ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

PHẠM VĂN HIẾU

DỰ ĐOÁN TƯƠNG TÁC PROTEIN – PROTEIN SỬ DỤNG KỸ THUẬT KHAI PHÁ DỮ LIỆU

Ngành: CÔNG NGHỆ THÔNG TIN
Chuyên ngành: Hệ thống thông tin
Mã số: 60480104

TÓM TẮT LUẬN VĂN THẠC SĨ
NGÀNH CÔNG NGHỆ THÔNG TIN

Hà Nội - 2017

MỤC LỤC

CHƯƠNG 1 : MỞ ĐẦU

1.1 LÝ DO CHỌN ĐỀ TÀI

Protein là thành phần quan trọng trong tế bào và cơ thể sống. Tương tác protein – protein là cách để protein thể hiện được chức năng sinh học. Vì vậy hiểu về các tương tác protein – protein (PPI) sẽ giúp ta biết hơn về các chức năng protein, và tìm được vai trò của các protein mới.

Vào thời điểm bắt đầu nghiên cứu tương tác protein – protein, các nhà khoa học sử dụng phương pháp hóa sinh. Tuy nhiên phương pháp này tốn chi phí, nhiều khi khó thực hiện. Vì vậy yêu cầu đặt ra là dự đoán PPI bằng khai phá dữ liệu như là sự bổ sung cho các phương pháp thực nghiệm. Đó cũng là lý do tôi quyết định chọn đề tài **"Dự đoán tương tác protein – protein sử dụng kỹ thuật khai phá dữ liệu"**.

1.2 MỤC TIÊU ĐỀ TÀI

Trong khuôn khổ luận văn, tôi trình bày một phương pháp tính toán cho dự đoán tương tác PPI theo hướng áp dụng thuật toán phân loại tổng hợp, hay là sự kết hợp mô hình các bộ phân loại đơn lẻ yếu hơn thành một mô hình mạnh, nhằm đạt được hiệu quả phân loại tối ưu. Kết quả đó cũng là mục tiêu đề tài hướng tới.

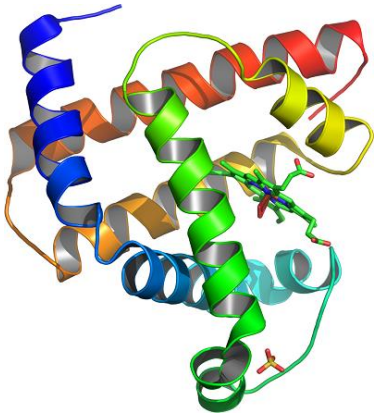
Để đạt được mục tiêu, các công việc tôi thực hiện trong luận văn này là: Nghiên cứu cơ sở lý thuyết khái niệm về protein, cấu trúc protein phục vụ cho việc trích xuất thuộc tính; Nghiên cứu cơ sở lý thuyết về các kỹ thuật khai phá dữ liệu (nói chung) và kỹ thuật phân lớp dữ liệu (nói riêng), làm cơ sở cho xây dựng chương trình thực nghiệm.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

Chương 2 trình bày cơ sở lý thuyết, bao gồm các thông tin giới thiệu về các khái niệm trong sinh học liên quan đến protein, cấu trúc protein; Các khái niệm khai phá dữ liệu nền tảng liên quan đến kỹ thuật phân lớp dữ liệu, nhằm củng cố kiến thức và tạo tiền đề áp dụng giải quyết bài toán "Dự đoán tương tác protein – protein sử dụng kỹ thuật khai phá dữ liệu".

2.1 CÁC KHÁI NIỆM LIÊN QUAN ĐẾN PROTEIN

Protein là đại phân tử, phức tạp và có vai trò quan trọng trong tế bào (nói riêng) và cơ thể sống (nói chung). Chúng được tạo thành từ hàng trăm hoặc hàng ngàn các đơn vị nhỏ hơn được gọi là các amino acid. Protein được tạo ra bởi sự liên kết của hai hoặc nhiều polypeptide, là chuỗi được ghép từ các amino acid liên kết với nhau, được xếp thành một cấu trúc đặc biệt cho mỗi một protein cụ thể [1].



Hình 2-1: Minh họa cấu trúc 3D một protein [2]

2.1.1 Cấu trúc Protein

Protein được hình thành do các amino acid liên kết lại với nhau bởi các liên kết peptide tạo ra chuỗi polypeptide. Amino acid được cấu tạo bởi 3 thành phần : nhóm amin ($-NH_2$), nhóm caboxyl ($-COOH$) và cuối cùng là

nguyên tử cacbon trung tâm đính với 1 nguyên tử hydro và nhóm biển đổi R quyết định tính chất của amino acid.

Các loại cấu trúc protein gồm có: Cấu trúc sơ cấp, cấu trúc bậc hai, cấu trúc bậc ba, cấu trúc bậc bốn [3]. Cụ thể: cấu trúc sơ cấp là cấu trúc mô tả thứ tự các amino acid liên kết với nhau, cấu trúc bậc 2 là cấu trúc đề cập đến việc xoắn hoặc gấp một chuỗi polypeptide cho protein hình dạng 3D, cấu trúc bậc ba là cấu trúc đề cập đến cấu trúc 3-D toàn diện của chuỗi polypeptide của một protein và cấu trúc bậc bốn đề cập đến cấu trúc của một phân tử protein được hình thành bởi các tương tác giữa nhiều chuỗi polypeptide.

2.1.2 Chức năng của Protein

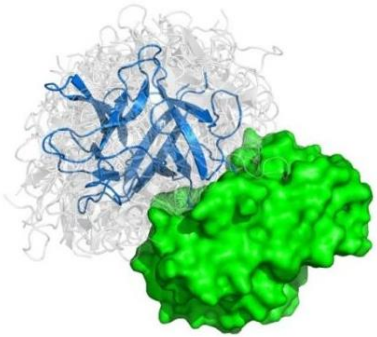
Protein đảm nhiệm các chức năng liên quan đến toàn bộ hoạt động sống của tế bào, quy định các tính trạng và các tính chất của cơ thể sống. Cụ thể :

Bảng 2.1: Bảng chức năng các loại protein cơ bản [4]

Loại Protein	Chức năng
Protein cấu trúc	Cấu trúc, nâng đỡ
Protein Enzyme	Xúc tác sinh học : chọn lọc các phản ứng sinh học
Protein Hormone	Điều hòa các hoạt động sinh lý
Protein vận chuyển	Vận chuyển các chất
Protein vận động	Tham gia chức năng vận động của tế bào, cơ thể
Protein thụ quan	Cảm nhận, đáp ứng các kích thích của môi trường
Protein dự trữ	Dự trữ chất dinh dưỡng

2.1.3 Định nghĩa quan hệ tương tác protein – protein (PPI)

Tương tác protein – protein là quá trình tác động qua lại giữa các protein với nhau trong tế bào. Các loại tương tác protein – protein bao gồm : Tương tác ổn định, tương tác tạm thời, tương tác mạnh, tương tác yếu.



Hình 2-2: Minh họa tương tác protein – protein *[5]*

2.1.4 Tầm quan trọng của tương tác protein – protein

Sự tương tác của protein – protein là nền tảng cơ bản của các chức năng của tế bào và khi quá trình tương tác này bị tổn hại sẽ gây ảnh hưởng trực tiếp đến cơ thể sống [6].

2.2 KHÁI NIỆM CƠ BẢN VỀ KHAI PHÁ DỮ LIỆU

2.2.1 Định nghĩa về khai phá dữ liệu

Khai phá dữ liệu là một lĩnh vực đa ngành, dựa trên kết quả từ trí thông minh nhân tạo và các lĩnh vực khác. Nó cho phép chương trình "học tập" và tự động cải thiện năng lực từ kinh nghiệm tích lũy [7]. Theo cách sử dụng được chia làm 2 loại chính: Thuật toán học máy – có giám sát (phân lớp), và thuật toán học máy – không giám sát (phân cụm).

2.2.2 Định nghĩa về học có giám sát

Thuật toán học có giám sát lấy một tập dữ liệu đầu vào đã biết kết quả đầu ra, và xây dựng một mô hình để tạo ra các dự đoán hợp lý cho kết quả của một dữ liệu mới.

2.2.3 Khái niệm về thuật toán phân loại trong học có giám sát

Phân lớp (loại) là cách thức xử lý xếp các mẫu dữ liệu vào một lớp đã định nghĩa trước. Các mẫu dữ liệu được xếp về các lớp dựa vào giá trị của các

thuộc tính của mẫu dữ liệu đó. Các thuật toán phân loại tiêu biểu gồm có: Cây quyết định, mạng Bayes, SVM, …

2.2.4 Bài toán phân lớp

Một bài toán phân lớp bao gồm 3 bước sau: Chuẩn bị dữ liệu, xây dựng mô hình từ tập dữ liệu huấn luyện, kiểm tra và đánh giá kết quả.

2.2.5 Tổng quan về một số thuật toán phân lớp cơ bản

a, Mạng Bayes

Phương pháp phân lớp dựa vào thống kê theo định lý của Bayes. Hiệu quả trong nhiều ứng dụng liên quan, bao gồm phân lớp văn bản, chẩn đoán y tế và quản lý hiệu năng hệ thống [8].

b, Cây quyết định

Cây quyết định (Decision Tree) là cây phân cấp có cấu trúc dùng phân lớp các đối tượng dựa vào dãy các luật. Cơ sở toán học của cây quyết định là thuật toán tham lam. Ứng dụng trong nhiều lĩnh vực như tài chính, tiếp thị, kỹ thuật và y học [9].

c, Support Vector Machine (SVM)

SVM là một thuật toán phân loại nhị phân, SVM nhận dữ liệu vào và phân loại chúng vào hai lớp khác nhau.

2.2.6 Kết hợp các bộ phân loại

Phương pháp phân loại tổng hợp (ensemble) là mô hình tổng hợp từ nhiều mô hình con yếu (weaker model) được huấn luyện độc lập. Kết quả dự đoán cuối cùng dựa trên việc "bỏ phiếu" kết quả của từng mô hình con cho kết quả đầu ra.

Phương pháp phân loại tổng hợp thường tạo ra các dự đoán chính xác hơn so với các phương pháp phân loại đơn lẻ.

2.2.7 Một số phương pháp kết hợp các bộ phân loại cơ bản

a, Phương pháp Bagging

Giới thiệu: Mô hình Bagging được Breiman đề xuất năm 1996 nhằm làm giảm lỗi variance nhưng không làm tăng lỗi bias quá nhiều.

Mô hình hoạt động: Tạo ra các bộ phân loại từ các tập mẫu con ngẫu nhiên, chấp nhận lặp từ tập mẫu dữ liệu ban đầu, và một thuật toán học máy tương ứng. Các bộ phân loại sẽ được kết hợp bằng phương pháp biểu quyết theo số đông.

b, Phương pháp Boosting

Giới thiệu: Phương pháp *Boosting* được giới thiệu lần đầu bởi Freund & Schapire (1997), kỹ thuật này phù hợp cho vấn đề phân loại 2 lớp.

Mô hình hoạt động: Là thuật toán học quần thể bằng cách xây dựng nhiều thuật toán học cùng lúc và kết hợp chúng lại. Ý tưởng chính của giải thuật là lặp lại quá trình học của một bộ phân lớp yếu nhiều lần và sau mỗi lần gán trọng số ưu tiên cho mẫu dự đoán sai.

c, Phương pháp Random Forest

Giới thiệu: *Random Forest* được đề xuất bởi Breiman (2001). Nó cho độ chính xác cao và độ chịu nhiễu tốt.

2.2.8 Đánh giá mô hình phân lớp

a, Khái niệm

Mô hình phân lớp cần được đánh giá để xem có hiệu quả không và để so sánh khả năng của các mô hình. Hiệu năng của một mô hình thường được đánh giá dựa trên tập dữ liệu kiểm định (test data).

b, Độ đo Accuracy (độ chính xác)

Cách đánh giá này tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm định.

c, Confusion matrix (ma trận nhầm lẫn)

Đánh giá được các giá trị : dương tính đúng, dương tính sai, âm tính đúng, âm tính sai, quy ước ký hiệu : TP, FP, TN, FN. Gọi *accucary* là độ chính xác của mô hình sẽ được tính như sau:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.1)$$

d, Precision & recall (độ chính xác & độ bao phủ)

Precision đối với lớp c_i:

$$Precision = \frac{TP}{TP+FP} \tag{2.2}$$

Recall đối với lớp c_i:

$$Recall = \frac{TP}{TP+FN} \tag{2.3}$$

e, Độ đo F

Tiêu chí đánh giá là sự kết hợp của 2 tiêu chí đánh giá *Precision* và *Recall* theo công thức:

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.4}$$

CHƯƠNG 3 DỰ ĐOÁN TƯƠNG TÁC PROTEIN - PROTEIN

Như đã đề cập ở giới thiệu mở đầu, việc nghiên cứu dự đoán tương tác protein – protein trong tin sinh học có ý nghĩa đặc biệt quan trọng trong việc tìm hiểu chức năng của protein mới, và ảnh hưởng của các hoạt động tương tác này tới tế bào trong cơ thể sống. Nội dung của bài toán trong nghiên cứu này là: đầu vào là tập các dữ liệu quan hệ tương tác giữa các cặp protein – protein đã được gán nhãn theo 2 lớp (dương tính – có tương tác, âm tính – không tương tác), qua thuật toán phân loại tổng hợp xây dựng một mô hình để kiểm chứng kết quả kiểm định và tính toán độ chính xác của mô hình thuật toán.

3.1 MÔ HÌNH DỰ ĐOÁN TƯƠNG TÁC PROTEIN – PROTEIN

Để giải quyết bài toán dự đoán tương tác protein – protein, nhiều phương pháp tin sinh học đã được đề xuất, như: Sử dụng thông tin cấu trúc 3D của protein và tạo ra thuật toán PrePPI để dự đoán PPI ở người và nấm men [*Zhang & cộng sự,2012*][10]. Phương pháp mở rộng mỗi polymerase: thu thập các chuỗi polypeptide ngắn liên tục xảy ra giữa các cặp tương tác protein đã biết [*Pitre & cộng sự,2006*][11]. Sử dụng hệ thống học máy k-nearest neighbors dựa trên thành phần amino acid giả và lựa chọn thuộc tính [*Liu & cộng sự,2009*][12]. Trích xuất thuộc tính genomic/proteomic và lựa chọn đặc trưng dự đoán PPI bằng cách sử dụng thuật toán VSM [*Urquiza & cộng sự,2011*][13]. Sử dụng công cụ tìm kiếm cho việc truy xuất dữ liệu cơ sở dữ liệu tương tác gen để dự đoán các PPI trên cơ sở hợp nhất và hình thành gen [*Szklarczyk & cộng sự,2011*][14].

Các phương pháp đề xuất khác nhau trong thuật toán trích xuất đặc trưng và xây dựng mô hình. Đối với trích xuất thuộc tính, nhiều phương pháp khai thác thông tin đã được đề xuất. Ví dụ: Phương pháp trích xuất thuộc tính 188-D dựa vào tính chất hóa lý và sự phân bố các amino acid của protein [*Cai & cộng sự*][15], phương pháp trích xuất thuộc tính 20-D từ

chuỗi protein trên cơ sở của vị trí protein – ma trận điểm riêng biệt [*Zou & cộng sự*][16], phương pháp n-gram, các công cụ trích xuất đặc trưng đặc biệt như Pse-in-One, RepDNA, RepRNA…

Về xây dựng mô hình, các phương pháp đề xuất có hai hướng xây dựng mô hình phân lớp: Mô hình phân loại đơn lẻ, mô hình phân loại tổng hợp (ensemble). Ưu điểm của mô hình này so với các mô hình phân loại truyền thống là có hiệu suất dự đoán tốt hơn, và lỗi dự đoán thấp hơn, nhưng nhược điểm là chi phí xây dựng phải bỏ ra cao hơn.

Trong luận văn này, tôi nghiên cứu và xây dựng một phương pháp tính toán dự đoán tương tác protein – protein dựa trên phương pháp Bagging của Breiman và cộng sự năm 1996, phương pháp AdaBoost của Freund và cộng sự năm 1997 và phương pháp Random Forest của Breiman và cộng sự năm 2001. Phương pháp đề xuất gồm 3 điểm chính: Xây dựng số liệu, khai thác thuộc tính, phân loại.

- Xây dựng số liệu: sử dụng bộ số liệu dương tính, bộ số liệu âm tính đã được kiểm chứng xác thực qua các thực nghiệm sinh học.
- Khai thác thuộc tính: sử dụng 2 phương pháp là n-gram, và MLD để xây dựng bộ thuộc tính căn cứ vào tần suất của các amino acid có mặt trong protein. Sau đó áp dụng phương pháp lựa chọn thuộc tính để tạo ra một tập hợp các thuộc tính được tối ưu hóa.
- Phân loại: sử dụng mô hình phân loại tổng hợp, cụ thể là 3 bộ phân loại Bagging, AdaBoost và Random Forest vào tính toán dự đoán tương tác protein – protein và so sánh hiệu quả thu được với các bộ phân loại đơn lẻ cơ sở tương ứng.

Sau thực nghiệm, các kết quả cho thấy hiệu quả tốt của mô hình được xây dựng trong dự đoán PPI.

3.2 XÂY DỰNG MÔ HÌNH THỰC NGHIỆM

3.2.1 Xây dựng bộ dữ liệu

Dự đoán tương tác PPI thuộc bài toán phân loại nhị phân, vì vậy chúng ta cần xây dựng các tập dữ liệu dương tính và âm tính. Trong luận văn này,

tập dữ liệu dương tính được thu thập từ nguồn dữ liệu DIP (Database of Interacting Protein) trên Internet, có địa chỉ trang web tại: http://dip.doe-mbi.ucla.edu/dip/Main.cgi [17]. Tập dữ liệu âm tính ta có được từ tích lũy kết quả các thực nghiệm. Tên bộ dữ liệu âm tính là Negatome, được lấy về từ địa chỉ trang web http://mips.helmholtz-muenchen.de/proj/ppi/negatome/ [18].

Để đảm bảo tỷ lệ dữ liệu dương tính cân bằng với dữ liệu âm tính theo tỷ lệ 1:1.

3.2.2 Trích xuất thuộc tính/đặc trưng

Trong nghiên cứu này tôi sử dụng hai phương pháp để so sánh sự hiệu quả là n-gram và MLD.

Phương pháp n-gram được tạo ra từ thuật toán ngôn ngữ tự nhiên [19]. Các n-gram được sử dụng để mã hóa protein xây dựng bằng cách tính tần số xuất hiện của n chuỗi amino acid theo 1-gram,2-gram,3-gram và nhân 3 thuộc tính. Kết quả tạo ra một vector đặc trưng có 8420 chiều.

Phương pháp MLD là phương pháp được đề xuất để biến đổi chuỗi trình tự amino acid trong protein thành các vector đặc trưng bằng cách sử dụng một lược đồ mã hóa nhị phân. Mỗi một chuỗi trình tự protein có 567 thuộc tính. Cặp protein PPI (hoặc PPNI) được kết hợp để tạo ra vector đặc trưng cuối cùng bằng cách ghép 2 vector 567 chiều của mỗi protein, sinh ra một vector 1134 chiều đại diện cho cặp protein đó [20].

3.2.3 Lựa chọn thuộc tính/đặc trưng

Lựa chọn các thuộc tính có độ quan trọng cao trong bộ dữ liệu thuộc tính ban đầu là cần thiết. Trong luận văn này, phương pháp MRMD được sử dụng. Mục tiêu chính của phương pháp là tìm kiếm thuộc tính có sự liên quan cao giữa tập hợp thuộc tính và lớp đích, và tính thừa thấp của bộ thuộc tính. Hệ số tương quan Pearson được sử dụng để đo lường sự liên quan. Ba loại hàm khoảng cách (ED, khoảng cách cosine, và hệ số Tanimoto) được sử dụng để tính toán sự thừa. Thuộc tính với tổng lớn hơn của sự liên quan và khoảng cách được chọn làm bộ thuộc tính cuối cùng.

Đây là bộ dữ liệu dùng làm đầu vào cho việc phân loại và đánh giá kết quả phân loại. Ta chia tập thuộc tính đặc trưng này theo phương pháp k-fold cross validation, chia dữ liệu thành 10 phần có kích thước bằng nhau, lấy lần lượt 1 phần dữ liệu test và 9 phần dữ liệu còn lại làm thực nghiệm.

3.2.4 Phân loại đặc trưng

Trong nghiên cứu này, ta thực nghiệm xử lý phân loại theo hướng sử dụng thuật toán phân loại tổng hợp với 3 bộ phân loại là: AdaBoostM1, Baggingvà Random Forest để làm rõ ưu điểm so với các thuật toán phân loại đơn lẻ sử dụng đối chứng trong nghiên cứu là Decision Stump, REPTree và Random Tree.

Trong đó bộ phân loại tổng hợp Bagging sử dụng thuật toán cơ bản là REPTree, với dữ liệu huấn luyện là n_1 mẫu huấn luyện và n_2 mẫu kiểm định với tỉ lệ $n_1 : n_2 = 9 : 1$. Từ n_1 mẫu huấn luyện ta tạo ra k tập dữ liệu huấn luyện con, trong đó các mẫu huấn luyện được chọn ngẫu nhiên và có thể lặp. Tạo tương ứng các mô hình với mỗi tập huấn luyện trong k tập huấn luyện con cùng thuật toán REPTree và kết quả cuối cùng thông qua biểu quyết theo số lượng kết quả các mô hình con.

Thứ hai, bộ phân loại tổng hợp AdaBoostM1 trong nghiên cứu này sử dụng thuật toán cơ bản là Decision Stump (cây quyết định một cấp). Cách thực hiện giải thuật AdaBoostM1 là thực hiện xây dựng lặp lại các mô hình cơ bản trên tập dữ liệu huấn luyện có trọng số thay đổi sau mỗi lần traning, theo hướng: ở vòng training trước, mẫu dữ liệu nào dự đoán đúng sẽ gán trọng số thấp đi, mẫu dữ liệu nào dự đoán sai sẽ được gán trọng số cao hơn, mục đích là ở vòng training sau mẫu dữ liệu sai này sẽ có vai trò quan trọng hơn trong việc phân loại. Kết quả cuối cùng tính bằng trung bình kết quả các mô hình con.

Thứ ba, bộ phân loại Random Forest trong nghiên cứu này sử dụng thuật toán cơ bản là Random Tree. Cách thực hiện giải thuật là xây dựng lặp lại k Random Tree. Sau đó từ các mô hình lặp lấy ra các kết quả dự đoán tương ứng, bỏ phiếu chọn ra phương pháp được bình chọn nhiều nhất làm kết quả dự đoán cuối cùng.

CHƯƠNG 4 KẾT QUẢ THỰC NGHIỆM VÀ KẾT LUẬN

4.1 CHƯƠNG TRÌNH CÀI ĐẶT

4.1.1 Yêu cầu cấu hình

Chương trình thực nghiệm dự đoán tương tác protein - protein sử dụng kỹ thuật khai phá dữ liệu được lập trình bằng ngôn ngữ Java. Yêu cầu cần có để chạy được chương trình là:

- Môi trường java tối thiểu 1.6
- Phần cứng:
    - CPU Dual-core+, RAM 8G+ (cho trường hợp chạy lựa chọn thuộc tính/đặc trưng sau trích xuất thuộc tính/đặc trưng n-gram)
    - CPU Dual-core+, RAM 4G+ (cho trường hợp chạy lựa chọn thuộc tính/đặc trưng sau trích xuất thuộc tính/đặc trưng MLD)
- Client chạy ứng dụng phải là máy cài hệ điều hành Windows.

4.1.2 Cài đặt

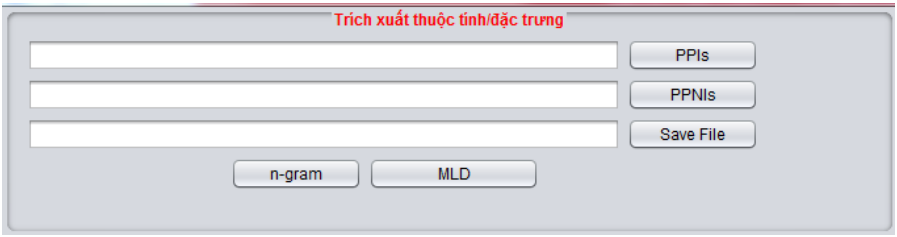
a, Chuẩn bị dữ liệu

Dữ liệu dương tính: Tải về từ nguồn DIP có địa chỉ tại: http://dip.doe-mbi.ucla.edu/dip/Main.cgi. Số lượng các cặp PPI lấy ngẫu nhiên 6445 cặp.

Dữ liệu âm tính: Tải về từ nguồn có địa chỉ tại http://mips.helmholtz-muenchen.de/proj/ppi/negatome/. Số lượng PPNI lấy ngẫu nhiên: 6445 cặp.

Dữ liệu có dạng tệp nén chứa các file đuôi *.fasta, trong mỗi file có dữ liệu thô chứa thông tin về cặp protein.

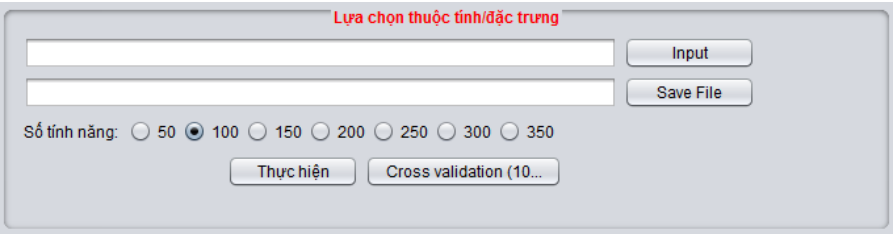
b, Trích xuất thuộc tính/đặc trưng



Hình 4-1: Giao diện chức năng trích xuất thuộc tính/đặc trưng

Nhấn button [PPIs], chọn thư mục chứa các cặp protein tương tác. Nhấn button [PPNIs], chọn thư mục chứa các cặp protein không tương tác. Nhấn button [Save File], chọn thư mục lưu file kết quả trích xuất. Nhấn button [n-gram] để thực hiện trích xuất thuộc tính/đặc trưng theo phương pháp trích xuất n-gram, hoặc nhấn button [MLD] thực hiện trích xuất thuộc tính/đặc trưng theo phương pháp trích xuất MLD.

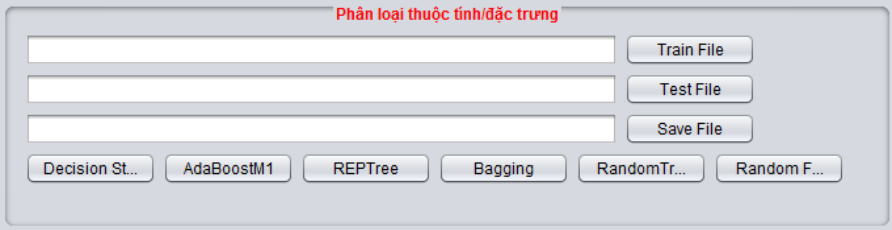
c, Lựa chọn thuộc tính/đặc trưng



Hình 4-2: Giao diện chức năng lựa chọn thuộc tính/đặc trưng

Nhấn [Input] chọn file dữ liệu trích xuất được ở bước *b, Trích xuất thuộc tính/đặc trưng* làm đầu vào. Nhấn [Save File] chọn thư mục cần lưu file kết quả lựa chọn thuộc tính/đặc trưng. Nhấn [Thực hiện] thực hiện gọi hàm lựa chọn tính năng/đặc trưng. Nhấn [Cross validation (10-fold)] thực hiện chia file kết quả sau khi lựa chọn thuộc tính/đặc trưng thành 10 phần có kích thước bằng nhau, mỗi phần lần lượt là dữ liệu kiểm định và 9 phần còn lại làm dữ liệu huấn luyện.

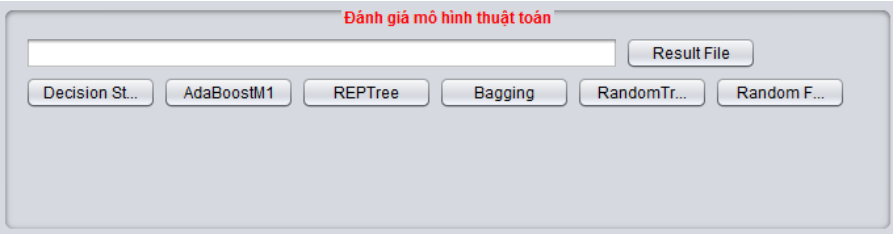
d, Phân loại đặc trưng



Hình 4-3: Giao diện chức năng Phân loại thuộc tính/đặc trưng

Nhấn button [Train File] chọn file dữ liệu huấn luyện. Nhấn button [Test File] chọn file dữ liệu kiểm định. Nhấn button [Save File] chọn thư mục lưu file kết quả phân lớp từ đầu vào là dữ liệu kiểm định. Nhấn 1 trong 6 button [Decision Stump], [AdaBoostM1], [REPTree], [Bagging], [Random Tree], hoặc [Random Forest] thực hiện phân loại đặc trưng tương ứng cho mỗi thuật toán: thuật toán phân loại đơn lẻ Decision Stump, REPTree và Random Tree, thuật toán phân loại tổng hợp AdaBoostM1, Bagging và Random Forest.

e, Độ đo đánh giá



Hình 4-4: Giao diện chức năng Đánh giá mô hình thuật toán

Nhấn button [Result File] chọn file kết quả vừa thu được qua bước phân loại thuộc tính/đặc trưng.

Nhấn 1 trong 6 button [Decision Stump], [AdaBoostM1], [REPTree], [Bagging], [Random Tree], hoặc [Random Forest] thực hiện gọi hàm tính toán độ đo tương ứng cho mỗi thuật toán phân loại Decision Stump, REPTree, Random Tree, AdaBoostM1, Bagging hoặc Random Forest.

4.2 KẾT QUẢ DỰ ĐOÁN TƯƠNG TÁC PROTEIN - PROTEIN

Tiến hành thực nghiệm với 6 thuật toán, 3 thuật toán phân loại tổng hợp là AdaBoostM1, Bagging và Random Forest, 3 thuật toán phân loại đơn lẻ là Decision Stump, REPTree, và RandomTree. Như đã đề cập ở phần 3.3.3. Lựa chọn thuộc tính/đặc trưng, ta áp dụng phương pháp k-fold cross validation, bằng cách xây dựng một hàm chia file dữ liệu ban đầu thành 10 phần bằng nhau. Lấy lần lượt mỗi phần làm dữ liệu kiểm định và 9 phần còn lại làm dữ liệu huấn luyện, ta thu được 10 bộ dữ liệu. Mỗi bộ dữ liệu có 2 file: file dữ liệu huấn luyện và file dữ liệu kiểm định với tỉ lệ 9:1. Để biểu diễn kết quả ngắn gọn và tường minh, trong nghiên cứu sử dụng độ đo F để hiển thị trên chương trình tương ứng với các thuật toán trên mỗi bộ dữ liệu. Ta hiển thị kết quả theo 2 hướng: sử dụng thuật toán trích xuất thuộc tính/đặc trưng n-gram và thuật toán trích xuất thuộc tính/đặc trưng MLD. Sau bước trích xuất thuộc tính/đặc trưng, ta lựa chọn thuộc tính với số thuộc tính lựa chọn nhỏ hơn số thuộc tính ban đầu. Trong nghiên cứu này, ta thực hiện lựa chọn thuộc tính với số thuộc tính rút gọn là 100 thuộc tính và so sánh kết quả phân loại đặc trưng của tập dữ liệu ban đầu và tập dữ liệu đã rút gọn thuộc tính.

Bảng 4.1: Kết quả thực nghiệm phương pháp trích xuất thuộc tính MLD, không giảm chiều số thuộc tính (1134 thuộc tính)

Độ đo F (%)	Decision Stump	AdaBoost	REPTree	Bagging	Random Tree	Random Forest
S1	69,72	71,09	82,66	87,33	79,85	87,88
S2	70,04	71,30	79,65	87,55	79,48	86,39
S3	66,06	67,92	78,96	84,3	76,06	83,80
S4	65,27	67,59	79,47	85,54	78,31	84,57
S5	69,88	73,36	78,75	85	75,92	84,60

S6	68,03	68,45	76,74	86,25	78,50	85,76
S7	67,41	75,17	81,05	87,62	78,54	85,43
S8	67,28	67,21	77,26	85,58	77,91	84,09
S9	64,01	70,46	82,15	87,42	76,65	85,56
S10	69,90	74,55	79,60	86,33	76,92	85,82
TB	67,76	70,71	79,63	86,29	77,81	85,39

Bảng 4.2: Kết quả thực nghiệm phương pháp trích xuất thuộc tính MLD, giảm chiều còn 100 thuộc tính

ĐĐ F (%)	Decision Stump	AdaBoost	REPTree	Bagging	Random Tree	Random Forest
S1	68,81	69,82	76,96	82,91	76,41	81,52
S2	67,39	70,06	75,88	82,10	74,74	82,52
S3	67,47	67,71	73,09	82,80	73,12	81,85
S4	67,09	70,17	75,38	84,15	73,57	82,07
S5	68,35	69,63	75,93	80,51	74,49	80,51
S6	67,90	68,84	73,88	81,72	72,66	80,28
S7	69,69	72,42	76,59	82,94	76,17	82,81
S8	67,13	67,80	74,68	81,80	76,39	81,66
S9	66,42	66,67	72,86	82,64	75,04	81,46
S10	68,46	71,45	73,92	82,80	73,36	81,07
TB	**67,87**	**69,46**	**74,92**	**82,44**	**74,60**	**81,58**

Bảng 4.3: Kết quả thực nghiệm phương pháp trích xuất thuộc tính n-gram, không giảm chiều thuộc tính

Độ đo F (%)	Decision Stump	AdaBoost	REPTree	Bagging	Random Tree	Random Forest
S1	67,45	73,83	77,59	85,60	78,41	84,77
S2	70,27	76,99	77,20	85,55	77,82	85,60
S3	68,82	76,12	78,10	86,01	77,53	85,82
S4	70,22	76,29	76,84	86,25	81,50	85,67
S5	69,65	76,22	78,25	85,37	78,15	84,69
S6	71,40	76,30	79,35	86,55	79,53	86,19
S7	67,55	74,61	78,95	86,06	79,11	86,09
S8	69,02	73,27	79,27	85,74	79,45	84,68
S9	68,98	76,01	81,10	87,54	78	85,35
S10	68,41	73,12	79,21	85,56	77,46	84,71
TB	**69,18**	**75,28**	**78,59**	**86,02**	**78,70**	**85,36**

Bảng 4.4: Kết quả thực nghiệm phương pháp trích xuất thuộc tính n-gram, giảm chiều còn 100 thuộc tính

ĐĐ F (%)	Decision Stump	AdaBoost	REPTree	Bagging	Random Tree	Random Forest
S1	67,45	73,89	75,14	80,52	77,07	81,54
S2	70,27	76,01	80,03	81,99	78,03	82,54
S3	68,92	76,15	75,52	80,91	74,56	82,37

S4	70,12	76,26	78,86	81,74	78,54	82,58
S5	69,62	76,20	77,06	82,33	77,09	82,74
S6	71,39	76,18	79,72	82,76	77,15	83,04
S7	67,50	74,57	78,97	81,27	77,69	82,89
S8	69,07	73,57	79,64	82,79	77,69	82,10
S9	68,97	75,73	77,96	80,72	77,38	81,86
S10	68,39	72,74	77,05	80,35	76,08	81,41
TB	69,17	75,13	78,00	81,54	77,13	82,31

4.3 NHẬN XÉT

Về tổng quan ta nhận thấy các mô hình phân loại đơn lẻ có độ chính xác trong kiểm định thấp hơn nhiều so với các mô hình phân loại tổng hợp tương ứng mà sử dụng mô hình phân loại đơn lẻ đó làm cơ sở. Cụ thể, hiệu quả dự đoán của mô hình thuật toán Decision Stump thấp hơn mô hình thuật toán AdaBoostM1, hiệu quả dự đoán mô hình thuật toán REPTree thấp hơn mô hình thuật toán Bagging, và hiệu quả dự đoán mô hình thuật toán Random Tree thấp hơn mô hình thuật toán Random Forest.

Tiếp theo, nhận xét về hiệu quả dự đoán phân lớp khi sử dụng phương pháp lựa chọn thuộc tính/đặc trưng MRMD để giảm chiều dữ liệu. Ta thấy các thuật toán phân lớp sử dụng đầu vào là tập vector thuộc tính rút gọn có chi phí giảm đáng kể so với sử dụng đầu vào giữ nguyên là tập vector thuộc tính ban đầu, nhưng hiệu quả dự đoán giảm xuống, dao động trong khoảng [1;4](%) (theo độ đo F). Mức hiệu quả dự đoán bị giảm trên có thể chấp nhận được so với chi phí chạy chương trình tiết kiệm được.

So sánh giữa hai phương pháp trích xuất thuộc tính/đặc trưng là n-gram và MLD. Hiệu quả cho 2 phương pháp trích xuất thuộc tính/đặc trưng là tương đương nhau, chi phí bỏ ra chạy thuật toán thì phương pháp MLD có chi phí thấp hơn nhiều lần so với phương pháp n-gram. Vì vậy nếu xét tính hiệu quả ta sẽ chọn MLD thay vì n-gram.

So sánh giữa các cặp thuật toán với nhau, ta thấy cặp Decision Stump – AdaBoostM1 có hiệu quả dự đoán thấp hơn 2 cặp còn lại. Hai cặp REPTree – Bagging và Random Tree – Random Forest có hiệu quả dự đoán tương đương nhau, xét chi phí cho thuật toán thì cặp Random Tree – Random Forest có chi phí bỏ ra thấp hơn nhiều lần so với cặp REPTree – Bagging.

Từ những nhận xét trên, ta rút ra kết quả cuối cùng: Phương pháp hiệu quả nhất trong nghiên cứu này cho dự đoán bài toán "Dự đoán tương tác protein – protein sử dụng phương pháp khai phá dữ liệu" là phương pháp phân lớp Random Forest, sử dụng phương pháp trích xuất thuộc tính/đặc trưng MLD và có giảm chiều thuộc tính.

4.4 KẾT LUẬN

Luận văn đã đạt được hai kết quả quan trọng trong quá trình xây dựng chương trình dự đoán tương tác protein - protein sử dụng kỹ thuật khai phá dữ liệu.

Về nghiên cứu tìm hiểu:

- Nghiên cứu khái niệm sinh học liên quan protein, cấu trúc protein
- Nghiên cứu khái niệm khai phá dữ liệu nền tảng liên quan đến kỹ thuật phân lớp dữ liệu
- Tìm hiểu tổng quan về một số thuật toán phân lớp cơ bản
- Tìm hiểu về phương pháp phân loại tổng hợp (ensemble) và một số phương pháp kết hợp các bộ phân loại cơ bản
- Tìm hiểu các khái niệm về đánh giá mô hình phân lớp

Về thực nghiệm:

- Xây dựng được chương trình dự đoán tương tác protein - protein bằng phương pháp phân loại tổng hợp
- Xây dựng được hàm đánh giá và so sánh kết quả thực nghiệm giữa phương pháp phân loại tổng hợp và phân loại đơn lẻ
- Tiến hành thử nghiệm trên nhiều tập dữ liệu ngẫu nhiên khác nhau để đảm bảo tính chính xác khách quan
- Xây dựng giao diện trực quan, dễ dàng sử dụng cho người dùng

Luận văn đã giới thiệu phương pháp áp dụng mô hình phân loại tổng hợp vào nghiên cứu dự đoán tương tác protein - protein. Cũng như chứng minh được về mặt lý thuyết và thực nghiệm rằng phương pháp áp dụng mô hình phân loại tổng hợp này ưu việt hơn giải thuật mô hình phân loại đơn lẻ, có độ chính xác cao hơn và độ ổn định tốt hơn.

4.5 HƯỚNG NGHIÊN CỨU TRONG TƯƠNG LAI

Trong luận văn tôi chưa đi sâu vào tìm hiểu được cách kết hợp các thuật toán con trong thuật toán phân loại tổng hợp. Về ngôn ngữ lập trình vấn đề tối ưu thời gian và hiệu suất xử lý nguồn dữ liệu lớn còn hạn chế, từ đó làm giảm độ chính xác của kết quả thực nghiệm. Vì vậy, trong tương lai, tôi mong muốn được tìm hiểu và áp dụng sâu hơn các cách kết hợp giải thuật đơn lẻ vào mô hình phân loại tổng hợp và thực hiện tối ưu về mặt ngôn ngữ lập trình đảm bảo xử lý dữ liệu lớn một cách nhanh chóng cả về thời gian và hiệu suất xử lý.

TÀI LIỆU THAM KHẢO

[1] H. Geoffrey M. Cooper (2004). *The Cell: A Molecular Approach*, 832 pages.

[2] P. J. Chaput (2012).[online] Available at: http://www.futura-sciences.com/sante/ actualites/medecine-alzheimer-parkinson-nouvelle-piste-300-maladies-35922/ [Accessed 12 September 2017]

[3] D. Whitford (2005). *Proteins: Structure and Function*, 542 pages.

[4] R. Bailey (2017). [online] Available at: https://www.thoughtco.com/protein-function-373550 [Accessed 12 September 2017]

[5] G. Filiano (2016). [online]. Available at: http://sb.cc.stonybrook.edu/news/ general/2016-07-12-new-method-to-model-protein-interactions-may-help-accelerate-drug-development.php [Accessed 12 September 2017].

[6] G. Waksman (2005). *Proteomics and Protein-Protein Interactions: Biology, Chemistry, Bioinformatics, and Drug Design*, pp. 90-91.

[7] T. M. Mitchell (1997). Machine Learning. *McGraw-Hill Science/Engineering/ Math*, (March 1, 1997), pp. 3-5.

[8] I. Rish (2001). *An empirical study of the naive Bayes classifier*, pp. 2-3

[9] O. M. Lior Rokach (2008). Data mining with decision trees: theory and applications. *World Scientific Publishing Co. Pte. Ltd*, pp.4-5

[10] Zhang Q. et al (2012). *Structure-based prediction of protein-protein interactions on a genome-wide scale*, pp. 2-3.

[11] Pitre S. et al (2006). *PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs*, pp. 2-3.

[12] Liu B. et al (2009). *Prediction of protein-protein interactions based on*, pp. 2-3.

[13] Urquiza J. et al (2011). *Method for Prediction of Protein-Protein Interactions in Yeast Using Genomics/Proteomics Information and Feature Selection,* pp. 2-3.

[14] Szklarczyk D. et al (2011). *The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored*, pp. 2-3.

[15] Cai L. et al (2003). *SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence*, pp.3-4

[16] Zou Q et al (2013). Identifying Multi-Functional Enzyme by Hierarchical. *Journal of Computational & Theoretical Nanoscience*, pp. 1038-1043.

[17] Ioannis X. et al (2000). DIP: the Database of Interacting Proteins. *PubMed Central,* pp. 289-291.

[18] Philipp B. et al (2014). Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *PubMed Central,* 42:D396-D400.

[19] Liu B. et al (2008). A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. *BMC Bioinformatics,* 9:510.

[20] Zhu-Hong Y. et al (2015). Predicting Protein-Protein Interactions from Primary Protein Sequences Using a Novel Multi-Scale Local Feature Representation Scheme and the Random Forest. *PLoS One* 10.