|  |  |  |
| --- | --- | --- |
|  |  | |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  **---------------------------------------** | | |
|  | | |
|  | | |
| **Hoàng Văn Thành** | | |
|  | | |
| **XÁC ĐỊNH ĐOẠN ĐIỀU HÒA GEN TRÊN TRÌNH TỰ ADN BẰNG PHƯƠNG PHÁP TÍNH TOÁN** | | |
|  | | |
| **LUẬN VĂN THẠC SĨ KỸ THUẬT** | | |
| HÀ NỘI – 2013 | | |
|  | |  | |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  **---------------------------------------** | | | |
|  | | | |
|  | | | |
| **Hoàng Văn Thành** | | | |
|  | | | |
|  | | | |
| **XÁC ĐỊNH ĐOẠN ĐIỀU HÒA GEN TRÊN TRÌNH TỰ ADN BẰNG PHƯƠNG PHÁP TÍNH TOÁN** | | | |
|  | | | |
| **Chuyên ngành: Khoa học máy tính** | | | |
| **Mã số: 60.48.01**  **LUẬN VĂN THẠC SĨ KỸ THUẬT**  NGƯỜI HƯỚNG DẪN KHOA HỌC  PGS.TS TỪ MINH PHƯƠNG  HÀ NỘI - 2013 | | | |
|  | | | |

# LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả luận văn

**MỤC LỤC**

[LỜI CAM ĐOAN i](#_Toc366053528)

**[MỤC LỤC](#_Toc366053529)** [ii](#_Toc366053529)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT iv](#_Toc366053530)

[DANH MỤC CÁC BẢNG v](#_Toc366053531)

[DANH MỤC CÁC HÌNH vi](#_Toc366053532)

[MỞ ĐẦU 1](#_Toc366053533)

[CHƯƠNG 1: CÁC KHÁI NIỆM VÀ VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN XÁC ĐỊNH VÙNG TĂNG CƯỜNG 4](#_Toc366053534)

**[1.1.](#_Toc366053535)****[Các khái niệm](#_Toc366053535)** [4](#_Toc366053535)

**[1.1.1.](#_Toc366053536)****[Sự hoạt động của gen](#_Toc366053536)** [4](#_Toc366053536)

**[1.1.2.](#_Toc366053537)****[Điều hòa sự hoạt động của gen](#_Toc366053537)** [7](#_Toc366053537)

**[1.1.3.](#_Toc366053538)****[Nhân tố phiên mã (transcription factor)](#_Toc366053538)** [8](#_Toc366053538)

**[1.1.4.](#_Toc366053539)****[Vùng tăng cường (enhancer)](#_Toc366053539)** [11](#_Toc366053539)

**[1.2.](#_Toc366053540)****[Các nhóm phương pháp xác định vùng tăng cường](#_Toc366053540)** [13](#_Toc366053540)

**[1.3.](#_Toc366053541)****[Kết luận chương](#_Toc366053541)** [15](#_Toc366053541)

[CHƯƠNG 2: PHƯƠNG PHÁP XÁC ĐỊNH VÙNG TĂNG CƯỜNG DỰA TRÊN SVM 17](#_Toc366053542)

**[2.1.](#_Toc366053543)****[Phương pháp xác định vùng tăng cường dựa trên SVM](#_Toc366053543)** [17](#_Toc366053543)

**[2.1.1.](#_Toc366053544)****[Giới thiệu về SVM](#_Toc366053544)** [17](#_Toc366053544)

**[2.1.2.](#_Toc366053545)****[Giới thiệu về hàm kernel trong SVM](#_Toc366053545)** [19](#_Toc366053545)

**[2.1.3.](#_Toc366053546)****[Phương pháp xác định vùng tăng cường dựa trên SVM](#_Toc366053546)** [20](#_Toc366053546)

**[2.1.4.](#_Toc366053547)****[Một số hàm kernel đã được sử dụng để giải quyết bài toán](#_Toc366053547)** [21](#_Toc366053547)

**[2.2.](#_Toc366053548)****[Các hàm kernel được đề xuất](#_Toc366053548)** [22](#_Toc366053548)

**[2.2.1.](#_Toc366053549)****[Hàm kernel dựa trên entropy](#_Toc366053549)** [22](#_Toc366053549)

**[2.2.2.](#_Toc366053550)****[Hàm kernel phân cấp (hierarchical kernel)](#_Toc366053550)** [24](#_Toc366053550)

[CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 27](#_Toc366053551)

**[3.1.](#_Toc366053552)****[Cài đặt phương pháp](#_Toc366053552)** [27](#_Toc366053552)

**[3.1.1.](#_Toc366053553)****[Xây dựng bộ dữ liệu mẫu](#_Toc366053553)** [27](#_Toc366053553)

**[3.1.2.](#_Toc366053554)****[Cài đặt phương pháp](#_Toc366053554)** [28](#_Toc366053554)

**[3.2.](#_Toc366053555)****[Tiêu chí đánh giá](#_Toc366053555)** [29](#_Toc366053555)

**[3.3.](#_Toc366053556)****[Đánh giá kết quả](#_Toc366053556)** [31](#_Toc366053556)

**[3.4.](#_Toc366053557)****[So sánh kết quả với các phương pháp khác](#_Toc366053557)** [34](#_Toc366053557)

**[3.5.](#_Toc366053558)****[Nhận xét về kết quả và hướng phát triển.](#_Toc366053558)** [35](#_Toc366053558)

[KẾT LUẬN 36](#_Toc366053559)

[TÀI LIỆU THAM KHẢO 37](#_Toc366053560)

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Tiếng Anh** | **Tiếng Việt** |
| ADN | Acid Deoxyribo Nucleic | Axit Deoxyribo Nucleic |
| ARN | Acid ribonucleic | Axit ribonucleic |
| AUC | Area under the ROC curve | Diện tích của vùng nằm dưới đường cong ROC |
| ChIP | Chromatin immunoprecipitation | Phương pháp xác định vùng ADN của một tương tác protein - ADN cụ thể |
| ChIP-chip | ChIP-on-chip | kỹ thuật kết hợp ChIP với DNA microarray |
| ChIP-seq | ChIP-sequencing | Kỹ thuật kết hợp ChIP với các chuỗi ADN song song lớn |
| CRM | CIS-regulatory module | Module điều hòa ở trên cùng một phía |
| MACS | Model-based analysis of ChIP-Seq | Phương pháp phân tích ChIP-seq dựa vào mô hình |
| mARN | MessengerARN | ARN thông tin |
| rARN | ARN riboxom | ARN riboxom |
| ROC | Receiver Operating Characteristics | Đường cong đặc trưng hoạt động của bộ thu nhận |
| SVM | Support vector machine | Máy vector hỗ trợ quyết định |
| tARN | TransportARN | ARN vận chuyển |
| TF | Transcription factor | Nhân tố phiên mã |
| TFBS | Transcription factor binding site | Vùng liên kết nhân tố phiên mã |

# DANH MỤC CÁC BẢNG

**[Bảng 3.1. Giá trị AUC () đối với hàm kernel phân cấp ứng với số lượng các phần khác nhau và giữa các phần không chồng lên nhau. Giá trị AUC cao nhất ứng với mỗi TF được bôi đậm](#_Toc366053561)** [32](#_Toc366053561)

**[Bảng 3.2. Bảng giá trị AUC () đối với hàm kernel dựa trên entropy ứng với số lượng các phần khác nhau và giữa các phần không chồng lên nhau.](#_Toc366053562)** [32](#_Toc366053562)

**[Bảng 3.3. Bảng giá trị AUC () đối với hàm kernel phân cấp ứng với số lượng phần con là 7 và các độ dài vùng chồng lên nhau khác nhau.](#_Toc366053563)** [33](#_Toc366053563)

**[Bảng 3.4. Bảng giá trị AUC () đối với hàm kernel phân cấp ứng với số lượng phần là 7 và phương pháp đưa ra bởi Lee et al.](#_Toc366053564)****[[6].](#_Toc366053564)** [34](#_Toc366053564)

# DANH MỤC CÁC HÌNH

**[Hình 1.1. Sơ đồ khái quát sự hoạt động của gen ở sinh vật nhân sơ (a) và nhân thực (b)](#_Toc366053565)** [5](#_Toc366053565)

**[Hình 1.2. Sơ đồ khái quát quá trình phiên mã](#_Toc366053566)** [6](#_Toc366053566)

**[Hình 1.3. Phân nhóm các phương pháp xác định vùng tăng cường.](#_Toc366053567)** [14](#_Toc366053567)

**[Hình 2.1. Hình minh họa cho hàm kernel dựa trên entropy (a) và hàm kernel phân cấp (b)](#_Toc366053568)** [23](#_Toc366053568)

# MỞ ĐẦU

Trong khuôn khổ luận văn, em tập trung nghiên cứu phương pháp xác định vùng tăng cường (enhancer) từ trình tự ADN của động vật bậc cao, sử dụng phương pháp học máy vector tựa SVM (support vector machines). Em đã nghiên cứu một số phương pháp mới để xây dựng hàm kernel dùng để biểu diễn trình tự ADN khi sử dụng với SVM và thử nghiệm trên một số bộ dữ liệu thực. Phương pháp đề xuất có kết quả tốt, cho phép cải thiện độ chính xác so với phương pháp đã có. *Các kết quả được công bố trong bài báo “Enhancer prediction using distance aware kernels”. Bài báo đã được chấp nhận đăng tại kỷ yếu hội nghị RIVF 2013 sẽ diễn ra vào tháng 11/2013*.

Trải qua nhiều thập kỉ, khoa học máy tính đã có những bước phát triển vượt bậc. Ngày nay, khoa học máy tính đã khám phá ra những khả năng mới trong một lĩnh vực đầy hứa hẹn như tin sinh học. Sinh học và những nguyên lý liên quan của nó là một vùng đất màu mỡ để các nhà nghiên cứu tiếp tục đào sâu tìm hiểu. Tin sinh học hay sinh học tính toán liên quan đến việc sử dụng các kỹ thuật từ toán học ứng dụng, tin học, thống kê… để giải quyết các vấn đề sinh học. Một số mảng nghiên cứu chính đang được phát triển như là phân tích chuỗi, phân tích biểu diễn gen, dự đoán cấu trúc ADN và protein, các thuật toán dùng cho tin sinh học…

Công nghệ sinh học phân tử hiện đại cho phép thu thập rất nhiều loại dữ liệu, thông tin liên quan tới gen và protein. Có một vấn đề đặt ra là sau khi xác định được chức năng của gen, nếu phát hiện được gen đấy có nằm trong một chuỗi ADN, thì làm thế nào để ta có thể xác định được nó hoạt động hay không? Đây không phải là một vấn đề mới trong sinh học, nhưng có vai trò quan trọng trong việc tìm hiểu cơ chế hoạt động của các quá trình sinh học.

Một trong những tác nhân quan trọng tác động lớn đến sự hoạt động của gen là vùng tăng cường. Vì vậy, bài toán về xác định sự hoạt động của gen có thể được đưa về bài toán nhỏ hơn là xác định trong chuỗi ADN chứa gen ấy có chứa các vùng tăng cường tương ứng của nó hay không. Nhưng làm thế nào để có thể tìm ra được các vùng tăng cường tương ứng với một gen?

Công việc này, trước đây, có thể được thực hiện thông qua quá trình thực nghiệm. Phương pháp này rất chính xác, tuy nhiên, nó rất chậm và tốn rất nhiều công. Đặc biệt là với tình hình hiện nay, khi mà số lượng gen tìm được càng ngày càng nhiều, thì yêu cầu được đặt ra là cần tìm ra được những phương pháp mới nhanh và rẻ hơn, và phương pháp tính toán (tin sinh học) là một trong những phương pháp có thể đáp ứng được những yêu cầu đó. Đã có nhiều phương pháp tính toán được đưa ra để giải quyết bài toán, mỗi phương pháp có những ưu điểm, nhược điểm riêng. Trong luận văn này, em sẽ tìm hiểu về nhóm phương pháp sử dụng máy vector hỗ trợ quyết định SVM để tìm ra các vùng tăng cường tương ứng với gen. Do đặc trưng của SVM, khi ta áp dụng các hàm kernel khác nhau thì kết quả được trả về cũng sẽ khác nhau. Trong luận văn này, em xin đề xuất một cách xây dựng hàm kernel mà khi áp dụng vào SVM để xác định một chuỗi ADN có phải là vùng tăng cường của một gen hay không sẽ đưa ra được kết quả với độ chính xác cao hơn khi so sánh với những phương pháp khác.

Báo cáo bao gồm 3 chương

**Chương 1: CÁC KHÁI NIỆM VÀ VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN XÁC ĐỊNH VÙNG TĂNG CƯỜNG**

Trong chương này, em sẽ giới thiệu về các khái niệm trong tin sinh học liên quan đến bài toán xác định vùng tăng cường như sự hoạt động của gen, điều hòa hoạt động của gen, nhân tố phiên mã (transcription factor), vùng tăng cường (enhancer) và trình bày về các nhóm phương pháp tìm kiếm vùng tăng cường đã có.

**Chương 2: PHƯƠNG PHÁP XÁC ĐỊNH VÙNG TĂNG CƯỜNG DỰA TRÊN SVM**

Trong chương này, em sẽ trình bày về nhóm phương pháp xác định vùng tăng cường dựa trên máy vector hỗ trợ quyết định SVM [1], [14], giới thiệu về các hàm kernel đã được sử dụng để giải quyết bài toán và đề xuất cách xây dựng hàm kernel để đạt được kết quả tối ưu.

**Chương 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ**

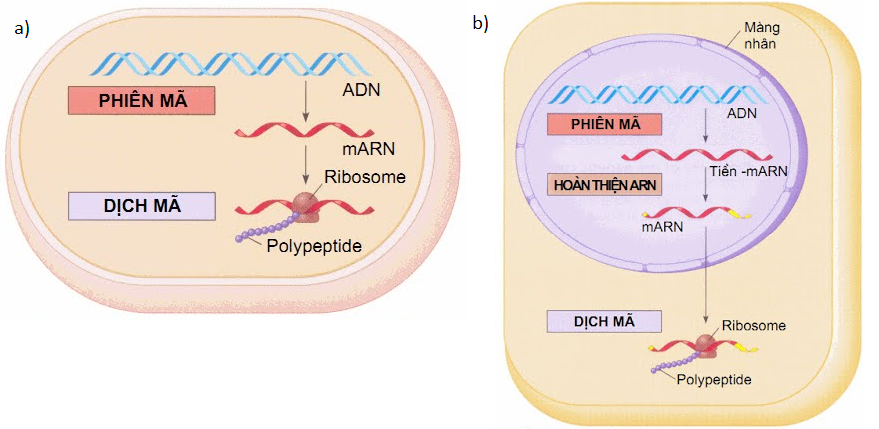
Trong chương này, em sẽ trình bày về cách thức xây dựng bộ dữ liệu mẫu, quá trình cài đặt phương pháp xác định vùng tăng cường đã được giới thiệu ở chương 2, tiêu chí để đánh giá tính hiệu quả của phương pháp, đánh giá kết quả, so sánh kết quả với các phương pháp khác, đưa ra nhận xét về phương pháp cũng như hướng phát triển trong tương lai.

# CHƯƠNG 1: CÁC KHÁI NIỆM VÀ VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN XÁC ĐỊNH VÙNG TĂNG CƯỜNG

1. **Các khái niệm**
2. **Sự hoạt động của gen**

Các gen là đơn vị mang thông tin di truyền của sinh vật. Tùy vào loại tế bào và điều kiện môi trường cụ thế, một gen có thể hoạt động hay không hoạt động. Sự hoạt động của gen (hay còn gọi là biểu hiện gen, thuật ngữ tiếng anh: gene expression) là quá trình tổng hợp thành các sản phẩm gen từ những thông tin được mã hóa trong gen. Các sản phẩm gen thường là các protein, nhưng cũng tồn tại các gen không mã hóa thông tin cho protein như rARN, tARN hay snARN, sản phẩm của những gen này là các ARN chức năng.

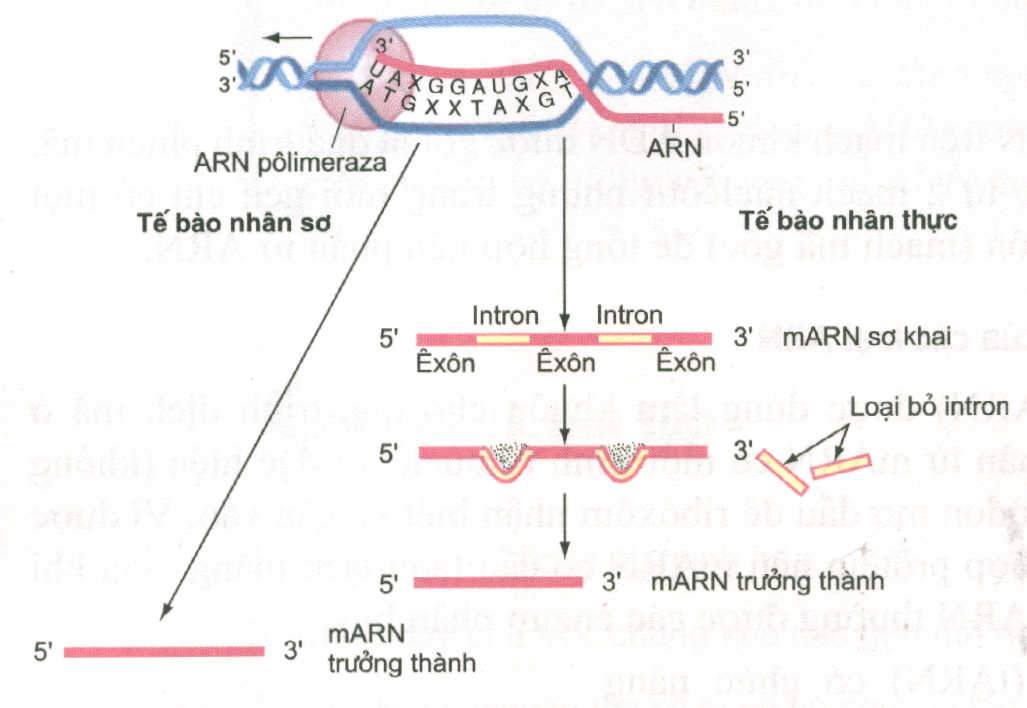
Quá trình hoạt động của gen bao gồm 2 giai đoạn chính là phiên mã và giải mã như minh họa ở Hình 1.1. Đối với sinh vật nhân sơ, do trong tế bào không có nhân nên mARN được tạo ra từ quá trình phiên mã sẽ được dùng ngay để dịch mã mà không cần biến đổi gì thêm. Còn đối với các sinh vật có nhân thực thì do nhân tạo thành không gian tách biệt cho quá trình phiên mã nên bản phiên mã ARN đầu tiên, được gọi là tiền-ARN được biến đổi qua một số bước trước khi rời nhân ở dạng ARN hoàn thiện.



**Hình 1.1. Sơ đồ khái quát sự hoạt động của gen ở sinh vật nhân sơ (a) và nhân thực (b)**

1. **Quá trình phiên mã (hay còn gọi là sao mã)**

Phiên mã là quá trình sao chép thông tin di truyền được mã hoá dưới dạng trình tự các nucleotide trên gen thành dạng trình tự các ribonucleotide trên ARN thông tin (mARN) để mARN trực tiếp thực hiện chức năng truyền đạt thông tin di truyền đến cấu trúc phân tử protein trong quá trình giải mã nhờ đó mà tổng hợp những protein đặc thù cho gene. Quá trình phiên mã ở nhân sơ và nhân thực cơ bản giống nhau nhưng cũng có một số đặc trưng riêng.



**Hình 1.2. Sơ đồ khái quát quá trình phiên mã**

ARN được phiên mã từ ADN nhờ phức hệ enzim ARN polymerase. Phức hệ này vừa liên kết với ADN, vừa mở xoắn. Giống như enzim ADN polymerase, phức hệ phiên mã di chuyển từ đầu 3’ đến đầu 5’ của ADN và tổng hợp một sợi có chiều ngược lại (từ đầu 5’ đến đầu 3’), trong trường hợp này là một sợi gồm các ribonucleotid. Ở nhóm nhân sơ, enzim ARN polymerase chịu trách nhiệm cho sự tổng hợp tất cả các ARN. Ở nhóm nhân chuẩn, enzim ARN polymerase II chịu trách nhiệm đối với sự tổng hợp mARN. Hai loại enzim ARN polymerase khác tổng hợp tARN và rARN và các ARN khác sử dụng trong cấu trúc tế bào hoặc enzim.

Như vậy, ta có thể thấy rằng: ADN luôn được sử dụng làm khuôn mẫu cho sự tổng hợp ARN. Sự tổng hợp ARN xảy ra theo cách thức tương tự như sự tạo thành ADN mới: hai sợi của phân tử ADN tách rời nhau ra, ARN được tổng hợp dọc theo một trong hai sợi nhờ enzim ARN polymerase. Quá trình này tuân theo quy tắc bắt cặp bazơ như sau: ứng với adenine (A) trên sợi khuôn của ADN là uracil (U) trên sợi ARN, với timin (T) là adenine (A), với guanin (G) là cytosin (C) và với cytosin (C) là guanine (G).

1. **Quá trình giải mã (hay còn gọi là dịch mã)**

Sau khi quá trình phiên mã kết thúc, ARN thông tin (mARN) được tạo ra sẽ có chứa trình tự các bazơ qui định thành phần và thứ tự các acid amin trong phân tử protein do gen mã hóa. Trước trình tự này là phần mở đầu và cuối nó là phần kết thúc ngắn. Trình tự này sẽ tiếp tục được giải mã ở tế bào chất bởi một quá trình gọi là quá trình giải mã. Trong quá trình này, tế bào phải “phiên dịch” trình tự các bazơ của một phân tử mARN thành trình tự các axit amin của một chuỗi polypeptit. Vị trí diễn ra sự dịch mã là các ribosome; đó là phức hệ dạng hạt tạo điều kiện thuận lợi cho sự kết nối các axit amin theo một trật tự nhất định để hình thành nên các chuỗi polypeptit. Sau đó, từ các chuỗi polypeptit này, sẽ tạo thành protein đơn giản (chỉ bao gồm các chuỗi polypeptit) hoặc protein phức tạp (ngoài các chuỗi polypeptit ra còn có thêm một số thành phần phi protein như: axit nucleic, lipit, cacbonhidrat).

1. **Điều hòa sự hoạt động của gen**

Mỗi tế bào trong cơ thể của một sinh vật đa bào đều nhận trọn vẹn một bộ nhiễm sắc thể, một bản sao của ADN trong tế bào ban đầu mà từ đó sinh vật phát triển. Các trình tự nucleotid của ADN này chứa đựng toàn bộ thông tin di truyền, đó chính là cơ sở của tiến hóa tế bào. Dù cho mỗi tế bào trong cơ thể nhận được một bộ thông tin giống nhau nhưng tế bào này khác với các tế bào kia và cách thức mỗi tế bào phản ứng lại với các tác động từ môi trường là khác hẳn nhau. Nguyên nhân của hiện tượng này là do sự hoạt động của gen diễn ra trong tế bào này không giống với tế bào khác. Chỉ có một số gen trong bộ gen của tế bào là sản xuất protein và ở từng thời điểm chỉ có vài phần trăm ADN này hoạt động. Sự hoạt động của gen khác nhau giữa các tế bào là kết quả của một quá trình gọi là điều hòa sự hoạt động của gen. Quá trình này sẽ quyết định sự phát triển và phản ứng của gen đối với môi trường.

Ở quá trình phiên mã, sự hoạt động của gen bị điều khiển bởi các nhân tố phiên mã (transcription factor – TF). Các TF sẽ tác động đến gen dựa vào việc nhận diện các chuỗi nucleotide đặc trưng đối với gen đó, các chuỗi nucleotide này được gọi là các mô típ. Những mô típ này được gọi là với cái tên chung là các vùng liên kết nhân tố phiên mã (transcription factor binding site – TFBS). Các TFBS thường có vị trí ở gần nhau trên chuỗi ADN. Những TFBS ở gần gen mà nó tác động được gọi là các module điều hòa ở trên cùng một phía (cis-regulatory module – CRM), còn những TFBS ở xa gen thì được gọi là các vùng tăng cường (enhancer). Chính xác hơn, ta có thể xem các vùng tăng cường chính là các CRM có thể tăng cường khả năng hoạt động của gen từ xa. Việc xác định được các vùng tăng cường và giải mã được những quy luật liên quan đến chức năng của nó sẽ cung cấp cho ta những thông tin cần thiết để có thể lý giải được cách thức điều hòa sự hoạt động của gen diễn ra trong tế bào.

1. **Nhân tố phiên mã (transcription factor)**

Như đã đề cập ở phần trước, nhân tố phiên mã (TF) là một thành phần quan trọng trong quá trình điều hòa sự hoạt động của gen ở giai đoạn phiên mã. Chúng thường là các protein, nhưng cũng có thể bao gồm một đoạn ARN ngắn, và không được mã hóa. Những TF có liên quan đến nhau thường nhóm lại và thực hiện tương tác lẫn nhau để điều khiển quá trình phiên mã theo các mức độ khác nhau. Các gen luôn có những vùng có những mô típ đặc trưng riêng biệt (được gọi chung là các CRM) và có thể được nhận diện bởi rất nhiều loại TF. Các TF này sẽ liên kết, tác động lên một loạt các TF khác và tạo thành một chuỗi tác động phức tạp. Cuối cùng sẽ tác động đến enzim ARN polymerase, từ đó bắt đầu (hoặc gây ức chế) quá trình phiên mã.

Ở các sinh vật nhân thực, các gen thường được mặc định là “tắt” – tức là chúng không thể hoạt động được, và các TF sẽ nhân tố kích hoạt để “mở” các gen này, giúp chúng có thể hoạt động. Ở sinh vật nhân sơ thì ngược lại, mặc định, các gen sẽ ở chế độ “mở”, tức là luôn luôn hoạt đông, chỉ khi có TF gây ức chế, tác động đến nó, không cho gen ấy hoạt động thì nó sẽ chuyển về trạng thái “tắt”.

1. **Cơ chế hoạt động của TF**

Để điều hòa hoạt động của một gen, các TF liên kết với những CRM của gen đấy. Phụ thuộc vào TF, quá trình phiên mã của gen đấy sẽ là điều hòa “lên” hay điều hòa “xuống” (tức là được thực hiên hay không được thực hiện). Các TF sử dụng nhiều cơ chế để điều hòa hoạt động của gen, gồm có:

* Làm ổn định hoặc gây ức chế sự ràng buộc của enzim ARN polymerase lên ADN.
* Gây xúc tác trực tiếp hoặc thông qua những protein khác để thúc đẩy quá trình axetyl hóa hoặc deaxetyl hóa của protein histone (là một protein đơn giản kết hợp với một nucleic axit để tạo nên một nucleoprotein, có nhiệm vụ nhóm lại và sắp xếp các ADN thành một nucleosome). Quá trình axetyl hóa của protein histone sẽ tương ứng với sự điều hòa “lên” trong khi ngược lại, quá trình deaxetyl hóa sẽ tương ứng với sự điều hòa “xuống” sự hoạt động của gen.
* Kết hợp với những protein khác như protein phối hợp hoạt hóa (coactivator) hay protein phối hợp kìm hãm (corepressor) để tạo thành một TF phức hợp nhằm tác động đến quá trình phiên mã.

1. **Sự điều hòa hoạt động của các TF.**

Trong sinh vật học, đối với những quá trình quan trọng, thông thường có rất nhiều tầng cho quá trình điều hòa và điều khiển. Điều này cũng đúng với các TF: không chỉ các TF điều khiển mức độ phiên mã nhằm điều hòa hoạt động của gen mà các TF cũng được điều hòa bởi các TF khác. Dưới đây là một vài cách thức mà các TF điều hòa tác động lẫn nhau:

* Sự tổng hợp hóa học: các TF cũng là các protein, do đó, chúng cũng đều được phiên mã từ một gen trên một nhiễm sắc thể để thành ARN, và tiếp đó từ ARN sẽ được dịch thành protein. Tất cả những bước này có thể bị điều hòa bởi một hoặc một vài TF nào đó. Hơn nữa, các TF có thể tự điều hòa chính mình. Ví dụ, trong một vòng lặp phản hồi âm tính (negative feedback loop), TF hoạt động như là chất kìm hãm của chính nó: nếu TF liên kết với ADN của gen của chính nó, nó sẽ điều hòa “xuống” sự hoạt động của gen ấy, từ đó làm giảm quá trình tạo ra TF đó. Đây là một cơ chế để duy trì sự ổn định về mặt số lượng của một TF trong một tế bào.
* Định vị nhân: ở sinh vật nhân thực, giống như các protein khác, các TF được phiên mã ở trong nhân và tiếp đó được dịch ở tế bào chất của tế bào. Nhiều protein hoạt động ở trong nhân có chứa tín hiệu định vị nhân để đưa chúng trở lại nhân sau khi được phiên dịch ra. Nhưng với nhiều loại TF, đây là điểm chính trong quá trình điều hòa của chúng [18]. Những lớp quan trọng của các TF như một vài cơ quan cảm thụ của nhân (nuclear receptor) đầu tiên phải được liên kết với một phối tử (ligand) trong tế bào chất trước khi chúng có thể định vị lại vị trí của nhân [18]. Nếu tín hiệu này không được đảm bảo, TF có thể không được tạo ra, hoặc không thể hoạt động được.
* Sự hoạt động của TF: các TF có thể hoạt động (hoặc không hoạt động) thông qua miền cảm ứng tín hiệu (signal-sensing domain) của chúng bằng một trong các cơ chế sau:
  + Liên kết phối tử (ligand binding): liên kết phối tử không chỉ ảnh hưởng đến vị trí của TF trong tế bào mà còn ảnh hưởng đến trạng thái hoạt động của TF và khả năng liên kết với ADN hoặc các nhân tố phối hợp (co-factor) khác (như cơ quan cảm thụ của nhân – nuclear receptor).
  + Quá trình photphatryl hóa: nhiều TF cần phải được photphatryl hóa trước bị có thể liên kết với ADN, chẳng hạn như các STAT protein.
  + Tương tác với các TF khác hoặc các protein đồng điều chỉnh.
* Mức độ sẵn sàng của những TF hay các nhân tố phối hợp (co-factor) khác: hầu hết các TF không hoạt động một cách độc lập mà khi quá trình phiên mã gen xảy ra mà một lượng lớn các TF phải liên kết với chuỗi ADN cần điều hòa. Các TF này, lần lượt, kết hợp với các protein trung gian như các nhân tố phối hợp (co-factor) giúp làm tăng hiệu quả quá trình kết hợp của phức hợp khởi đầu (preinitiation complex) với enzim ARN polymerase. Vì thế, để một TF đơn có thể khởi đầu quá trình phiên mã, tất cả những protein khác liên quan và nó phải ở trong một vùng sao cho TF này có thể liên kết với chúng khi cần thiết.

1. **Vùng tăng cường (enhancer)**

Vùng tăng cường (enhancer) là một đoạn ADN ngắn có thể liên kết với các TF để tăng cường khả năng phiên mã của gen trong một nhóm gen (gene cluster – là một tập gồm hai hoặc nhiều gen được dùng để mã hóa những sản phẩm (sản phẩm của gen gồm protein và ARN) giống nhau hoặc tương tự nhau). Nhưng khác với các CRM khác, vùng tăng cường không cần phải ở vị trí gần với gen mà nó tác động đến, thậm chí trong một số trường hợp còn không nằm cùng một sợi nhiễm sắc thể.

1. **Vị trí của vùng tăng cường**

Một vùng tăng cường có thể ở trước hoặc sau so với gen mà nó điều hòa. Thậm chí, vùng tăng cường không cần phải ở gần vùng khởi đầu phiên mã để tác động đến quá trình phiên mã, mà một vài vùng tăng cường còn được tìm thấy ở khoảng cách khá xa so với vùng khởi đầu, có lúc lên đấy vài trăm ngàn bp. Trong tế báo của sinh vật (cả nhân sơ và nhân thực), cấu trúc của nhóm chất nhiễm sắc (chromatin complex) của ADN được sắp xếp theo hình dạng xoắn ốc. Vì thế, mặc dù các vùng tăng cường thường cách xa gen mà nó tác động nếu tính theo số lượng nucletotide, nhưng về mặt hình học, các vùng này lại rất gần gen đấy. Điều này cho phép vùng tăng cường có thể tương tác với những TF và enzim ARN polymerase. Ở sinh vật nhân thực, có một nhân tố đối nghịch hoàn toàn với vùng tăng cường, đó là vùng làm giảm (silencer), nghĩa là khi vùng làm giảm liên kết với những TF thích hợp (được gọi là các chất kìm hãm – repressor), nó sẽ làm giảm khả năng hoạt động của gen, thay vì làm tăng cường như vùng tăng cường. Vùng làm giảm và vùng tăng cường thường ở gần nhau, hay thậm chí là cùng một vùng ADN, lúc này chúng chỉ được phân biệt bởi TF mà vùng ADN ấy liên kết.

Vùng tăng cường không tác động trực tiếp lên vùng khởi đầu mà thông qua các protein hoạt hóa (activator protein). Những protein hoạt hóa này sẽ tương tác với các phức hợp trung gian, khiến cho các phức hợp này liên kết với enzim ARN polymerase và các TF để bắt đầu quá trình phiên mã gen. Hướng của vùng tăng cường không ảnh hưởng đến chức năng của nó, hơn nữa, vùng tăng cường còn có thể bị cắt bỏ và chèn vào ở một vùng khác trên nhiễm sắc thể mà vẫn có thể gây ảnh hưởng đến quá trình phiên mã gen.

Các vùng tăng cường có thể được nhận dạng bằng kỹ thuật bẫy vùng tăng cường (enhancer trap) sử dụng một một gen báo hiệu hoặc so sánh chuỗi cần phân tích với những gen đã được xác định bằng cách sử dụng phương pháp tính toán (computational genomics).

1. **Vai trò của vùng tăng cường đối với sự phát triển của sinh vật.**

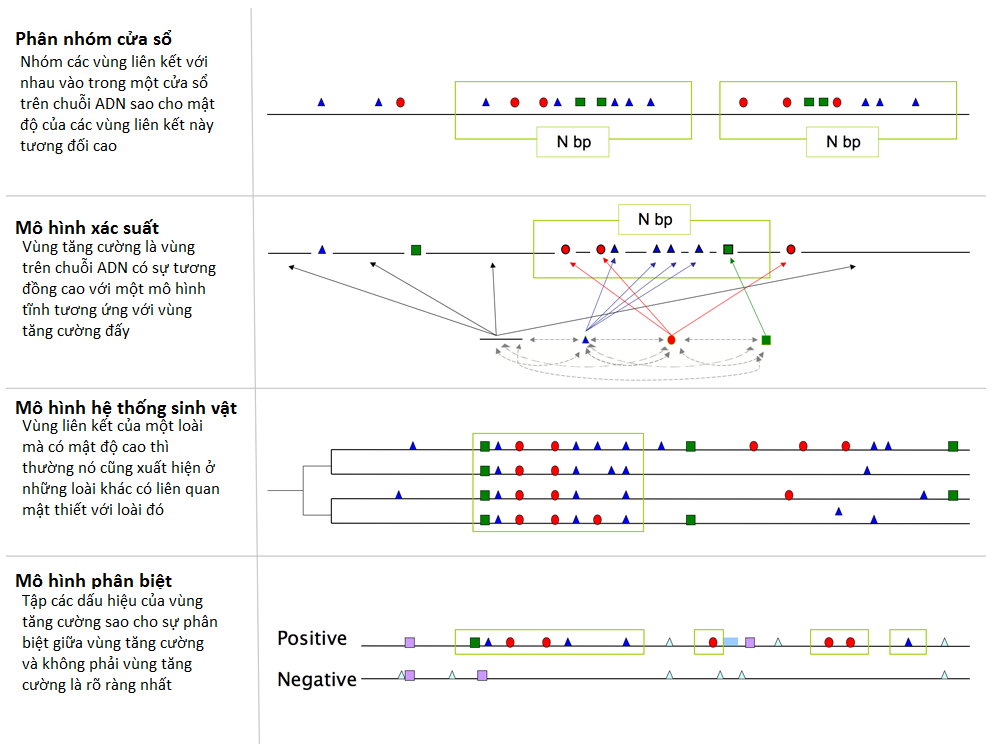
Quá trình phát triển của sinh vật, bao gồm quá trình phân tách, phát triển của các tế bào và mô, đòi hỏi phải được điều hòa chính xác các mô hình của biểu hiện gen. Các vùng tăng cường điều hòa cả về mặt không gian và thời gian đối với quá trình điều khiển sự phát triển bằng cách “bật” quá trình phiên mã đối với những tế bào thích hợp và/hoặc “tắt” quá tình phiên mã đối với những gen khác. Vì thế, sự kết hợp của các TF và những protein liên kết ADN khác trong một chuỗi phát triển sẽ quyết định những gen nào được hoạt động ở trong chuỗi đó. Các gen này có thể được sử dụng ở nhiều quá trình khác nhau (về cả mặt không gian và thời gian) nhờ vào vùng tăng cường.

Một vài gen có vai trò cực kỳ quan trọng trong quá trình phát triển có thể có nhiều vùng tăng cường tương ứng với nó, các vùng tăng cường này có chức năng trùng lặp với nhau. Những vùng tăng cường phụ (còn được gọi là các “vùng tăng cường bóng” – “shadow enhancer”) này có thể được tìm thấy ở khoảng cách nhiều kbp so với vùng tăng cường chính (“primary enhancer” – là vùng tăng cường được tìm thấy đầu tiên và thường ở gần gen mà nó điều hòa hơn so với những vùng tăng cường khác). Mỗi vùng tăng cường trong nhóm này điều hòa các mô hình của biểu hiện gen theo cách thức gần như giống nhau. Điều này dẫn đến câu hỏi liệu có phải là sự lãng phí khi có nhiều hơn một vùng tăng cường cùng điều hòa một gen? Những nghiên cứu gần đây đã chỉ ra rằng việc có nhiều vùng tăng cường cho phép sinh vật có thể vẫn sống và phát triển khi có sự thay đổi của môi trường (chẳng hạn như khi nhiệt độ tăng). Khi sự thay đổi của môi trường đến một ngưỡng nào đó, nếu chỉ có một vùng tăng cường thì việc điều hòa quá trình hoạt động của gen có thể thất bại, tuy nhiên, nếu như có nhiều vùng tăng cường cùng tham gia thì sự điều hòa này vẫn diễn ra bình thường. Điều này đảm bảo cho sinh vật có thể thích nghi với sự thay đổi của môi trường tốt hơn nhiều.

1. **Các nhóm phương pháp xác định vùng tăng cường**

Để có thể xác định được một vùng tăng cường, ta có thể dựa vào sự ràng buộc về mặt không gian (nghĩa là một vùng tăng cường phải ở gần những vùng ADN khác liên kết với nó) và sự ràng buộc về mặt hệ thống loài (nghĩa là một vùng tăng cường có thể tồn tại ở nhiều loài có mối liên quan chặt chẽ đến nhau). Hiện nay có rất nhiều phương pháp được đưa ra để tìm kiếm vùng tăng cường. Những phương pháp này có thể chia một cách tương đối thành 4 nhóm sau: phân nhóm cửa sổ (windows clustering), mô hình xác suất (probabilistic modeling), dựa vào mô hình hệ thống sinh vật (phylogenetic footprinting) và mô hình phân biệt (discriminative modeling).

* Phân nhóm cửa sổ (windows clustering): phương pháp này thực hiện nhóm các vùng liên kết với nhau vào trong một cửa sổ trên chuỗi ADN sao cho trong cửa sổ này, mật độ của các vùng liên kết này tương đối cao.
* Mô hình xác suất (probabilistic modeling): phương pháp này thực hiện tìm kiếm vùng tăng cường bằng cách xác định các chuỗi ADN có sự tương đồng cao với một mô hình tĩnh tương ứng của vùng tăng cường đấy.
* Dựa vào mô hình hệ thống sinh vật (phylogenetic footprinting): phương pháp này dựa vào đặc điểm về sự ràng buộc về mặt hệ thống loài để tìm vùng tăng cường của một loài bằng cách tìm trong những vùng ADN có mật độ các vùng liên kết cao và chúng đã xuất hiện ở những loài khác có liên quan mật thiết với loài đó.
* Mô hình phân biệt (discriminative modeling): phương pháp này tìm cách xác định những dấu hiệu của vùng tăng cường sao cho khi dựa vào những dấu hiệu này, sự phân biệt giữa một đoạn ADN là vùng tăng cường và không phải vùng tăng cường là rõ ràng nhất.



**Hình 1.3. Phân nhóm các phương pháp xác định vùng tăng cường.**

Lưu ý là sự phân nhóm này chỉ mang tính chất tương đối, có nhiều phương pháp mang đặc điểm của hai hoặc nhiều nhóm ở trên. Trong các nhóm phương pháp trên, 3 nhóm phương pháp đầu chỉ cần một tập các vùng tăng cường để làm dữ liệu đầu vào, hay thậm chí là không cần dữ liệu đầu vào nào. Trong khi ở phương pháp mô hình phân biệt, nó cần 2 tập dữ liệu, một tập gồm các chuỗi là các vùng tăng cường, và tập còn lại là những chuỗi chắc chắn không phải là vùng tăng cường.

Nhóm phương pháp này sẽ xây dựng lên một mô hình để phân biệt những đoạn là vùng tăng cường và những đoạn không phải là vùng tăng cường một cách rõ ràng nhất bằng cách sử dụng các thuật toán phân loại. Nhóm này được đánh giá khá cao do chúng có thể xác định được các vùng tăng cường của những sinh vật có cấu trúc ADN phức tạp, như động vật có vú, trong khi đây là một thách thức khá lớn đối với phương pháp khác. Quá trình xây dựng mô hình cũng chính là quá trình đi tìm những nét đặc trưng thích hợp để có thể sử dụng chúng nhằm xác định được vùng tăng cường. Những nét đặc trưng thích hợp này có thể là những thước đo về độ tương tự [4] giữa các chuỗi ADN hoặc là hàm kernel [11] để sử dụng cùng với những phương pháp phân loại dựa trên hàm kernel, ví dụ như SVM. Để làm được như vậy, các phương pháp trong nhóm này có thể chỉ cần các chuỗi ADN để làm dữ liệu đầu vào, hoặc cần thêm một số thông tin khác nữa mô tả về các vùng tăng cường đã được sử dụng làm đầu vào. Kết quả đánh giá các phương pháp hiện có đã cho thấy rằng, việc chỉ sử dụng thông tin về các chuỗi ADN có thể đưa đến kết quả tương đương với việc sử dụng thêm những thông tin khác, trong khi lại dễ dàng và đơn giản hơn khi tiến hành cài đặt.

1. **Kết luận chương**

Ở trong chương 1, em đã giới thiệu về các khái niệm trong tin sinh học liên quan đến bài toán xác định vùng tăng cường như sự hoạt động của gen, nhân tố phiên mã (transcription factor), vùng tăng cường (enhancer) và đã trình bày về các nhóm phương pháp tìm kiếm vùng tăng cường đã có. Trong các nhóm phương pháp đã trình bày thì nhóm phương pháp mô hình phân biệt có những điểm nổi trội hơn so với những nhóm phương pháp khác. Chính vì thế, phương pháp em đề xuất ở trong báo cáo này cũng thuộc về nhóm này. Nó tương tự với các phương pháp được trình bày trong [4], [5] nhưng có sử dụng thêm cả thông tin về vị trí của các vùng tăng cường khi xây dựng hàm kernel, từ đó giúp cho kết quả thu được có độ chính xác cao hơn. Ở phần kế tiếp, em sẽ trình bày về nhóm phương pháp xác định vùng tăng cường dựa trên SVM [1], [14] , cũng như cách xây dựng hàm kernel của mình để đạt được kết quả tối ưu.

# CHƯƠNG 2: PHƯƠNG PHÁP XÁC ĐỊNH VÙNG TĂNG CƯỜNG DỰA TRÊN SVM

* 1. **Phương pháp xác định vùng tăng cường dựa trên SVM**
     1. **Giới thiệu về SVM**

Thuật toán phân lớp SVM được đề xuất bởi Cortes và Vapnik vào năm 1995 [14]. Nhưng những ý tưởng chính của thuật toán phân lớp SVM bắt nguồn từ hai công trình của Vapnik và Lerner vào năm 1963 [15] và Vapnik và Chervonenkis vào năm 1964 [16].

Thuật toán SVM là một bộ phân lớp nhị phân được xây dựng cho một tập dữ liệu huấn luyện như sau:

Gọi với là tập dữ liệu đầu vào và tương ứng là tập dữ liệu kết quả, hay còn gọi là nhãn của các mẫu dữ liệu đầu vào với . được gọi là tập dữ liệu huấn luyện cho thuật toán phân lớp SVM.

Bộ phân lớp tuyến tính được mô hình như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Với là vector trọng số và . Khi đó, ta có ràng buộc dữ liệu cho thuật toán SVM như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Ta có thể kết hợp tập điều kiện (2.2) thành:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Với điều kiện ràng buộc như trong (2.3), với các tập dữ liệu không thể phân tách được trên tất cả các mẫu học thì lời giải cho thuật toán phân lớp SVM là rỗng, điều này rất dễ xảy ra trong thực tế, do dữ liệu huấn luyện luôn có nhiễu. Để giải quyết cho trường hợp này, Cortes và Vapnik [3] đã thay đổi công thức (2.3) thành:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Với biến slack để giải quyết cho trường hợp một số mẫu trong tập dữ liệu huấn luyện vi phạm điều kiện phân lớp. Ta có thể thấy những mẫu có là những mẫu vi phạm điều kiện phân lớp so với ràng buộc trong (2.3).

Công thức tối ưu dạng nguyên thủy (primal problem) theo không gian trọng số của SVM có dạng như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Với C là một số nguyên dương, được sử dụng để điều khiển giữa việc tối ưu hàm mục tiêu và những mẫu vi phạm ràng buộc phân lớp của SVM trong (2.3).

Từ (2.5), ta có biểu thức Lagrangian tương ứng là:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Với các hệ số Lagrangian với Từ biểu thức (2.6), ta

có lời giải của vấn đề tương ứng với việc giải bài toán:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Lấy đạo hàm từng phần cho mỗi biến của hàm Lagrangian L trong (2.6), ta có:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Thay (2.8) vào (2.6) ta có bài toán đối ngẫu dạng tối ưu bậc hai (Dual Quadratic Programming) cho bài toán SVM như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Do biểu thức (2.9) là dạng bài toán tối ưu bậc hai (Quadratic Programming), do vậy có thể sử dụng các bộ giải tối ưu (optimization solvers) để tìm lời giải.

* + 1. **Giới thiệu về hàm kernel trong SVM**

Để mở rộng khả năng phân lớp của thuật toán SVM, thay vì sử dụng hàm tích nội (inner product), người ta đưa ra khái niệm kernel để đo độ tương đồng giữa 2 mẫu trong không gian dữ liệu huấn luyện đầu vào.

Đầu tiên, dữ liệu huấn luyện sẽ được chuyển sang không gian *H* (chính là không gian đặc trưng (feature space)) bằng hàm ánh xạ như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Để tính độ tương đồng giữa các mẫu học trong *H*, ta có thể sử dụng hàm tích nội tương ứng trong không gian *H*, ký hiệu . Để tiện lợi, ta định nghĩa hàm tương ứng như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

mà thỏa điều kiện:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Hàm như ở công thức (2.12) chính là hàm kernel. Với cách thức thay hàm tích nội bởi hàm kernel như trên, chúng ta có thể sửa dụng SVM trong không gian đặc trưng *H* mà hơn nữa, nếu chúng ta có thể tính được trực tiếp giá trị của mà không cần phải tính chính xác các vector đặc trưng (feature vector), từ đó có thể giúp cho việc tính toán sẽ trở nên nhẹ nhàng hơn rất nhiều đối với không gian đặc trưng có nhiều chiều.

* + 1. **Phương pháp xác định vùng tăng cường dựa trên SVM**

SVM là một phương pháp học máy được đánh giá khá cao và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. Điểm mạnh của phương pháp này đó là tính chính xác và sự linh hoạt, nó có thể làm việc với những nguồn và loại dữ liệu khác nhau, thậm chí là với những dữ liệu không được biểu diễn dưới dạng các vector đặc trưng. Điều này thực sự hữu ích vì những các chuỗi gen của sinh vật thường là những dữ liệu không được biểu diễn dưới dạng vector.

Đối với phương pháp xác định vùng tăng cường dựa trên SVM, bài toán được đưa về dạng bài toán phân loại nhị phân, nghĩa là xác định xem một chuỗi ADN là vùng tăng cường hay không dựa vào một bộ phân loại được huấn luyện trước. Lúc này, bộ phân loại sẽ kiểm tra các chuỗi gen đầu vào xác định vùng nào là vùng tăng cường ta cần tìm kiếm.

Như đã giới thiệu ở trên, quá trình tìm ra một bộ phân loại dựa vào SVM, cũng chính là quá trình ta định nghĩa ra một hàm kernel có thể tính ra được độ tương tự giữa hai đối tượng x và y. Hàm kernel được định nghĩa ra nên phản ánh được miền tri thức mà ta đang sử dụng và điều này là cực kỳ quan trọng, nó sẽ ảnh hưởng đến mức độ hiệu quả của phương pháp. Dưới đây, em sẽ giới thiệu về một số hàm kernel đã được giới thiệu và sử dụng trong các phương pháp xác định vùng tăng cường dựa vào SVM đã có trước đó.

* + 1. **Một số hàm kernel đã được sử dụng để giải quyết bài toán**

Hàm kernel đầu tiên em muốn nói đến là hàm spectrum kernel. Đây là hàm kernel được giới thiệu bởi Leslie [7] và được xem là một trong những hàm kernel đơn giản và dễ dàng sử dụng nhất khi tính toán. Hơn nữa, nó có thể phù hợp với rất nhiều mô hình dữ liệu khác nhau, và khi áp dụng hàm kernel vào SVM thì thời gian để phân loại dữ liệu đầu vào đều ở dạng tuyến tính. Đối với dữ liệu dạng chuỗi, hàm kernel này lại càng được sử dụng rộng rãi do những ưu điểm của nó.

Leslie đã đưa ra định nghĩa *k-mer* là tất cả những đoạn có độ dài *k* có thể tạo được từ tất cả các ký tự thuộc tập *Ʃ (Ʃ = {A, C, T, G}* đối với các chuỗi ADN). Khi đó, vector đặc trưng cho một chuỗi *x* có độ dài *l* (trong chuỗi *x* có chứa các ký tự thuộc tập *Ʃ*) có thể được biểu diễn như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Trong đó, là số lần *k-mer* xuất hiện ở trong chuỗi *x*. Đối với trường hợp của ADN, vector đặc trưng sẽ có trường, tương ứng với *k-mer*.

Lúc này, hàm spectrum kernel sẽ trở thành như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Mặc dù số trường trong vector đặc trưng có thể rất lớn do tăng theo *k* với cấp số mũ, nhưng số giá trị khác 0 ở trong vector đặc trưng tối đa chỉ là . Hơn nữa, bản chất của một hàm kernel là bắt cặp tự do (alignment-free) bởi vì nó không cần phải tiến thành bắt cặp hai chuỗi khi tính độ tương đồng giữa chúng, điều này giúp cho việc tính toán hàm kernel này trở nên dễ dàng và thuận tiện hơn rất nhiều.

Từ hàm spectrum kernel, Leslie đã mở rộng nó để tạo thành hàm kernel ghép đôi bất đối xứng (mismatch kernel) [8], được ký hiệu là , trong đó, *k* chính là dộ dài của *k-mer*, còn *m* quy định số lượng ký tự tối đa có thể khác khi đếm số lần xuất hiện của một *k-mer* trên một chuỗi. Nghĩa là, với một *k-mer* , khi đếm số lần xuất hiện của trên một chuỗi, ta sẽ tính thêm cả số lần xuất hiện của các chuỗi khác sao cho chỉ khác tối đa *m* ký tự. Điều này cho phép quá trình so sánh giữa các chuỗi có thể ghi nhận được cả các mô típ có thể bị biến đổi trong quá trình phát triển và tiến hóa.

Cũng được mở rộng từ hàm spectrum kernel, có một hàm kernel khác được đánh giá cao là hàm kernel của Lee et al., được giới thiệu ở [6]. Như ta đã biết, các chuỗi ADN được tạo thành từ các bazơ A, T, G, C, các bazơ này bắt cặp theo quy tắc: A bắt cặp với T và ngược lại, G bắt cặp với C và ngược lại. Do đó, ứng với mỗi *k-mer*, ta sẽ có một *k-mer* khác ở dạng đảo ngược với nó. Ví dụ như ta có chuỗi TTGCGAT, thì dạng đảo ngược của nó sẽ là ATCGCAA. Khi xây dựng vector đặc trưng cho hàm kernel mới, Lee et al. [6] đã xem một *k-mer* và *k-mer* khác ở dạng đảo ngược của nó là như nhau và sẽ chỉ sử dụng những *k-mer* khác biệt hoàn toàn để làm giảm độ lớn của không gian đặc trưng, từ đó giúp cho quá trình tính toán càng trở nên dễ dàng hơn và đồng thời cũng thu được kết quả tốt hơn.

* 1. **Các hàm kernel được đề xuất**

### **Hàm kernel dựa trên entropy**

Hàm spectrum kernel và những hàm mở rộng của nó như hàm kernel ghép đôi bất đối xứng (mismatch kernel) dựa trên sự xuất hiện cùng nhau của các *k-mer* trong các chuỗi khác nhau mà không quan tâm đến vị trí của các *k-mer* này. Vì thế, nếu như nhiều *k-mer* thường xuất hiện cùng với nhau hoặc ở vị trí gần giống nhau ở các chuỗi đầu vào thì các hàm kernel này cũng không thể ghi nhận được các đặc điểm này. Trong thực tế, để quá trình liên kết có thể diễn ra, trong một vài trường hợp, các TFBS của cùng một TF cần phải nằm gần nhau (khoảng cách giữa chúng thường trong khoảng 10 bp). Như trong Hình 2.1a*,* ta có hai chuỗi, mỗi chuỗi đều có 2 trường hợp của cùng một motif. Ở chuỗi đầu tiên, hai trường hợp có vị trí ở gần nhau và được xem như là 2 vùng liên kết với TF tương ứng. Nhưng ở trên chuỗi thứ 2, chúng lại cách xa nhau nên TF không thể nhận diện được chúng.

*H*=1.0, *fe*=0.37

*H*=0, *fe*=1.0

(a)

(b)

**Hình 2.1. Hình minh họa cho hàm kernel dựa trên entropy (a) và hàm kernel phân cấp (b)**

Trong phần này, em sẽ giới thiệu một hàm kernel mới có kết hợp với thông tin về khoảng cách giữa các *k-mer* giống nhau. Khi xem xét đến số lần mà mỗi *k-mer* xuất hiện ở trên chuỗi đầu vào, em sử dụng một đơn vị đo lường để mô tả mức độ tập trung của nó, nghĩa là xem xét nó xuất hiện gần nhau hay là trải đều trên toàn bộ chuỗi. Đơn vị đo lường được chọn là entropy.

Cách thức tính entropy như sau: cho một chuỗi có độ dài *l*, đầu tiên em chia chuỗi đó thành *n* phần có độ dài bằng nhau (các phần này có thể chồng lên nhau để đảm bảo các *k-mer* quan trọng không bị bỏ qua khi chúng nằm ở vùng ranh giới giữa các phần). Tiếp đó, lần lượt với mỗi *k-mer*, em đếm số lần mà nó xuất hiện trên *n* phần này, sau đó em chuẩn hóa các số đã thu được để tổng của chúng = 1, giả sử khi đó, các giá trị thu được là . Nếu các vị trí xuất hiện của một *k-mer* ở trên chuỗi nằm gần nhau, thì chỉ có một hoặc một vài là có giá trị khác 0. Ngược lại, giải giá trị của sẽ rất gần với giải giá trị chuẩn. Để có thể lượng hóa đặc trưng của sự phân bố này, em tính entropy như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Giá trị của entropy này sẽ trải đều từ (khi *k-mer* chỉ xuất hiện ở trong cùng một phần), đến , khi *k-mer* này trải đều trên toàn bộ *n* phần. Từ giá trị entropy này, em đưa ra giá trị đặc trưng cho một *k-mer* theo công thức sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Từ công thức trên, ta có thể thấy rằng, giá trị của sẽ nằm trong đoạn . Hình 2.1a minh họa một ví dụ về entropy và cách tính giá trị cho 2 chuỗi. Mỗi chuỗi sẽ được chia thành 3 phần có độ dài tương đương nhau, các hình vuông tượng trưng cho các vị trí mà một *k-mer* xuất hiện ở trên chuỗi. Với chuỗi thứ nhất, do *k-mer* chỉ xuất hiện tập trung ở một phần nên ta có . Còn với chuỗi thứ hai thì ta sẽ có . Sau đó, giá trị đặc trưng sẽ được kết hợp với đặc trưng của hàm spectrum. Có khá nhiều cách để kết hợp hai đặc trưng với nhau, như ta có thể tính từng hàm kernel cho từng đặc trưng riêng lẻ, rồi sau đó cộng các hàm kernel đó lại với nhau. Ở đây, em chỉ đơn giản là nối vector của đặc trưng dựa trên entropy với đặc trưng của spectrum để tạo thành vector đặc trưng. Tiếp đó, hàm kernel được tính như là hàm tích nội của vector đặc trưng được kết hợp này.

### **Hàm kernel phân cấp (hierarchical kernel)**

Đặc trưng dựa trên entropy được giới thiệu ở phần trước chỉ có thể ghi nhận được những thông tin về vị trí tương đối của các vị trí xuất hiện của cùng 1 *k-mer*. Tuy nhiên, trên thực tế, có nhiều vùng tăng cường chứa nhiều nhóm của các TFBS khác nhau và các TFBS này thường vị trí gần nhau. Đặc trưng dựa trên entropy chưa thể hiện được các ràng buộc về khoảng cách giữa TFBS khác nhau này. Trong phần này, em sẽ giới thiệu về một loại đặc trưng và hàm kernel khác có thể ghi nhận được thông tin này.

Ý tưởng chính của phương pháp này là: 1) chia chuỗi đầu vào thành các phần có độ dài tương tự nhau; 2) đếm số lần xuất hiện của các *k-mer* trong mỗi phần; và 3) chia các phần này thành các phần con và lặp lại quá trình này cho đến khi độ dài của các phần này là nhỏ nhất để có thể chứa các *k-mer*. Khi quá trình này kết thúc, ta sẽ thu được các con số và chúng sẽ được dùng như là các đặc trưng để tạo thành hàm kernel. Kết quả của quá trình này được thể hiện theo sự phân cấp của các đặc trưng mà trong đó, các đặc trưng ở mức thấp hơn sẽ ghi nhận thông tin về sự xuất hiện cùng nhau của các *k-mer* thường có vị trí gần nhau, còn các đặc trưng ở mức cao hơn sẽ cho thấy sự xuất hiện của các k-mer trong chuỗi đầu vào với ràng buộc về vị trí sẽ yếu hơn.

Với trường hợp phân cấp ở mức 2 lớp, là trường hợp mà em sẽ sử dụng để thực nghiệm, các đặc trưng và hàm kernel sẽ được tính như sau. Đầu tiên, em chia chuỗi đầu vào thành *n* phần có độ dài bằng nhau giống như với trường hợp đặc trưng dựa trên entropy. Tiếp đó, em đếm số lần xuất hiện của mỗi *k-mer* trên toàn bộ chuỗi và trên mỗi phần (Hình 2.1b). Như vậy, ta sẽ có *n + 1* vector chứa số lần xuất hiện của các *k-mer* trên chuỗi đầu vào và các phần của nó. Để làm giảm sự ảnh hưởng của việc các chuỗi đầu vào có thể có độ dài khác nhau, em chuẩn hóa các vector về dạng đơn vị, tức là sao cho . Khi đó, giá trị kernel giữa hai chuỗi *x* và *x’* sẽ được tính như là tổng của *n + 1* kernel, với mỗi kernel trong đó tương ứng với vector đếm số lần xuất hiện của các *k-mer* trên 2 chuỗi x và x’, như sau:

|  |  |  |
| --- | --- | --- |
|  |  | (2.) |

Trong đó, là hàm kernel giữa các vector đếm từ hai chuỗi *x* và *x’*. Nếu là hàm kernel tuyến tính, nghĩa là , thì hàm kernel ở trên có thể được tính bằng cách nối tất cả *n + 1* vector đếm vào một vector dài, và sẽ tính hàm tích nội giữa các vector dài của 2 chuỗi này. Em gọi chúng là các đặc trưng và hàm kernel phân cấp, bởi vì chúng được xây dựng dựa trên quy tắc phân cấp khi chia chuỗi đầu vào thành các phần bé hơn và tiếp tục chia nhỏ trên các phần này.

Điểm chú ý cuối cùng là khi xác định các *k-mer*, tương tự như phương pháp của Lee được đưa ra ở [5], trong phương pháp này, một *k-mer* và *k-mer* ở dạng đảo ngược của nó sẽ được xem là như nhau và sẽ chỉ sử dụng những *k-mer* khác biệt. Ví dụ, AACTTG và CAAGTT sẽ được xem là các *6-mer* giống nhau. Vì thế với phương pháp được cài đặt với , sẽ chỉ có 2080 *6-mer* khác biệt được sử dụng, thay vì là *6-mer* như ở hàm spectrum kernel.

# CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

1. **Cài đặt phương pháp**

### **Xây dựng bộ dữ liệu mẫu**

Để đánh giá phương pháp, em sử dụng bộ dữ liệu về vùng tăng cường của nhiều loại TF khác nhau của người và loài giun tròn. Cụ thể, em sử dụng các dữ liệu có sẵn từ các phương pháp ChIP-chip và ChIP-seq để tạo ra một tập các enhancer cho các TF sau: TAL1 (được lấy từ [10]), HNF4A, GATA6, CDX2, H3K4me2 (được lấy từ [17]) và PHA-4 (được lấy từ [21]). Trừ dữ liệu của TF PHA-4 là dữ liệu của loài giun tròn, các bộ dữ liệu khác đều là của người.

Để có được các tập dữ liệu dương (tức là những tập dữ liệu chứa các vùng tăng cường), em thực hiện các bước tương tự như cách thức đã được sử dụng bởi Yanez-Cuna [19]. Dữ liệu thô ChIP-seq được lấy từ [17], [10] và [21] sẽ được ánh xạ lên trình tự hệ gen của người và giun tròn được lấy từ UCSD Genome Browser (<http://genome.ucsd.edu>). Công đoạn ánh xạ được thực hiện bằng chương trình Bowtie được giới thiệu ở [5]. Bằng cách sử dụng MACS [20], em có thể lấy được danh sách các peak của ChIP (peak là những vùng mang tín hiệu để định danh các vùng chức năng trên ADN trong các quá trình liên kết protein). Sau đó, đối với các TF của người, em sẽ lấy 1000 peak có giá trị p-value cao nhất, còn với loài giun tròn thì sẽ là 500 peak. Với những peak bị trùng lên nhau, là những peak mà khoảng cách giữa chúng chỉ cách nhau tầm khoảng 300bp, em sẽ chỉ lấy 1 peak, còn những vùng khác trong nhóm đó sẽ bị loại. Sau khi đã xác định được một peak, em sẽ mở rộng nó về 2 phía ở trên gen với độ dài khoảng từ 300 đến 500 bp để có được 1 chuỗi vùng tăng cường có độ dài xấp xỉ 1000 bp. Độ dài của phần được mở rộng này được chọn theo cách này để đảm bảo rằng vị trí của peak sẽ không chỉ ở vị trí trung tâm của chuỗi vùng tăng cường, điều đó sẽ làm cho việc dự đoán trở nên khó khăn hơn, giúp cho quá trình đánh giá phương pháp được chính xác hơn.

Để tạo ra các tập dữ liệu âm (nghĩa là các tập dữ liệu gồm các chuỗi đã được xác định không phải là vùng tăng cường), em sử dụng thuật toán tương tự như phương pháp của Lee [5]. Ứng với mỗi tập dữ liệu dương đã có ở trước, em sử dụng thuật toán lấy mẫu – loại bỏ dưới đây để chọn ra ngẫu nhiên các chuỗi gen của loài sinh vật tương ứng, từ đó tạo thành tập dữ liệu âm tương ứng với tập dữ liệu dương. Các tập dữ liệu này được tạo ra sao cho phù hợp với sự phân bố về độ dài chuỗi và trị số lặp lại (repeat fraction) các thành phần của tập dữ liệu dương tương ứng. Để có được trị số lặp lại, ta dựa vào dữ liệu về các chuỗi lặp lại (repeat sequence) được lấy từ UCSD Genome Browser. Dưới đây là thuật toán để có được tập dữ liệu âm, gồm các bước như sau:

1. Lấy ngẫu nhiên một chuỗi dương (là chuỗi vùng tăng cường) có độ dài *l*.
2. Lấy ngẫu nhiên một chuỗi có độ dài *l* từ bộ gen tương ứng với chuỗi dương đã có ở trước và tính ra trị số lặp của nó.
3. Xác suất để chuỗi có ở bước 2 được chọn vào tập dữ liệu là , trong đó, và là xác suất mà có khả năng xuất hiện ở trên chuỗi enhancer và chuỗi gen tương ứng, còn là số tiêu chuẩn.
4. Lặp đi lặp lại các bước từ 1-3 cho đến khi số lượng các chuỗi âm gấp đôi số lượng các chuỗi dương.

Dữ liệu các chuỗi thuộc tập dữ liệu âm và dương sẽ được lưu dưới định dạng file fasta để làm đầu vào cho chương trình được cài đặt nhằm kiểm tra tính hiệu quả của phương pháp.

### **Cài đặt phương pháp**

Phương pháp đề xuất được cài đặt bằng ngôn ngữ Python và có sử dụng bộ công cụ hỗ trợ học máy Shogun [12]. Em sử dụng hàm nhân CommWordStringKernel có sẵn trong bộ công cụ Shogun để cài đặt hàm spectrum kernel. Hàm nhân mới đề xuất được em cài đặt hoàn toàn mới. Em sử dụng thư viện SVM của Shogun với giá trị . Dựa trên kết quả của phương pháp của Lee [5], khi ta có thể thu được kết quả tốt nhất với , tức là sử dụng *6-mer*, em cũng sẽ chỉ sử dụng *6-mer* mà không tiến hành thực nhiệm với những độ dài khác.

1. **Tiêu chí đánh giá**

Lâu nay, đường cong ROC (Receiver Operating Characteristics) vẫn được sử dụng để đánh giá các thuật toán học máy. Ở phương pháp đánh giá này, ta cần sắp xếp các giá trị sau khi phân loại thành thành một dãy tăng dần, sau đó biểu diễn chúng trên đồ thị hai chiều. Khi đó, các điểm này sẽ tạo thành một đường cong, được gọi là đường cong ROC. Tuy nhiên, thông thường, khi so sánh hiệu năng của 2 thuật toán, sự phân biệt giữa hai đường cong ROC tương ứng với chúng ở trên đồ thị là không rõ ràng. Chính vì thế, để so sánh chính xác hơn, người ta đã đưa ra một phép đo khác dựa trên đường cong ROC, đó là AUC (area under the ROC curve). Giá trị AUC chính là diện tích của vùng nằm dưới đường cong ROC (area under the ROC curve), và cũng là một con số thể hiện hiệu năng của thuật toán [9]. Giá trị của AUC nằm trong khoảng (0, 1). Nếu giá trị AUC = 1 thì khi đó ta có thể khẳng định rằng bộ phân loại được đưa ra hoạt động hoàn hảo, nghĩa là tất cả những chuỗi vùng tăng cường sẽ được tách biệt với những chuỗi không phải là vùng tăng cường, khi giá trị AUC = 0.5 thì sẽ tương ứng với quá trình lựa chọn ngẫu nhiên, còn nếu AUC bé hơn 0.5 thì bộ phân loại bị đánh giá là không có hiệu quả.

Thông thường, các SVM sử dụng mức ngưỡng = 0 để quyết định nhãn mà dữ liệu đầu vào có thể được nhận. Tuy nhiên, trong một vài trường hợp, chúng ta có thể phải thay đổi giá trị ngưỡng này để có thể cân bằng giữa độ nhạy và độ đặc trưng. Trong những trường hợp đó, AUC là một tiêu chí phù hợp bởi vì nó có thể tổng hợp cả hiệu năng của bộ phân loại (mà ở đây là SVM) trên toàn bộ dải giá trị của mức ngưỡng và ta không cần phải chọn ra ngưỡng giới hạn cụ thể. Hơn nữa, khi ta sử dụng bộ phân bộ phân loại để gán nhãn cho một tập dữ liệu đầu vào, độ đo AUC sẽ tương đương với xác suất mà bộ phân loại sẽ xếp các dữ liệu dương ở vị trí cao hơn các dữ liệu âm khác [3]. Và như trong [9] đã chứng minh, nếu chúng ta xây dựng một bộ phân loại được tối ưu theo AUC, thì không chỉ giá trị AUC, mà độ chính xác của nó cũng sẽ cao hơn so với những bộ phân loại khác được tối ưu theo độ chính xác.

Để tính giá trị AUC, em sử dụng thuật toán sau [3]:

|  |
| --- |
| **Thuật toán**: Tính AUC |
| **Đầu vào**: L, tập dữ liệu kiểm tra; f(i), xác suất bộ phân loại xác định dữ liệu thứ i là dương; P and N, lần lượt là số lượng dữ liệu dương và âm. |
| **Kết quả**: A, độ đo AUC. |
| **Yêu cầu**: P > 0 and N > 0 |
| 1: Lsorted 🡨 (Sắp xếp L theo thứ tự giảm dần của giá trị f)  2: FP 🡨 TP 🡨 0  3: FPprev 🡨 TPprev 🡨 0  4: A 🡨 0  5: fprev 🡨 - ∞  6: i 🡨 1  7: **while** **do**  8: **if** **then**  9: A 🡨 A + TRAPEZOID\_ AREA (FP, FPprev, TP, TPprev)  10: fprev 🡨 f(i)  11: FPprev 🡨 FP  12: TPprev 🡨 TP  13: **end if**  14: **if** (i là dữ liệu dương) **then**  15: TP 🡨 TP + 1  16: **else** /\* i là dữ liệu âm \*/  17: FPprev 🡨 FP + 1  18: **end if**  19: i 🡨 i + 1  20: **end while**  21: A 🡨 A + TRAPEZOID\_ AREA (N, FPprev, N, TPprev)  22: A 🡨 A/(P × N) /\* đưa giá trị của AUC về khoảng (0,1) \*/  23: **end** |
| 1: **function** TRAPEZOID\_ AREA (X1, X2, Y1, Y2)  2: Base 🡨  3: Heightavg 🡨 (Y1 + Y2)/2  4: **return** Base × Heightavg  5: **end function** |

1. **Đánh giá kết quả**

Để đánh giá hiệu quả của phương pháp, em sử dụng phương thức năm lần xác nhận chéo (five-fold cross-validation). Nghĩa là mỗi tập dữ liệu sẽ được chia ngẫu nhiên thành 5 tập con có số lượng chuỗi dương và âm gần như bằng nhau. Một tập con sẽ được giữ lại để làm tập dữ liệu kiểm tra, còn 4 tập còn lại sẽ được sử dụng để huấn luyện SVM, sau đó SVM này sẽ được dùng để dự đoán trên tập dữ liệu kiểm tra. Quá trình này sẽ được lặp đi lặp lại 5 lần sao cho tất cả các tập con đều sẽ được chọn làm tập dữ liệu kiểm tra. Độ đo AUC sẽ được tính trung bình từ các giá trị có được từ 5 lần lặp đó.

Đầu tiên, em so sánh hai hàm kernel đã đề xuất là hàm kernel phân cấp và hàm kernel dựa trên entropy. Để so sánh hai hàm kernel này, em thay đổi số lượng phần con *n* từ 3 đến 11 phần và các phần con không chồng lên nhau. Giá trị AUC ứng với hàm kernel phân cấp và hàm kernel dựa trên entropy được tổng kết lần lượt ở Bảng 3.1 và Bảng 3.2.

**Bảng 3.1. Giá trị AUC () đối với hàm kernel phân cấp ứng với số lượng các phần khác nhau và giữa các phần không chồng lên nhau. Giá trị AUC cao nhất ứng với mỗi TF được bôi đậm**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TF** | **n=3** | **n=4** | **n=5** | **n=6** | **n=7** | **n=8** | **n=9** | **n=10** | **n=11** |
| TAL1\_erythroid | 89.01 | 86.57 | 92.48 | 89.35 | 93.50 | 90.83 | **93.67** | 91.28 | 93.17 |
| HNF4A\_prolCell | 82.16 | 79.83 | 84.50 | 82.45 | 85.31 | 83.13 | **85.60** | 83.81 | 84.79 |
| GATA6\_prolCell | 97.99 | 96.48 | **98.53** | 97.09 | 98.46 | 97.23 | 98.37 | 97.76 | 97.97 |
| CDX2\_prolCell | 88.00 | 86.96 | **90.55** | 89.16 | 90.11 | 90.10 | 90.05 | 90.15 | 89.65 |
| PHA4\_embryo | 89.68 | 87.36 | 93.15 | 90.40 | **93.46** | 92.22 | 93.45 | 92.97 | 92.96 |
| H3K4me2\_prolCell | 78.90 | 78.67 | 81.77 | 80.62 | 82.74 | 82.10 | 82.40 | **83.16** | 82.13 |
| Trung bình | 87.62 | 85.97 | 90.16 | 88.18 | **90.60** | 89.27 | 90.59 | 89.86 | 90.11 |

**Bảng 3.2. Bảng giá trị AUC () đối với hàm kernel dựa trên entropy ứng với số lượng các phần khác nhau và giữa các phần không chồng lên nhau.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TF** | **n=3** | **n=4** | **n=5** | **n=6** | **n=7** | **n=8** | **n=9** | **n=10** | **n=11** |
| TAL1\_erythroid | 86.22 | 86.14 | 86.05 | 86.00 | 85.87 | 85.80 | 85.62 | 85.57 | 85.65 |
| HNF4A\_prolCell | 78.27 | 78.21 | 78.20 | 78.08 | 78.13 | 78.12 | 78.08 | 77.94 | 77.85 |
| GATA6\_prolCell | 94.85 | 94.89 | 94.71 | 94.86 | 94.62 | 94.85 | 94.54 | 94.82 | 94.29 |
| CDX2\_prolCell | 85.15 | 85.17 | 85.08 | 85.10 | 84.93 | 85.01 | 84.86 | 84.71 | 84.54 |
| PHA4\_embryo | 86.34 | 86.34 | 86.38 | 86.19 | 86.27 | 86.32 | 86.00 | 86.10 | 86.09 |
| H3K4me2\_prolCell | 79.89 | 79.85 | 79.84 | 79.83 | 79.98 | 79.73 | 79.58 | 79.58 | 79.60 |
| Trung bình | 85.12 | 85.1 | 85.04 | 85.01 | 84.96 | 84.97 | 84.78 | 84.78 | 84.67 |

Từ kết quả có được, ta có thể thấy rằng hai phương pháp được đề xuất có điểm số AUC trung bình khá cao với 85% cho hàm kernel dựa trên entropy và 90% cho hàm kernel phân cấp. Tuy nhiên, khoảng cách điểm số giữa hai phương pháp là khá đáng kể khi điểm của hàm kernel dựa trên entropy thấp hơn khoảng 5% so với hàm kernel phân cấp. Nguyên nhân có thể là do những tập dữ liệu được sử dụng đều chứa các vùng tăng cường khác nhau của một TF, trong khi hàm kernel dựa trên entropy được thiết kế cho bài toán xác định một loại vùng tăng cường của một TF. Điểm số AUC trung bình của hàm kernel phân cấp vào khoảng 90.6% và điểm cao nhất là 98.53%, đạt được khi kiểm tra với tập dữ liệu vùng tăng cường được liên kết bởi TF GATA6 và các nhân tố phối hợp (co-factor). Ta cũng có thể nhận thấy rằng, các điểm số AUC tốt nhất mà phương pháp có thể đạt được đều là khi số phần con là từ 5 đến 9 phân, tương đương với độ dài của từng phần con vào khoảng từ 100 bp đến 200 bp.

**Bảng 3.3. Bảng giá trị AUC () đối với hàm kernel phân cấp ứng với số lượng phần con là 7 và các độ dài vùng chồng lên nhau khác nhau.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TF** | **L = 0** | **L = 10** | **L = 20** | **L = 30** | **L = 40** |
| TAL1\_erythroid | 93.50 | 93.11 | 92.39 | 91.70 | 91.09 |
| HNF4A\_prolCell | 85.31 | 85.08 | 84.66 | 84.17 | 83.52 |
| GATA6\_prolCell | 98.46 | 98.51 | 98.52 | 98.35 | 98.17 |
| CDX2\_prolCell | 90.11 | 90.46 | 90.36 | 90.33 | 89.95 |
| PHA4\_embryo | 93.46 | 93.48 | 93.06 | 92.33 | 91.66 |
| H3K4me2\_prolCell | 82.74 | 82.37 | 82.15 | 81.47 | 80.87 |
| Trung bình | 90.59 | 90.50 | 90.19 | 89.72 | 89.21 |

Tiếp theo, em đánh giá hiệu quả của phương pháp hàm kernel phân cấp với các cách chia có độ dài vùng chồng lên nhau khác nhau. Với bước đánh giá này, như kết quả đã có ở trên thì hàm kernel phân cấp có điểm số AUC khá cao khi n có giá trị từ 5 đến 9 nên trong phép so sánh này, em lấy số phần con *n* = 7, còn độ dài *L* của phần chồng lên nhau sẽ được thay đổi từ 0 (tức là các phần con không chồng lên nhau) đến 40 bp. Những điểm số AUC thu được ứng với các trường hợp có độ dài *L* thay đổi được tổng kết ở Bảng 3.3. Như ta có thể thấy, trong hầu hết các trường hợp, việc chia thành các phần con có độ dài phần chồng lên nhau lớn hơn 10 bp sẽ làm giảm hiệu năng của phương pháp. Vì thế, khi sử dụng hàm kernel phân cấp, ta chỉ nên chia phần con với vùng chồng lên nhau có độ dài bằng 0 hoặc hoặc bé hơn 10 bp.

1. **So sánh kết quả với các phương pháp khác**

Để chứng minh tính hiệu quả của phương pháp đề xuất so với những phương pháp hiện có, em so sánh kết quả với phương pháp được đưa ra bởi Lee et al. ở [6], đây là một trong những phương pháp được đánh giá có hiệu quả tốt nhất đến hiện tại khi áp dụng cho các bài toán tương tự như bài toán được đặt ra của luận văn.

**Bảng 3.4. Bảng giá trị AUC () đối với hàm kernel phân cấp ứng với số lượng phần là 7 và phương pháp đưa ra bởi Lee et al.** **[6].**

|  |  |  |
| --- | --- | --- |
| **TF** | **Lee et al. [6]** | **n = 7** |
| TAL1\_erythroid | 86.26 | 93.50 |
| HNF4A\_prolCell | 78.29 | 85.31 |
| GATA6\_prolCell | 94.88 | 98.46 |
| CDX2\_prolCell | 85.18 | 90.11 |
| PHA4\_embryo | 86.33 | 93.46 |
| H3K4me2\_prolCell | 79.88 | 82.74 |
| Trung bình | 85.14 | 90.60 |

Đối với phương pháp đề xuất, em chọn đại diện tiêu biểu nhất là hàm kernel phân cấp với số phần con *n* = 7 và giữa các phần này không có vùng chồng lên nhau. Với phương pháp của Lee et al. [6], do chương trình cài đặt được công khai nên em sẽ chạy chương trình đó với các cài đặt khác đều được để mặc định. Kết quả của quá trình so sánh được tổng kết ở Bảng 3.4. Từ bảng kết quả, ta có thể thấy rằng, kết quả của phương pháp đề xuất có điểm số AUC cao hơn phương pháp của Lee et al. [6] ở tất cả các trường hợp, với mức chênh lệch khoảng 5%. Đây thực sự là một bước cải tiến tuyệt vời khi phương pháp của Lee et al. [6] là một phương pháp được đánh giá khá cao.

1. **Nhận xét về kết quả và hướng phát triển.**

Từ những kết quả thu được từ thực nghiêm, ta có thể thấy rằng, đối với 2 hàm kernel được đề xuất để giải quyết bài toán thì hàm kernel phân cấp hiệu quả hơn so với hàm kernel dựa trên entropy. Đối với cách chia phần con các chuỗi đầu vào, thì ta chỉ nên chia các thành các phần với vùng chồng lên nhau bé hơn 10bp, không nên lớn hơn vì nếu như thế, hiệu năng của phương pháp sẽ bị giảm xuống. Và khi so sánh với những phương pháp khác, phương pháp đề xuất đã chứng tỏ được hiệu năng cao hơn so với phương pháp của Lee et al. [6], một trong những phương pháp được đánh giá có hiệu quả tốt nhất đến hiện tại, mức chênh lệch khoảng 5% với những dữ liệu của người và loài giun tròn. Các hàm kernel được đề xuất đều dựa trên những đặc điểm về chuỗi gen cơ bản và rất đơn giản để cài đặt, tính toán, nó không cần thêm những thông tin khác, cũng như quá trình chuẩn bị dữ liệu đầu vào đơn giản, không phức tạp. Điều này giúp cho phương pháp đề xuất có thể áp dụng vào nhiều bài toán khác nữa.

# KẾT LUẬN

Trong luận văn, em đã trình bày những đặc điểm cơ bản, cũng như các khái niệm cơ bản liên quan đến bài toán tìm kiếm vùng tăng cường của một chuỗi gen, giới thiệu về những nhóm phương pháp tìm kiếm vùng tăng cường đã có. Từ đó, luận văn giới thiệu nhóm phương pháp tìm kiếm vùng tăng cường dựa trên SVM. Phương pháp này đã được nhiều nhà nghiên cứu áp dụng bằng cách sử dụng nhiều hàm kernel khác nhau và đã thu được những kết quả tốt. Em cũng đã đề xuất hàm kernel mới chỉ sử dụng tập dữ liệu chuỗi gen và có thể ghi nhận được những thông tin về vị trí của các TFBS là hàm kernel dựa trên entropy và hàm kernel phân cấp.

Khi tiến hành thực nghiệm, do những tập dữ liệu được sử dụng đều chứa các vùng tăng cường khác nhau của một TF, trong khi hàm kernel dựa trên entropy được thiết kế cho bài toán xác định một vùng tăng cường của một TF nên kết quả của hàm kernel phân cấp cao hơn so với của hàm kernel dựa trên entropy. Khi áp dụng các hàm này vào thực tế, tùy vào loại dữ liệu đầu vào, ta sẽ chọn hàm kernel phù hợp. Kết quả thực nghiệm cũng đã chứng tỏ rằng phương pháp đề xuất có hiệu năng cao hơn so với những phương pháp khác đối với những dữ liệu của người và loài giun tròn. Hơn nữa, phương pháp đề xuất chỉ sử dụng dữ liệu về chuỗi gen mà không cần những dữ liệu khác nên rất đơn giản và dễ dàng khi tính toán, cài đặt, cũng như không phức tạp ở khâu chuẩn bị dữ liệu. Điều này giúp cho phương pháp này có thể áp dụng vào nhiều bài toán thực tế khác trong tương lai.

Kết quả của được luận văn được tóm tắt trong bài báo “Enhancer prediction using distance aware kernels”. Bài báo đã được chấp nhận đăng tại kỷ yếu hội nghị RIVF 2013 sẽ diễn ra vào tháng 11/2013.

# TÀI LIỆU THAM KHẢO

1. BE. Boser, IM. Guyon, VN. Vapnik, (1992), “*A training algorithm for optimal margin classifiers*”*,* trong *“Proceedings of the Fifth Annual Workshop on Computational Learning Theory”.* Association for Computing Machinery (ACM), New York.
2. D. Carter, L. Chakalova, CS. Osborne, Y. Dai, P. Fraser, (2002), *“Long-range chromatin regulatory interactions in vivo”*. Nat Genet 32: 623–626.
3. T. Fawcett, (2005). *“An introduction to ROC analysis”,* Pattern Recognition Letters, Vol. 27, pp: 861-874.
4. J. Göke, MH. Schulz, J. Lasserre, M. Vingron, (2012), “*Estimation of Pairwise Sequence Similarity of Mammalian Enhancers with Word Neighbourhood Counts. Bioinformatics*”, Vol. 28(5), pp: 656-663.
5. B. Langmead, C. Trapnell, M. Pop, SL. Salzberg, (2009), *“Ultrafast and memory-efficient alignment of short DNA sequences to the human genome”*, Genome Biol 10: R25.
6. D. Lee, R. Karchin and MA. Beer, (2011), “*Discriminative prediction of mammalian enhancers from DNA sequence*”, Genome research, Vol 21(12), pp: 2167-2180.
7. C. Leslie, E. Eskin, WS. Noble, (2002), *“The spectrum kernel: A string kernel for SVM protein classification”*. Pac Symp Biocomput 7, pp: 564–575.
8. C. Leslie, E. Eskin, A. Cohen, J. Weston, WS. Noble, (2003), *“Mismatch string kernels for discriminative protein classification”*, Bioinformatics 20, pp: 467–476.
9. CX. Ling, H. Jin, Z. Harry, (2003), *“AUC: a Statistically Consistent and more Discriminating Measure than Accuracy”*, IJCAI'03 Proceedings, pp: 519-524.
10. CG. Palii, C. Perez-Iratxeta, Z. Yao, Y. Cao, F. Dai, J. Davison, H. Atkins, D. Allan, FJ. Dilworth, R. Gentleman, (2011). *“Differential genomic targeting of the transcription factor TAL1 in alternate haematopoietic lineages”,* EMBO J.Vol 30, pp: 494–509.
11. SJ. Schultheiss, W. Busch, JU. Lohmann, O. Kohlbacher, G. Rätsch, (2009), “*KIRMES: Kernel-based identification of regulatory modules in euchromatic sequences. Bioinformatics*”, Vol 25(16), pp: 2126-2133.
12. S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, V. Franc. (2010), *“The SHOGUN Machine Learning Toolbox. J. Mach. Learn”,* Res. Vol. 11, pp: 1799-1802.
13. E. Ukkonen. (1995), *“On-line construction of suﬃx trees”*, Algorithmica, 14, pp: 249–260.
14. VN. Vapnik (1995), “*The nature of statistical learning theory”*, Springer, New York.
15. VN. Vapnik, A. Lerner. (1963), *“Pattern recognition using generalized portrait method”*, Automation and Remote Control, 24, pp: 774-780.
16. VN. Vapnik, A. Chervonenkis, (1964), *“A note on one class of perceptrons”*, Automation and Remote Control, 25.
17. MP. Verzi, H. Shin H, HH. He, R. Sulahian, CA. Meyer, RK. Montgomery, JC. Fleet, Brown M, Liu XS, and Shivdasani RA, (2010), *“Differentiation-Specific Histone Modifications Reveal Dynamic Chromatin Interactions and Partners for the Intestinal Transcription Factor CDX2”*, Developmental Cell, Vol 19, pp: 713–726.
18. ST. Whiteside, S. Goodbourn, (1993), *“Signal transduction and nuclear targeting: regulation of transcription factor activity by subcellular localization”*, Journal of Cell Science 104 (4), pp: 949–55.
19. JO. Yanez-Cuna, HQ. Dinh, EZ. Kvon, (2012), *“Uncovering cis-regulatory sequence requirements for context specific transcription factor binding”*, Genome research, Vol 22, pp: 2018-2030.
20. Y. Zhang, T. Liu, CA. Meyer, J. Eeckhoute, DS. Johnson, BE. Bernstein, C. Nussbaum, RM. Myers, M. Brown, W. Li, (2008), *“Model-based analysis of ChIP-Seq (MACS)”*. Genome Biol 9, pp: R137.
21. M. Zhong, W. Niu, ZJ. Lu, M. Sarov, JI. Murray, J. Janette, D. Raha, KL. Sheaffer, HYK. Lam, E. Preston, (2010), *“Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response”*. PLoS Genet 6: e1000848.