy chọn công thức cập nhật mùi mới như sau: sau khi giảm mùi theo tỉ lệ , bù lại phụ thuộc vào khoảng cách với lời giải tốt nhất theo tỉ lệ , điều này dựa trên quan sát rằng nếu tịnh tiến một tập vị trí liên kết tốt đi một vài đơn vị thì có thể đạt được tập vị trí khác tương đối tiềm năng:

c | Qi – j |

Ban đầu, tất cả vết mùi đều được khởi tạo bằng .

## Thủ tục bước đi ngẫu nhiên để xây dựng lời giải

Con kiến bước đi tuần tự từ hàng đầu tiên đến hàng . Giả sử kiến đang ở hàng , khi đó con kiến sẽ bước đi ngẫu nhiên đến một đỉnh thuộc hàng . Xác suất lựa chọn của các đỉnh thuộc hàng sẽ tỉ lệ thuận với vết mùi và thông tin heuristic của chúng.

Trong đó, là xác suất lựa chọn, là vết mùi, là thông tin heuristic của đỉnh trên hàng thứ của mạng.

## Tìm kiếm địa phương

Đối với đường đi của kiến và tập vị trí liên kết hiện tại , chúng tôi đề xuất 4 cách để tiến sang các cấu hình lân cận của đáp số và tối ưu hơn hàm mục tiêu IC như sau:

1. Thay thế bằng các giá trị từ tới .
2. Bỏ đi một vị trí đã có và thỏa mãn điều kiện (\*).
3. Thêm một vị trí mới và thỏa mãn điều kiện (\*).
4. Thêm một vị trí mới và xóa những vị trí làm vi phạm điều kiện (\*).

Thủ tục (a) được sử dụng với kết quả tốt nhất trong mỗi thế hệ trong khi thủ tục (b), (c), (d) sử dụng với kết quả cuối cùng khi đã chạy hết tất cả các thế hệ.

(\*) Mỗi chuỗi phải có ít nhất một vị trí liên kết, mọi cặp mà (thì khoảng cách giữa và phải không được nhỏ hơn (các vị trí liên kết trên cùng một chuỗi không vướng vào nhau).

# Thực nghiệm và Kết quả

## Cài đặt

Chúng tôi đã cài đặt thuật toán ACOSite thành một chương trình viết bằng C++ với hai tùy chọn:

* ops (one per sequence): Chỉ tìm một vị trí liên kết duy nhất trên mỗi chuỗi.
* mps (multiple per sequence): Có thể tìm nhiều vị trí liên kết trên mỗi chuỗi.

Hai chương trình này nhận đầu vào các chuỗi DNA đọc từ tệp định dạng FASTA (phần mở rộng .fa, định dạng tệp tiêu chuẩn để lưu trữ chuỗi DNA) và W - độ dài motif (đọc từ màn hình). Đầu ra của chúng là vị trí liên kết của motif trên các chuỗi.

Một số tham số trong thuật toán:

* , ,
* Số lượng kiến là 10.
* Điều kiện dừng của thuật toán là kết quả không tăng sau 20 thế hệ.

## Dữ liệu

Dữ liệu chúng tôi sử dụng là các bộ dữ liệu gồm nhiều chuỗi DNA thực đã được thu thập trong (Garbelini, 2018) và đã xác định được các vị trí liên kết thông qua thí nghiệm sinh học. Mỗi bộ dữ liệu sẽ gồm hai tệp, một lưu các chuỗi DNA và một lưu danh sách vị trí liên kết trên từng chuỗi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã hiệu | Nguồn | N | L | W |
| CRP | Stormo and Hartzell | 18 | 105 | 22 |
| CREB | ABS | 17 | 501 | 12 |
| HNF1 | ABS | 22 | 501 | 14 |
| MEF2 | ABS | 17 | 501 | 10 |
| MYOD | ABS | 17 | 501 | 6 |
| NFKB | ABS | 6 | 501 | 12 |
| SRF | ABS | 20 | 501 | 12 |
| TBP | ABS | 95 | 501 | 7 |
| MCB | SCPD | 6 | 550 | 6 |
| PDR3 | SCPD | 7 | 550 | 8 |
| REB1 | SCPD | 15 | 105 | 10 |
| **GATA2** | **JASPAR** | **4380** | **114** | **14** |
| **GATA3** | **JASPAR** | **4628** | **108** | **8** |
| **IRF1** | **JASPAR** | **1362** | **121** | **21** |
| **NFE2L2** | **JASPAR** | **726** | **115** | **15** |
| **TFAP2A** | **JASPAR** | **5098** | **115** | **15** |

***Bảng 1.*** *Thông tin các bộ dữ liệu*

Trong danh sách trên, chúng ta có thể chia thành hai tập là tập các bộ dữ liệu lớn (5 bộ JASPAR) và tập các bộ dữ liệu nhỏ (các bộ còn lại).

## Thước đo

Chúng tôi sử dụng cách đánh giá độ chính xác của tập vị trí liên kết như đề xuất trong (Chan, 2008). Theo đó, nếu một vị trí dự đoán trong đáp số chênh lệch với vị trí thật không quá delta=3 đơn vị thì nó được coi là dự đoán đúng. Chúng tôi tính Precision, Recall và F1 theo tiêu chí này.

## Kết quả

### Đối chiếu với thuật toán ACOMotif

Do thuật toán ACOMotif chỉ tìm một vị trí liên kết trên mỗi chuỗi đầu vào, do đó chúng tối sẽ chỉ so sánh với tùy chọn ops của ACOSite.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ACOMotif | | | ACOSite (ops) | | |
| Precision | Recall | F1 | Precision | Recall | F1 |
| CREB | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| CRP | 0.5417 | 0.7222 | 0.6190 | 0.7083 | 0.9444 | **0.8095** |
| HNF1 | 0.2222 | 0.2400 | **0.2308** | 0.0000 | 0.0000 | 0.0000 |
| MCB | 0.0000 | 0.0000 | 0.0000 | 0.5000 | 1.0000 | **0.6667** |
| MEF2 | 0.0000 | 0.0000 | 0.0000 | 0.8235 | 0.8235 | **0.8235** |
| MOYD | 0.0476 | 0.0333 | **0.0392** | 0.0000 | 0.0000 | 0.0000 |
| NFKB | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| PDR3 | 0.5556 | 1.0000 | **0.7143** | 0.3889 | 1.0000 | 0.5600 |
| REB1 | 0.4500 | 0.6000 | 0.5143 | 0.4500 | 0.6000 | 0.5143 |
| SRF | 0.2778 | 0.4545 | 0.3448 | 0.3333 | 0.6000 | **0.4286** |
| TBP | 0.0000 | 0.0000 | 0.0000 | 0.7579 | 0.7579 | **0.7579** |
| Trung bình | 0.1904 | 0.2773 | 0.2239 | 0.3602 | 0.5205 | **0.4146** |

***Bảng 2.*** *Đối chiếu giữa ACOMotif và tùy chọn ops của ACOSite  
(Các điểm F1 tốt hơn được in đậm)*

Hamming Score đã thể hiện điểm F1 tốt hơn ở một số bộ dữ liệu nhưng tổng quan và tính trung bình thì kém hơn IC. Bởi vậy việc đồ thị cấu trúc của ACOMotif bị phụ thuộc vào Hamming Score chúng tôi cho là một hạn chế rất lớn.

### Đối chiếu hiệu quả trong việc tối ưu hàm mục tiêu IC

Để kiểm định hiệu quả của thuật toán ACOSite tùy chọn ops, chúng tôi sẽ so sánh thuật toán của mình với thuật toán MEME, đề xuất bởi Timothy L. Bailey và Charles Elkan năm 1994. MEME là một trong số ít nghiên cứu công khai cung cấp mã nguồn và có các cấu hình lựa chọn tương đương với các đặc tính hiện tại của chương trình ACOSite, giúp cho quá trình đối chiếu đạt được sự công bằng nhất có thể. Cụ thể là:

* Hàm mục tiêu IC: MEME có tổng cộng 6 hàm mục tiêu để người sử dụng có thể lựa chọn. Trong đó cấu hình mặc định là hàm E-value của IC.
* MEME có tùy chọn tìm kiếm một vị trí liên kết trên mỗi chuỗi: Thuật toán ACOSite của chúng tôi hoạt động trên giả thiết là mỗi chuỗi phải chứa ít nhất một vị trí liên kết. Điều này thỏa mãn trên mọi dữ liệu thực nghiệm chúng tôi sử dụng nhưng về mặt lý thuyết có thể có chuỗi không tồn tại vị trí nào. Mặt khác, MEME có 3 tùy chọn tìm kiếm: oops (một vị trí mỗi chuỗi), zoops (không hoặc một vị trí mỗi chuỗi) và anr (số lượng vị trí bất kỳ). Bởi vậy, tùy chọn ops của ACOSite và chế độ oops của MEME là tương đương.

Chúng tôi tiến hành thực nghiệm để quan sát phương diện tối ưu hàm mục tiêu IC:

* Giữa tùy chọn ops của ACOSite và tùy chọn oops của MEME.
* Giữa tùy chọn mps và tùy chọn ops của ACOSite.

Mỗi chương trình được chạy 5 lần để lấy ra kết quả tốt nhất.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ACOSite (ops) | | | | MEME (oops) | | | |
| Precision | Recall | F1 | IC | Precision | Recall | F1 | IC |
| CREB | 0.0000 | 0.0000 | 0.0000 | **14.6061** | 0.1579 | 0.1765 | 0.1667 | 14.0110 |
| CRP | 0.7083 | 0.9444 | 0.8095 | **14.6456** | 0.7083 | 0.9444 | 0.8095 | 14.5746 |
| HNF1 | 0.0000 | 0.0000 | 0.0000 | **13.9204** | 0.2222 | 0.2727 | 0.2449 | 13.1703 |
| MCB | 0.5000 | 1.0000 | 0.6667 | 12.9893 | 0.5000 | 1.0000 | 0.6667 | 12.9893 |
| MEF2 | 0.8235 | 0.8235 | 0.8235 | **15.5390** | 0.6471 | 0.6471 | 0.6471 | 15.4046 |
| MOYD | 0.0000 | 0.0000 | 0.0000 | **10.6023** | 0.0000 | 0.0000 | 0.0000 | 10.4939 |
| NFKB | 0.0000 | 0.0000 | 0.0000 | 20.2940 | 0.0000 | 0.0000 | 0.0000 | 20.2940 |
| PDR3 | 0.3889 | 1.0000 | 0.5600 | 16.6012 | 0.3889 | 1.0000 | 0.5600 | 16.6012 |
| REB1 | 0.4500 | 0.6000 | 0.5143 | **10.7789** | 0.4500 | 0.6000 | 0.5143 | 10.4571 |
| SRF | 0.3333 | 0.6000 | 0.4286 | **16.7978** | 0.3611 | 0.6500 | 0.4643 | 16.5438 |
| TBP | 0.7579 | 0.7579 | 0.7579 | **10.7193** | 0.7368 | 0.7368 | 0.7368 | 10.6456 |
| GATA2 | 0.9838 | 0.9838 | 0.9838 | **12.3925** | 0.9829 | 0.9829 | 0.9829 | 12.3284 |
| GATA3 | 0.9756 | 0.9756 | 0.9756 | 12.2279 | 0.9749 | 0.9749 | 0.9749 | 12.2279 |
| IRF1 | 0.9170 | 0.9170 | 0.9170 | **15.4477** | 0.9112 | 0.9112 | 0.9112 | 15.4465 |
| NFE2L2 | 0.8898 | 0.8898 | 0.8898 | **16.0619** | 0.8829 | 0.8829 | 0.8829 | 16.0616 |
| TFAP2A | 0.9719 | 0.9719 | 0.9719 | **12.3871** | 0.9708 | 0.9708 | 0.9708 | 12.3850 |

***Bảng 2.*** *Đối chiếu giữa tùy chọn ops của ACOSite với chế độ oops của MEME  
(Các điểm IC cao hơn được in đậm)*

Ta có thể thấy thuật toán ACOSite (ops) luôn cho điểm IC tốt hơn hoặc bằng thuật toán MEME (oops), cụ thể là 4 trường hợp bằng nhau và 12 trường hợp còn lại là ACOSite tính được cao hơn.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ACOSite (ops) | | | | ACOSite (mps) | | | |
| Precision | Recall | F1 | IC | Precision | Recall | F1 | IC |
| CREB | 0.0000 | 0.0000 | 0.0000 | 14.6061 | 0.0000 | 0.0000 | 0.0000 | 14.6061 |
| CRP | 0.7083 | 0.9444 | 0.8095 | 14.6456 | 0.7083 | 0.9444 | 0.8095 | 14.6456 |
| HNF1 | 0.0000 | 0.0000 | 0.0000 | 13.9204 | 0.0000 | 0.0000 | 0.0000 | **13.9835** |
| MCB | 0.5000 | 1.0000 | 0.6667 | 12.9893 | 0.5000 | 1.0000 | 0.6667 | 12.9893 |
| MEF2 | 0.8235 | 0.8235 | 0.8235 | 15.5390 | 0.8235 | 0.8235 | 0.8235 | 15.5390 |
| MOYD | 0.0000 | 0.0000 | 0.0000 | 10.6023 | 0.0952 | 0.0769 | 0.0851 | **10.7770** |
| NFKB | 0.0000 | 0.0000 | 0.0000 | 20.2940 | 0.0000 | 0.0000 | 0.0000 | 20.2940 |
| PDR3 | 0.3889 | 1.0000 | 0.5600 | 16.6012 | 0.5556 | 1.0000 | 0.7143 | **16.7428** |
| REB1 | 0.4500 | 0.6000 | 0.5143 | 10.7789 | 0.4500 | 0.6000 | 0.5143 | 10.7789 |
| SRF | 0.3333 | 0.6000 | 0.4286 | 16.7978 | 0.3333 | 0.6000 | 0.4286 | 16.7978 |
| TBP | 0.7579 | 0.7579 | 0.7579 | 10.7193 | 0.7895 | 0.7282 | 0.7576 | **10.8147** |
| GATA2 | 0.9838 | 0.9838 | 0.9838 | 12.3925 | 0.9879 | 0.9798 | 0.9839 | **12.4005** |
| GATA3 | 0.9756 | 0.9756 | 0.9756 | 12.2279 | 0.9860 | 0.9657 | 0.9757 | **12.2382** |
| IRF1 | 0.9170 | 0.9170 | 0.9170 | 15.4477 | 0.9200 | 0.9067 | 0.9133 | **15.4759** |
| NFE2L2 | 0.8898 | 0.8898 | 0.8898 | 16.0619 | 0.9380 | 0.8335 | 0.8827 | **16.2864** |
| TFAP2A | 0.9719 | 0.9719 | 0.9719 | 12.3871 | 0.9743 | 0.9699 | 0.9721 | **12.3911** |

***Bảng 3.*** *Đối chiếu giữa tùy chọn ops của ACOSite với tùy chọn mps của ACOSite  
(Các điểm số IC cao nhất được in đậm)*

Tương tự bảng bên trên, tùy chọn cải tiến mps của ACOSite lại cũng luôn cho điểm IC tốt hơn hoặc tùy chọn ops, cụ thể là 4 trường hợp bằng nhau và 12 trường hợp còn lại là ACOSite tính được cao hơn.

# Bình luận và Kết luận

Quan sát cả hai bảng F1, ta thấy IC ở một số bộ dữ liệu cho thấy sự tương quan tốt với F1. Bằng chứng là nhiều đáp án đạt điểm F1 từ 0.75 đến 0.98, điểm IC tăng thì cũng điểm F1 cũng tăng. Nhưng không phải lúc nào cũng vậy, ta thấy cũng có những trường hợp mà cả MEME và ACOSite đều cho F1 bằng 0. Chúng tôi đã khảo sát những bộ dữ liệu đó và thấy rằng điểm IC của motif thực thấp hơn nhiều những giá trị mà chúng tôi tìm được.

Ngoài ra, có một điểm yếu nữa ở IC đó là nó không trực tiếp chịu ảnh hưởng của số lượng chuỗi liên kết (), bởi vậy khi chọn càng ít chuỗi, độ sai khác càng giảm đi. Nếu chỉ có duy nhất một chuỗi thì IC đạt cực đại. Đây chính là lí do mà thuật toán của chúng tôi đặt giả thiết mỗi chuỗi DNA đầu vào phải chứa tổi thiểu một vị trí liên kết.

Báo cáo này trình bày thuật toán ACOSite sử dụng tiếp cận tối ưu đàn kiến trong đó xây dựng motif theo cấu hình vị trí liên kết rồi tính điểm bằng hàm IC cùng với các đề xuất tương ứng về đồ thị cấu trúc mới, quy tắc cập nhật mùi mới, thủ tục tìm kiếm cục bộ mới để phù hợp với biểu diễn motif theo vị trí. Kết quả thực nghiệm cho thấy ACOSite đã giúp cực đại một cách hiệu quả hàm IC. Điểm mạnh của ACOSite là nó không nhất thiết phải sử dụng hàm chấm điểm IC mà có thể tích hợp với nhiều loại hàm chấm điểm khác, theo vị trí hoặc theo chuỗi đồng thuận. Do đó, khảo sát những hàm mục tiêu phù hợp hơn với dữ liệu thực là một trong những hướng dự kiến của chúng tôi để phát triển ACOSite.

# Tài liệu tham khảo

Bailey, T., Williams, N., M., & Li, W. (2006). MEME: Discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res,* *34*, 369-373.

Bouamama, S., Boukerram, A., & Al-Badarneh, A. (2010). Motif finding using ant colony optimization. *LNCS,* *6234*, 464-471.

Caldonazzo Garbelini, J.M., Kashiwabara, A.Y. & Sanches, D.S. Sequence motif finder using memetic algorithm. *BMC Bioinformatics* **19,**4 (2018). <https://doi.org/10.1186/s12859-017-2005-1>

Chan, T. M., Leung, K. S., & Lee, K. H. (2008). TFBS identification based on genetic algorithm with combined representations and adaptive post-processing. *Bioinformatics*, *24*(3), 341-349.

Dorigo, M., Maniezzo, V., & Golomi, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics,* *26*(1), 29-41.

Hertz, G., & Stormo, G. (1999). Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics,* *15*(7), 563-577.

Hoang, H. X., Duong, T. A., & Nguyen, H. T. (2015). An Efficient Ant Colony Algorithm for DNA Motif Finding. *Advances in Intelligent Systems and Computing,* *326*, 589-601. doi:doi.org/10.1007/978-3-319-11680-8\_47

Huan HX, Linh-Trung N, Dong DD, Huu-Tue H. 2012. Solving the traveling salesman problem with ant colony optimization: a revisit and new efficient algorithms. REV J. Electron. Commun. 2:121–129.

Li, M., Ma, B., & Wang, L. (1999). Finding similar regions in many strings. *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing,* 473-482.