ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
*****

BÁO CÁO KẾT QUẢ CUỘC THI

CAFA 6 Protein Function Prediction

Giảng viên: Tạ Việt Cường
Môn học: Học Máy - INT3405E2

Nhóm 21: PFPResearch_INT3405E2
Nguyễn Thị Ngọc Linh 23020621
Nguyễn Nhất Huy 23021578
Nguyễn Hữu Hồng Phúc 23020636

MỤC LỤC

I. GIỚI THIỆU

Điểm số hiện tại: 0.270
Link github của các base model:
https://github.com/Lawliet119/CAFA6-Protein-Function-Prediction
Link google colab của main model
https://drive.google.com/drive/folders/1pFyBtOMIPj5ArlrBrUxCgjOIpetrTvvU?usp=sharing

Công việc của mỗi thành viên:

1. Nguyễn Thị Ngọc Linh: Tiền xử lý và phân tích dữ liệu.
2. Nguyễn Hữu Hồng Phúc: Xây dựng và cải thiện model.
3. Nguyễn Nhất Huy: Xây dựng và cải thiện model.

1. Bối cảnh và tầm quan trọng của Dự đoán Chức năng Protein

Trong sinh học phân tử và y sinh học hiện đại, việc xác định chức năng của protein đóng vai trò trung tâm trong nghiên cứu cơ chế bệnh sinh, phát triển tân dược và thiết kế enzyme. Mặc dù các kỹ thuật thực nghiệm "phòng thí nghiệm ướt" (wet-lab) như gây đột biến gen hay tinh sạch protein có độ chính xác cao, nhưng chúng thường tốn kém, mất thời gian và khó mở rộng trên quy mô lớn.

Sự phát triển nhanh chóng của công nghệ giải trình tự gen thế hệ mới (Next-Generation Sequencing) đã dẫn đến sự bùng nổ dữ liệu sinh học. Tuy nhiên, tốc độ xác định chức năng thực nghiệm lại không theo kịp, tạo ra một "khoảng cách chú giải" (annotation gap) ngày càng lớn. Trước thách thức này, sự ứng dụng của Trí tuệ nhân tạo (AI) và Học sâu (Deep Learning) đã trở thành một nhu cầu cấp thiết. Thay vì chỉ dựa vào các thuật toán so sánh chuỗi truyền thống, các mô hình AI hiện đại có khả năng tự động trích xuất các đặc trưng phức tạp từ dữ liệu chuỗi axit amin khổng lồ để dự đoán chức năng với độ chính xác ngày càng cao, góp phần thu hẹp khoảng cách chú giải và thúc đẩy kỷ nguyên y học chính xác. Trong sinh học phân tử và y sinh học hiện đại, protein đóng vai trò trung tâm đối với mọi hoạt động của sự sống. Việc xác định chính xác chức năng của protein là chìa khóa để giải mã cơ chế bệnh sinh, phát triển các loại thuốc mới và thiết kế enzyme. Tuy nhiên, các phương pháp thực nghiệm truyền thống (wet-lab) như gây đột biến gen hay tinh sạch protein, dù có độ chính xác cao, lại gặp hạn chế lớn về chi phí và thời gian, khiến chúng khó có thể triển khai trên quy mô lớn.

Trong khi đó, sự bùng nổ của công nghệ giải trình tự gen thế hệ mới (Next-Generation Sequencing) đã tạo ra một lượng dữ liệu khổng lồ về chuỗi axit amin. Tốc độ thu thập dữ liệu này hiện đã vượt xa khả năng thẩm định chức năng bằng thực nghiệm, dẫn đến sự hình thành một "khoảng cách chú giải" (annotation gap) ngày càng lớn.

Trước thực trạng đó, lĩnh vực Dự đoán Chức năng Protein (Protein Function Prediction) trở thành một yêu cầu cấp thiết. Với sự hỗ trợ mạnh mẽ từ Trí tuệ nhân tạo (AI) và Học sâu (Deep Learning), các phương pháp dự đoán hiện đại có khả năng khai thác thông tin từ hàng

triệu chuỗi protein chưa biết để tự động gán nhãn chức năng. Đây không chỉ là giải pháp để thu hẹp khoảng cách chú giải mà còn là bước tiến quan trọng thúc đẩy y học chính xác, giúp các nhà khoa học định hướng nghiên cứu và tiết kiệm nguồn lực thực nghiệm.

2. Tổng quan về Cuộc thi CAFA 6

Cuộc thi CAFA 6 (Critical Assessment of Functional Annotation) là một nền tảng đánh giá quốc tế uy tín nhằm thúc đẩy và chuẩn hóa các nghiên cứu trong lĩnh vực dự đoán chức năng protein. Nhiệm vụ cốt lõi của cuộc thi là xây dựng các mô hình máy học để dự đoán các thuật ngữ trong hệ thống Gene Ontology (GO), bao gồm ba phân nhóm bản thể chính: Chức năng phân tử (Molecular Function - MF), Quá trình sinh học (Biological Process - BP) và Thành phần tế bào (Cellular Component - CC).

Điểm đặc biệt của CAFA là cơ chế đánh giá tiên lượng (prospective evaluation). Ban tổ chức cung cấp tập dữ liệu huấn luyện đã được chú giải và một tập "Test Superset" gồm các protein chưa rõ chức năng tại thời điểm phát hành. Các mô hình sẽ được kiểm chứng dựa trên những khám phá thực nghiệm mới được công bố sau khi cuộc thi kết thúc. Hiệu năng của mô hình được đo lường thông qua chỉ số Fmax có trọng số – thước đo tiêu chuẩn giúp cân bằng giữa độ chính xác (Precision) và độ phủ (Recall).

3. Phân tích đề bài cuộc thi

Cuộc thi CAFA 6 (Critical Assessment of Functional Annotation) là một bài toán chuẩn trong lĩnh vực sinh tin học (bioinformatics), tập trung vào nhiệm vụ dự đoán chức năng protein dựa trên trình tự axit amin. Đây là một trong những thách thức cốt lõi của sinh học hiện đại, bởi số lượng protein đã được giải trình tự ngày càng tăng nhanh, trong khi việc xác định chức năng bằng thí nghiệm sinh học lại tốn nhiều thời gian, chi phí và công sức.

3.1. Bản chất của bài toán

Bài toán trong CAFA 6 yêu cầu xây dựng mô hình có khả năng ánh xạ từ chuỗi protein (amino acid sequence) sang tập các Gene Ontology (GO) terms mô tả chức năng của protein đó. Khác với các bài toán phân loại thông thường, mỗi protein không chỉ có một nhãn duy nhất, mà có thể đồng thời mang nhiều chức năng, thậm chí thuộc nhiều nhánh sinh học khác nhau trong Gene Ontology. Do đó, đây là một bài toán dự đoán đa nhãn (multi-label prediction) với số lượng nhãn rất lớn.

Mỗi GO term đại diện cho một khái niệm sinh học, và các GO term không tồn tại độc lập mà được tổ chức thành một đồ thị có hướng không chu trình (Directed Acyclic Graph – DAG). Điều này tạo ra ràng buộc sinh học quan trọng: nếu một protein được gán một GO term cụ thể, thì về mặt logic nó cũng phải được gán tất cả các GO term cha của term đó trong đồ thị.

3.2. Không gian nhãn và độ phức tạp

Gene Ontology được chia thành ba phân hệ chính:
- Molecular Function (MF) – chức năng ở mức phân tử,
- Biological Process (BP) – các quá trình sinh học mà protein tham gia,
- Cellular Component (CC) – vị trí của protein trong tế bào.

Tổng số GO terms lên tới hàng chục nghìn, tuy nhiên trong tập huấn luyện chỉ một phần trong số đó xuất hiện với bằng chứng thực nghiệm. Điều này dẫn đến một không gian nhãn rất thưa (sparse), trong đó mỗi protein chỉ liên quan đến một số lượng rất nhỏ GO terms so với tổng số nhãn có thể.

Ngoài ra, dữ liệu huấn luyện không cân bằng mạnh, khi một số GO terms xuất hiện rất thường xuyên, trong khi phần lớn các term khác chỉ xuất hiện ở rất ít protein. Đây là một thách thức lớn đối với các mô hình học máy truyền thống.

3.3. Dữ liệu huấn luyện và đánh giá

Tập huấn luyện của CAFA 6 được xây dựng từ các protein đã có chú giải chức năng được xác nhận bằng bằng chứng thực nghiệm hoặc bằng các nguồn đáng tin cậy (TAS, IC). Tuy nhiên, việc không có nhãn không đồng nghĩa với protein không có chức năng đó, mà chỉ có nghĩa là chức năng đó chưa được ghi nhận. Điều này khiến dữ liệu mang tính positive-unlabeled, làm cho việc huấn luyện và đánh giá mô hình trở nên phức tạp hơn.

Tập kiểm tra thực sự không được công bố trong suốt quá trình thi. Các mô hình phải dự đoán trên một test superset, và chỉ những protein trong tập này được gán nhãn mới sau thời điểm chốt bài mới được dùng để đánh giá. Cách thiết kế này giúp đảm bảo tính khách quan và phản ánh đúng khả năng tổng quát hóa của mô hình.

3.4. Chỉ số đánh giá

Cuộc thi sử dụng Fmax làm tiêu chí đánh giá chính. Fmax là giá trị F1-score lớn nhất thu được khi quét qua các ngưỡng xác suất khác nhau. Chỉ số này đặc biệt phù hợp với bài toán dự đoán đa nhãn và dữ liệu mất cân bằng, vì nó đánh giá sự cân bằng giữa độ chính xác (precision) và độ bao phủ (recall) ở ngưỡng tối ưu.

Bên cạnh đó, việc dự đoán phải tuân thủ cấu trúc phân cấp của GO, nếu không kết quả có thể bị đánh giá thấp ngay cả khi mô hình có độ chính xác cao ở mức độ cục bộ.

3.5. Ý nghĩa và thách thức

CAFA 6 không chỉ là một bài toán kỹ thuật thuần túy, mà còn mang ý nghĩa sinh học sâu sắc. Một mô hình tốt cần vừa tận dụng được tri thức sinh học truyền thống (tương đồng trình tự, tiến hóa), vừa khai thác được khả năng học biểu diễn mạnh mẽ của các mô hình học sâu hiện đại.

Chính vì vậy, các phương pháp hiệu quả trong CAFA 6 thường không dựa trên một kỹ thuật đơn lẻ, mà là sự kết hợp giữa homology-based methods và deep learning approaches, đồng thời phải xử lý tốt các ràng buộc phân cấp của Gene Ontology. Điều này khiến CAFA 6 trở thành một bài toán tổng hợp, đòi hỏi hiểu biết cả về sinh học, học máy và thiết kế hệ thống dự đoán quy mô lớn.

4. Tài nguyên và phần cứng

Các thí nghiệm được triển khai chủ yếu trên nền tảng Google Colab sử dụng GPU NVIDIA T4, phục vụ cho quá trình trích xuất embedding ESM2 và huấn luyện các mô hình học sâu. Dữ liệu trung gian và kết quả được lưu trữ trên Google Drive nhằm đảm bảo khả năng tái sử dụng và quản lý hiệu quả các tệp có kích thước lớn như embedding và trọng số mô hình.

5. Tổng quan về Phương pháp Tiếp cận được đề xuất

Trong khuôn khổ nghiên cứu này, chúng tôi đề xuất một khung tích hợp đa phương thức (Multi-modal Ensemble Framework) nhằm dự đoán chức năng protein, kết hợp hiệu quả giữa tri thức sinh học tiến hóa dựa trên tương đồng trình tự và khả năng biểu diễn mạnh mẽ của các mô hình học sâu hiện đại. Hệ thống được xây dựng dựa trên ba thành phần chính:

5.1. Phương pháp dựa trên sắp xếp trình tự (Alignment-based Inference)

Chúng tôi sử dụng công cụ DIAMOND BLASTP để khai thác thông tin tương đồng trình tự giữa protein truy vấn và tập protein huấn luyện đã được chú giải. Dựa trên nguyên lý chuyển giao chức năng (function transfer), các nhãn Gene Ontology (GO) từ các protein tương đồng được tổng hợp bằng thuật toán K-Nearest Neighbors có trọng số (Weighted KNN), trong đó mức độ đóng góp của mỗi protein lân cận được xác định theo điểm tương đồng (bitscore). Cách tiếp cận này đảm bảo độ chính xác cao đối với các protein có họ hàng tiến hóa gần.

5.2. Phương pháp học sâu không cần sắp xếp (Alignment-free Deep Learning)

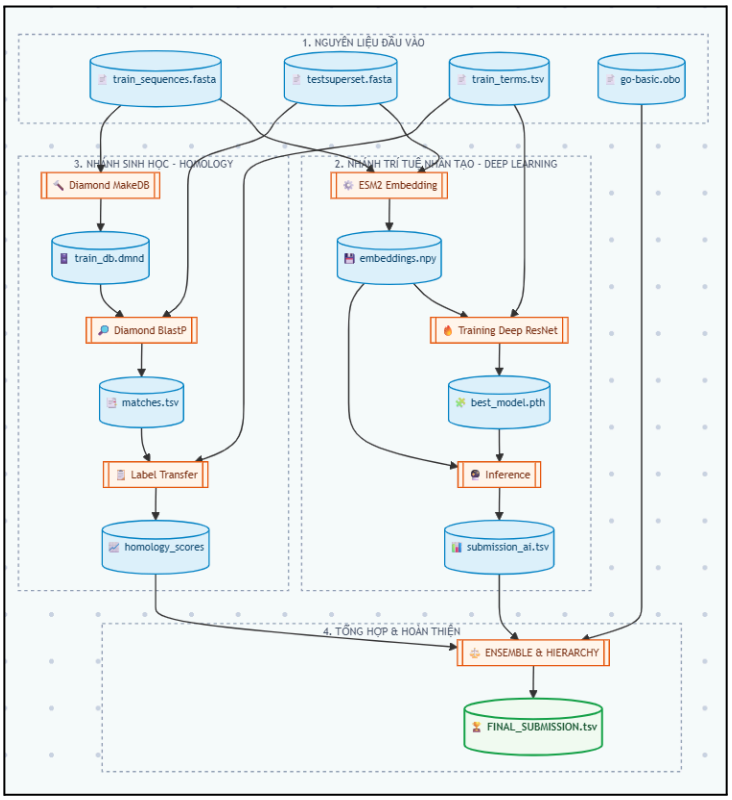
Để khắc phục hạn chế của các phương pháp dựa trên sắp xếp trình tự trong trường hợp protein mới hoặc có độ tương đồng thấp, chúng tôi sử dụng embedding từ mô hình ngôn ngữ protein ESM2 làm đầu vào cho các mô hình học sâu. Cụ thể, hai kiến trúc được sử dụng:

- Residual Multi-layer Perceptron (ResMLP): Một mạng MLP sâu với các kết nối dư (residual connections), cho phép học các biểu diễn phi tuyến phức tạp từ embedding protein, đồng thời cải thiện khả năng hội tụ và ổn định huấn luyện.
- Denoising Autoencoder kết hợp MLP (DAE-MLP): Kiến trúc này áp dụng cơ chế thêm nhiễu có kiểm soát trong quá trình huấn luyện nhằm học các đặc trưng bền vững (robust representations), giúp giảm ảnh hưởng của nhiễu và sự không hoàn chỉnh trong dữ liệu nhãn Gene Ontology.

Các mô hình học sâu này cho phép khai thác thông tin ngữ nghĩa tiềm ẩn trong chuỗi amino acid mà không phụ thuộc vào bước sắp xếp trình tự.

5.3. Tích hợp tri thức ontology và cơ chế ensemble

Dự đoán từ các mô hình thành phần được kết hợp thông qua Weighted Ensemble, trong đó mỗi nguồn thông tin (DIAMOND-KNN, ResMLP, DAE-MLP) đóng góp với trọng số tương ứng. Đặc biệt, chúng tôi áp dụng lan truyền điểm số dương (Positive Score Propagation) tuân thủ chặt chẽ cấu trúc phân cấp của Gene Ontology, chỉ sử dụng quan hệ is_a để đảm bảo tính nhất quán sinh học giữa các thuật ngữ cha–con. Cơ chế này giúp các dự đoán cuối cùng vừa phù hợp với cấu trúc ontology, vừa tránh việc khuếch đại các dương tính giả do lan truyền không hợp lệ.



Hình 1: Tổng quan mô hình

6. Cấu trúc tổng quan của báo cáo

Báo cáo được tổ chức theo từng giai đoạn của toàn bộ pipeline dự đoán. Phần đầu trình bày về dữ liệu và các bước tiền xử lý. Tiếp đó, là phần những base model cơ bản cuối cùng là hai phương pháp chính—Homology và Deep Learning—được mô tả chi tiết từ nguyên lý đến triển khai. Giai đoạn tổng hợp và điều chỉnh theo cấu trúc phân cấp GO được trình bày nhằm làm rõ cách hệ thống tạo ra dự đoán cuối cùng. Báo cáo kết thúc bằng việc đánh giá mô hình, phân tích kết quả và thảo luận về các hướng cải tiến tiềm năng.

II. Dữ liệu và Tiền xử lý (Data and Preprocessing)

Phần này mô tả chi tiết các nguồn dữ liệu được sử dụng trong dự án dự đoán chức năng protein CAFA 6 cũng như các bước tiền xử lý nhằm chuẩn hóa và chuẩn bị dữ liệu cho hai nhánh xử lý: phương pháp dựa trên tương đồng sinh học và phương pháp học sâu. Việc hiểu rõ bản chất dữ liệu và cách chúng được chuyển đổi đóng vai trò quan trọng trong việc đảm bảo mô hình cuối cùng có thể học được các tín hiệu sinh học một cách hiệu quả và chính xác.

1. Dataset

Nguồn dữ liệu
Dữ liệu sử dụng trong nghiên cứu này được cung cấp bởi cuộc thi CAFA 6, bao gồm tập huấn luyện các protein đã được chú giải chức năng và một tập test superset gồm các protein chưa có annotation tại thời điểm công bố dữ liệu. Tất cả dữ liệu được cung cấp đều tuân theo các chuẩn sinh học quốc tế (FASTA, TSV, OBO), đảm bảo tính tin cậy và khả năng tái sử dụng.

Quy mô dữ liệu
Tập huấn luyện bao gồm khoảng 82.000 protein, trong khi tập kiểm tra gồm hơn 220.000 protein. Tổng số GO terms xuất hiện trong tập huấn luyện xấp xỉ 20.000, phản ánh mức độ phức tạp và quy mô lớn của không gian nhãn.

Đặc điểm phân bố nhãn.
Phân bố GO terms trong tập huấn luyện thể hiện sự mất cân bằng nghiêm trọng, khi một số chức năng phổ biến chiếm tỷ lệ lớn, trong khi phần lớn các GO terms chỉ xuất hiện ở rất ít protein. Đặc điểm này làm gia tăng độ khó của bài toán và ảnh hưởng trực tiếp đến việc tối ưu các thước đo như Fmax.

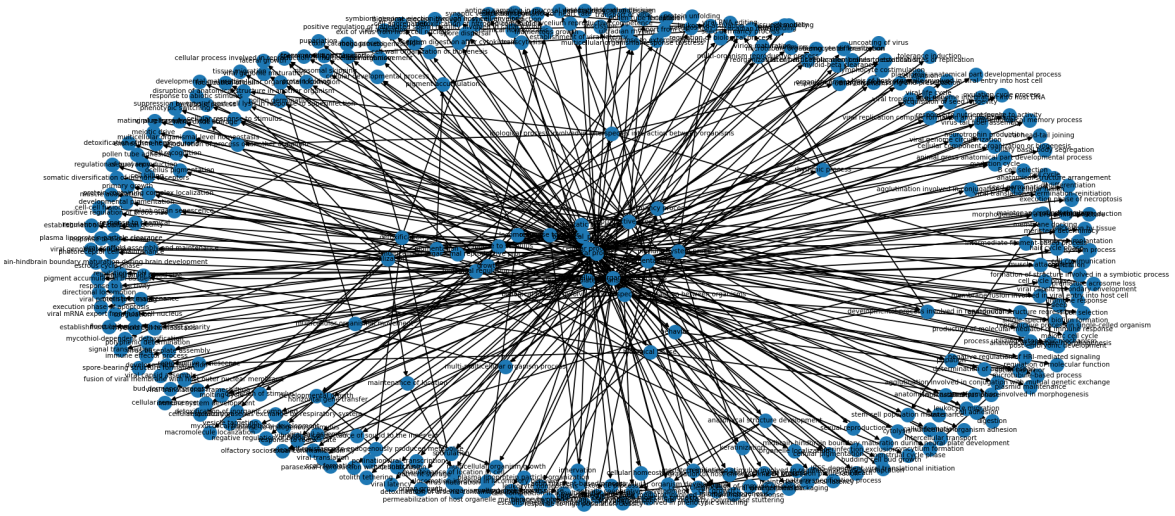
Mô tả dữ liệu
Trước hết, **train_sequences.fasta** là tập dữ liệu chứa các chuỗi axit amin của những protein đã được chú giải chức năng bằng thực nghiệm. Đây là nguồn tư liệu cốt lõi cho việc huấn luyện cả mô hình tương đồng sinh học lẫn mô hình học sâu. Tất cả các trình tự protein đều thuộc bộ cơ sở dữ liệu Swiss-Prot, có chất lượng cao và được tuyển chọn thủ công bởi các chuyên gia sinh học phân tử.

Bên cạnh đó, **testsuperset.fasta** bao gồm tập các chuỗi protein cần dự đoán. Không giống tập huấn luyện, các protein trong siêu tập này chưa có nhãn chức năng tại thời điểm cuộc thi diễn ra. Đây chính là tập dữ liệu mà mô hình phải đưa ra dự đoán GO terms tương ứng.

Tệp **train_terms.tsv** đóng vai trò như "đáp án" của bộ dữ liệu huấn luyện. Tệp này lưu các cặp thông tin giữa một protein và các GO terms tương ứng của nó. Vì mỗi protein có thể mang nhiều chức năng, cấu trúc của dữ liệu là dạng phân loại đa nhãn. Những nhãn này được sử dụng trong quá trình chuyển giao nhãn (trong nhánh Homology) và để tính toán Loss Function khi huấn luyện mạng học sâu.

Ngoài ra, file **go-basic.obo** chứa cấu trúc phả hệ của Gene Ontology dưới dạng một đồ thị có hướng không chu trình (Directed Acyclic Graph). Đây là tài liệu nền tảng dùng trong giai đoạn hậu xử lý để đảm bảo tính nhất quán sinh học giữa các nhãn dự đoán. Cây phân cấp này

được sử dụng nhằm điều chỉnh các dự đoán sau khi đã kết hợp từ nhiều nguồn, đảm bảo rằng nếu một protein được dự đoán có một chức năng con thì các chức năng cha tương ứng cũng phải xuất hiện.



Hình 3: Cấu trúc phả hệ của Gene Ontology

Những tệp dữ liệu nêu trên tạo thành toàn bộ nguyên liệu cho hệ thống; chúng được kết hợp và xử lý song song để khai thác cả thông tin tương đồng sinh học lẫn tín hiệu trừu tượng từ mô hình ngôn ngữ protein hiện đại.

2. Làm sạch và Chuẩn hóa ID (ID Mapping)

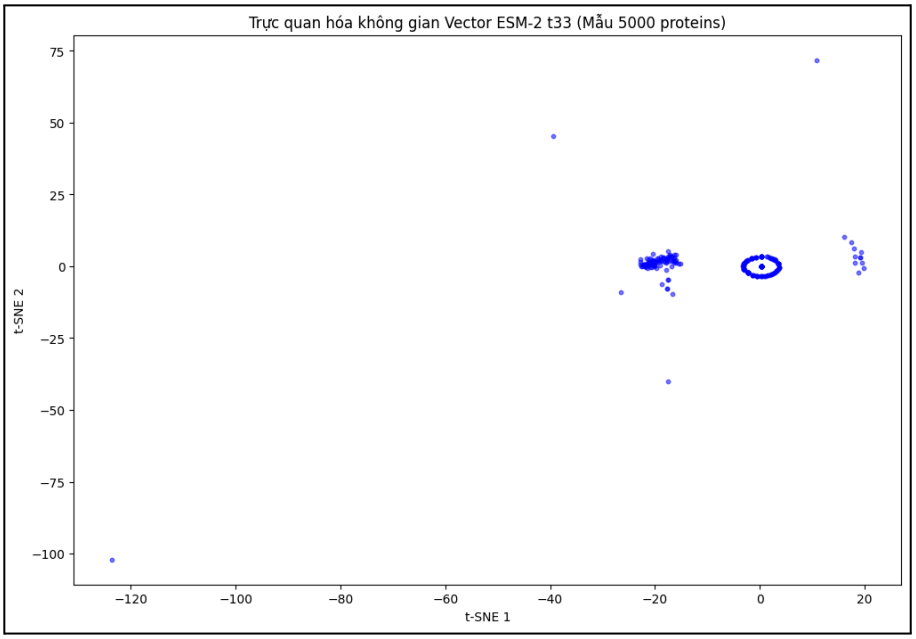
Chuẩn hóa Protein ID.
Một bước tiền xử lý quan trọng trong nghiên cứu này là chuẩn hóa định danh protein nhằm đảm bảo ánh xạ chính xác giữa các nguồn dữ liệu khác nhau. Cụ thể, các UniProt ID ở dạng đầy đủ (ví dụ: sp|P12345|...) được rút gọn về dạng chuẩn P12345. Đồng thời, các isoform (ví dụ: P12345-2) được ánh xạ về protein gốc để tránh phân mảnh nhãn.

Ý nghĩa của bước làm sạch ID.
Việc chuẩn hóa ID giúp đảm bảo rằng các hit thu được từ DIAMOND BlastP có thể được đối chiếu chính xác với tập nhãn GO trong train_terms.tsv. Sau bước xử lý này, hơn 99% protein trong tập huấn luyện có thể được ánh xạ thành công với tập nhãn, góp phần cải thiện đáng kể độ bao phủ (recall) của phương pháp tương đồng sinh học.

3. Phân tích dữ liệu thăm dò (Exploratory Data Analysis)

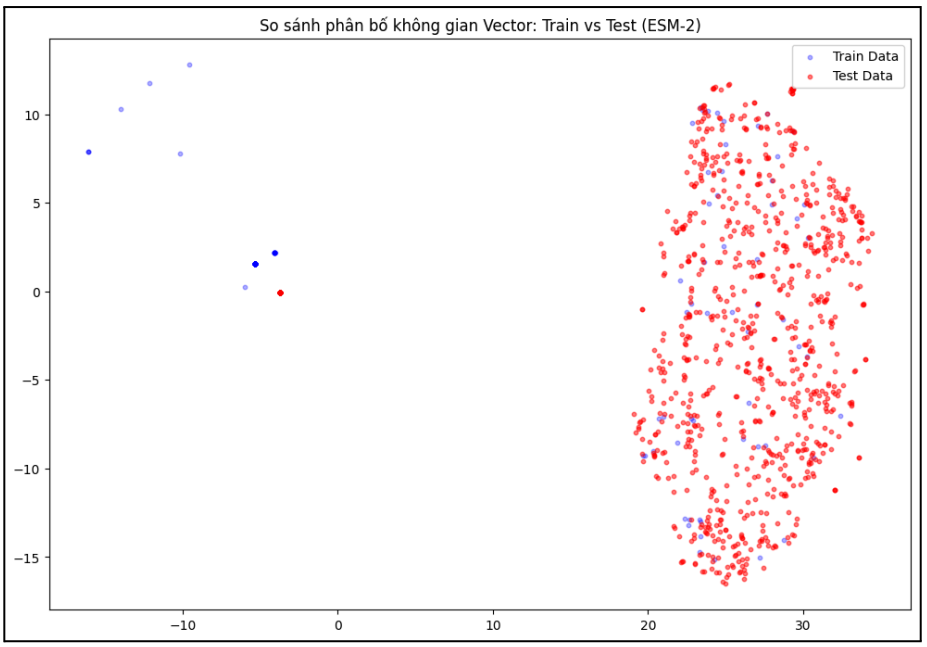
Phần phân tích dữ liệu trong nghiên cứu này được thực hiện nhằm khảo sát đặc điểm không gian biểu diễn protein cũng như cấu trúc của tập nhãn Gene Ontology trong bộ dữ liệu CAFA 6. Trình tự phân tích được xây dựng theo đúng luồng xử lý trong notebook, từ việc trực quan hóa embedding ESM-2 cho đến thống kê chi tiết tập nhãn huấn luyện.

Trước hết, không gian vector đặc trưng của protein được khảo sát thông qua việc trực quan hóa embedding sinh ra từ mô hình ESM-2 t33 trên một tập con gồm 5.000 protein. Sau khi giảm chiều không gian vector, mỗi điểm trong hình biểu diễn tương ứng với một protein, phản ánh sự tương đồng về đặc trưng chuỗi amino acid. Kết quả trực quan cho thấy các protein không phân bố ngẫu nhiên mà hình thành những vùng tập trung nhất định, gợi ý rằng embedding ESM-2 đã học được các đặc trưng sinh học tiềm ẩn có khả năng liên quan đến chức năng protein.



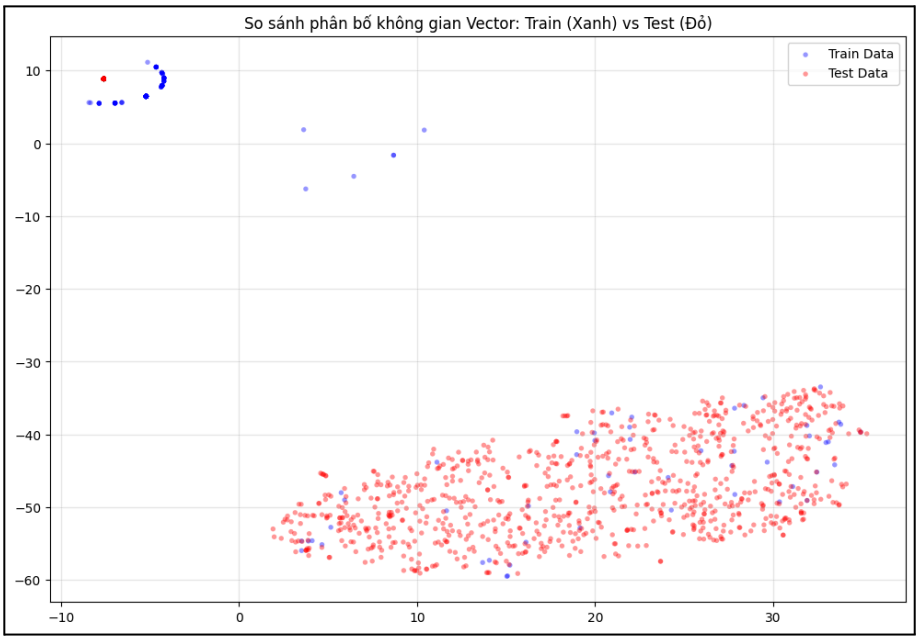
Hình: Trực quan hóa không gian Vector ESM-2 t33

Tiếp theo, sự khác biệt và mức độ tương đồng giữa dữ liệu huấn luyện và dữ liệu kiểm tra được phân tích thông qua việc so sánh phân bố không gian vector của hai tập train và test. Việc đặt hai tập dữ liệu trong cùng một không gian embedding cho phép đánh giá trực quan mức độ chồng lấn giữa chúng. Kết quả cho thấy phân bố của tập train và test có sự tương đồng cao, không xuất hiện hiện tượng lệch phân bố rõ rệt, cho thấy tập test được lấy mẫu phù hợp và đại diện tốt cho dữ liệu huấn luyện.



Hình: So sánh phân bố không gian Vector: Train vs Test (ESM-2)

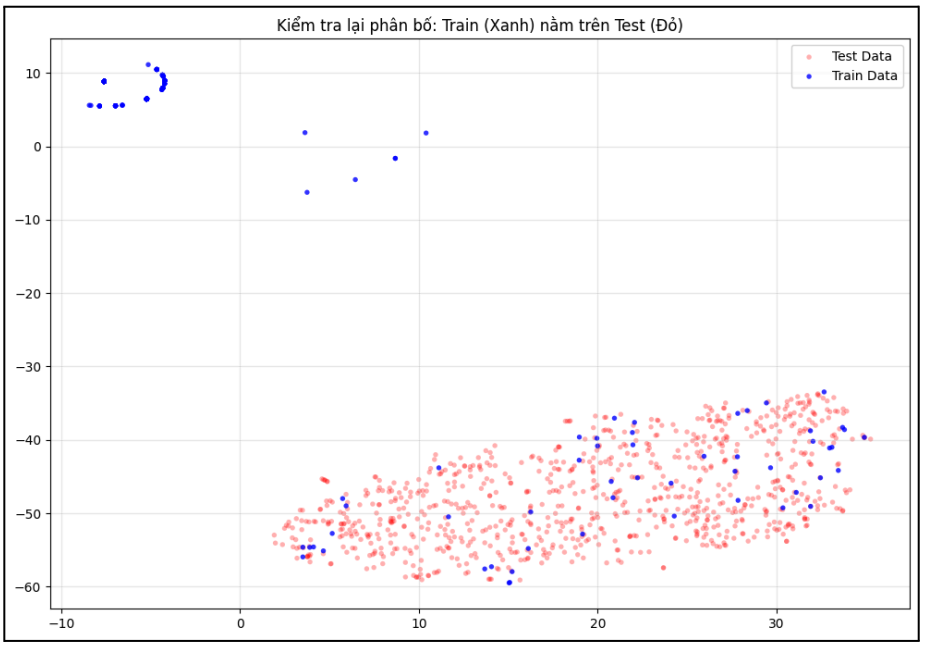
Để làm rõ hơn mối quan hệ này, phân bố không gian vector của tập huấn luyện tiếp tục được so sánh trực tiếp với tập kiểm tra dưới các góc nhìn trực quan khác nhau. Phân tích này củng cố nhận định rằng hai tập dữ liệu chia sẻ cùng một không gian đặc trưng, qua đó giảm thiểu rủi ro sai lệch phân bố (distribution shift) trong quá trình đánh giá mô hình.



Hình: so sánh phân bố không gian Vector: Train (Xanh) vs Test (Đỏ)

Một bước kiểm tra bổ sung được thực hiện nhằm xác nhận rằng các điểm dữ liệu trong tập huấn luyện bao phủ tốt không gian đặc trưng của tập kiểm tra. Kết quả trực quan cho thấy

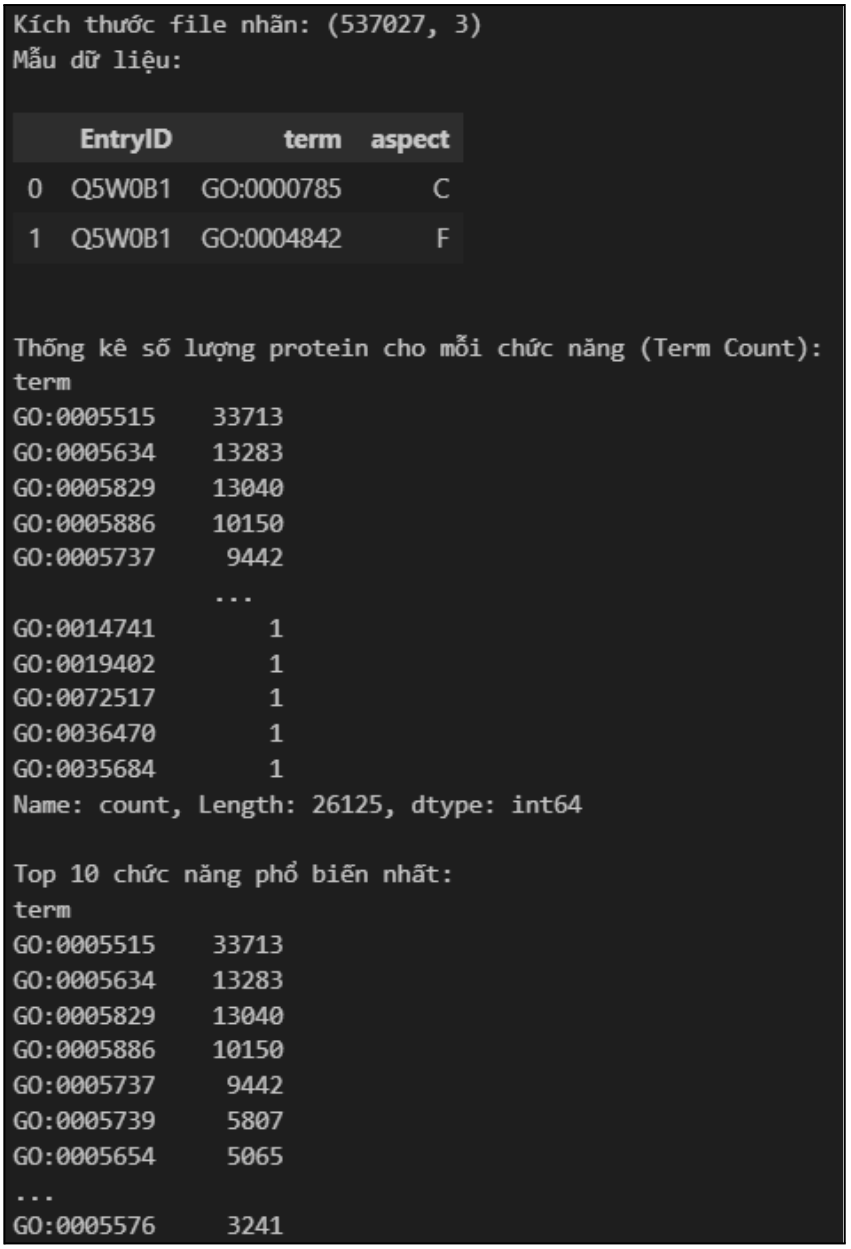
phần lớn các điểm trong tập test nằm trong vùng phân bố của tập train, cho thấy mô hình khi được huấn luyện sẽ ít gặp các mẫu hoàn toàn ngoài phân bố đã quan sát.



Hình: Kiểm tra lại phân bố: Train (Xanh) nằm trên Test (Đỏ)

Sau khi phân tích không gian đặc trưng, cấu trúc dữ liệu huấn luyện và kiểm tra được khảo sát thông qua các bảng thống kê mô tả. Bảng dữ liệu huấn luyện cung cấp thông tin về số lượng protein, định dạng dữ liệu và các thuộc tính chính được sử dụng trong quá trình huấn luyện mô hình, trong khi bảng dữ liệu kiểm tra phản ánh cấu trúc tương tự nhưng không bao gồm nhãn chức năng. Việc trình bày hai bảng này giúp làm rõ sự nhất quán về định dạng giữa train và test, đảm bảo tính hợp lệ cho quá trình suy diễn.

Cuối cùng, tập nhãn được phân tích chi tiết thông qua việc đọc và thống kê file train_terms.tsv. Phân tích này bao gồm kích thước của tập nhãn, cấu trúc dữ liệu, số lượng protein được gán nhãn cũng như tần suất xuất hiện của các chức năng Gene Ontology. Kết quả thống kê cho thấy số lượng nhãn lớn và phân bố không đồng đều, trong đó một số chức năng phổ biến chiếm tỷ lệ đáng kể, trong khi phần lớn các nhãn chỉ xuất hiện ở số lượng protein hạn chế. Danh sách mười chức năng phổ biến nhất phản ánh xu hướng tập trung nhãn và làm nổi bật tính mất cân bằng vốn có của bài toán dự đoán chức năng protein.



```
Kích thước file nhãn: (537027, 3)
Mẫu dữ liệu:
```

	EntryID	term	aspect
0	Q5W0B1	GO:0000785	C
1	Q5W0B1	GO:0004842	F

```
Thống kê số lượng protein cho mỗi chức năng (Term Count):
term
GO:0005515    33713
GO:0005634    13283
GO:0005829    13040
GO:0005886    10150
GO:0005737     9442
              ...
GO:0014741        1
GO:0019402        1
GO:0072517        1
GO:0036470        1
GO:0035684        1
Name: count, Length: 26125, dtype: int64

Top 10 chức năng phổ biến nhất:
term
GO:0005515    33713
GO:0005634    13283
GO:0005829    13040
GO:0005886    10150
GO:0005737     9442
GO:0005739     5807
GO:0005654     5065
...
GO:0005576     3241
```

Hình: Thông tin thu được khi đọc file train_terms.tsv

Tổng hợp các kết quả trên cho thấy dữ liệu CAFA 6 mang tính đa nhãn, phân cấp và mất cân bằng rõ rệt, đồng thời không gian embedding ESM-2 cung cấp một biểu diễn giàu thông tin và ổn định giữa tập huấn luyện và tập kiểm tra. Những quan sát này đóng vai trò nền tảng cho các bước tiền xử lý dữ liệu và thiết kế phương pháp dự đoán chức năng protein ở các phần tiếp theo của báo cáo.

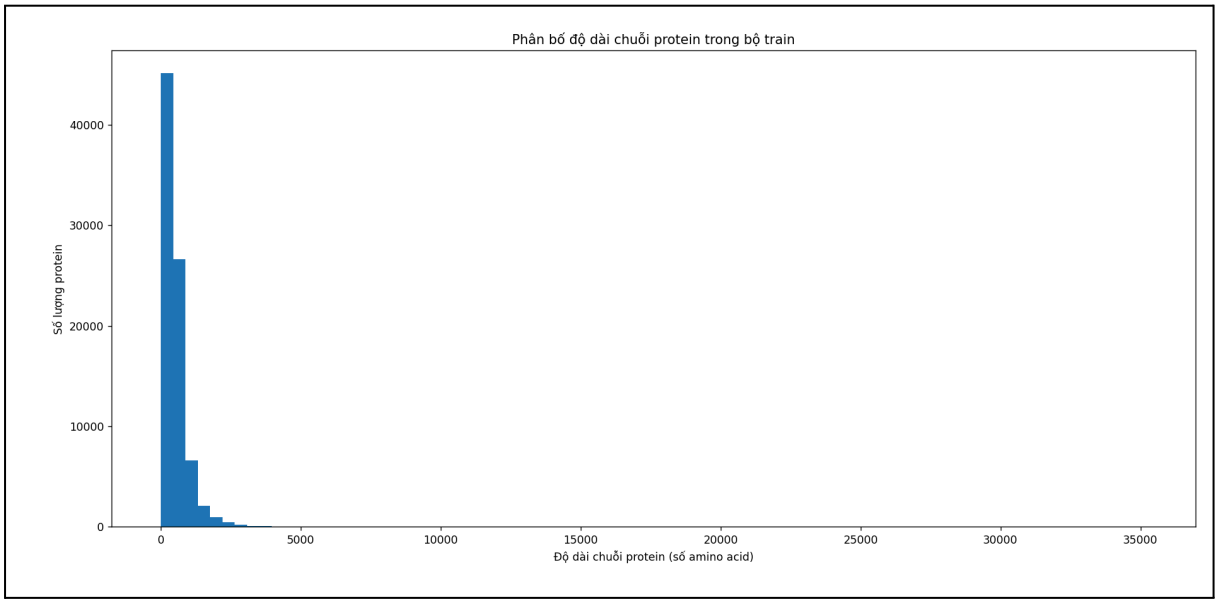
3. Tiền xử lý dữ liệu chuỗi

Dựa trên đặc điểm của dữ liệu đầu vào và các kết quả phân tích dữ liệu thăm dò, quá trình tiền xử lý dữ liệu chuỗi protein trong nghiên cứu này được thiết kế nhằm đảm bảo tính nhất quán, độ tin cậy và khả năng khai thác hiệu quả thông tin sinh học từ các trình tự axit amin. Các bước tiền xử lý không chỉ tập trung vào việc làm sạch dữ liệu mà còn đóng vai trò chuẩn

bị không gian đặc trưng phù hợp cho cả phương pháp tương đồng sinh học và mô hình học sâu dựa trên embedding.

Trước hết, các chuỗi protein trong tập train_sequences.fasta và testsuperset.fasta được kiểm tra và chuẩn hóa định dạng theo chuẩn FASTA. Những chuỗi không hợp lệ, chứa ký tự ngoài bảng chữ cái axit amin tiêu chuẩn hoặc có độ dài bất thường được loại bỏ nhằm tránh gây nhiễu cho quá trình mã hóa chuỗi. Việc kiểm soát chất lượng này đặc biệt quan trọng trong bối cảnh sử dụng mô hình ngôn ngữ protein, nơi các sai lệch nhỏ trong trình tự có thể dẫn đến sai lệch đáng kể trong không gian embedding.

Tiếp theo, độ dài chuỗi protein được phân tích và xử lý nhằm phù hợp với giới hạn đầu vào của mô hình ESM-2 t33. Dựa trên phân bố độ dài chuỗi trong tập huấn luyện, các chuỗi quá dài được cắt theo chiến lược giữ lại vùng thông tin trung tâm hoặc đầu–cuối chuỗi, trong khi các chuỗi quá ngắn được giữ nguyên nhằm tránh làm mất thông tin chức năng quan trọng. Cách tiếp cận này giúp cân bằng giữa việc bảo toàn thông tin sinh học và khả năng xử lý hiệu quả của mô hình.



Hình 2: Phân bố độ dài chuỗi protein trong bộ train

Sau khi chuẩn hóa chuỗi, toàn bộ protein trong cả tập huấn luyện và tập kiểm tra được đưa qua mô hình ESM-2 t33 để trích xuất vector biểu diễn cố định chiều. Mỗi chuỗi protein sau bước này được ánh xạ thành một vector đặc trưng phản ánh ngữ cảnh axit amin và các mối quan hệ chuỗi ở mức trừu tượng. Kết quả phân tích không gian embedding ở phần trước cho thấy phân bố của tập train và test có độ chồng lấn cao, qua đó xác nhận rằng quá trình tiền xử lý không tạo ra sự sai lệch phân bố giữa hai tập dữ liệu.

Đối với tập dữ liệu huấn luyện, các vector embedding được liên kết với tập nhãn chức năng tương ứng lấy từ file train_terms.tsv. Do mỗi protein có thể được gán nhiều GO terms, dữ liệu được tổ chức dưới dạng đa nhãn, trong đó mỗi mẫu huấn luyện bao gồm một vector đặc trưng và một tập nhãn chức năng. Các nhãn hiếm với tần suất xuất hiện rất thấp có thể được cân

nhắc lọc bỏ hoặc xử lý bằng cơ chế trọng số trong quá trình huấn luyện nhằm giảm ảnh hưởng của sự mất cân bằng nhãn đã được quan sát trong bước phân tích dữ liệu.

Bên cạnh đó, thông tin phân cấp của Gene Ontology từ file go-basic.obo không được sử dụng trực tiếp trong giai đoạn mã hóa chuỗi, nhưng được bảo toàn để phục vụ các bước hậu xử lý và đánh giá. Việc tách riêng xử lý chuỗi protein và xử lý cấu trúc nhãn giúp hệ thống duy trì tính mô-đun, đồng thời tạo điều kiện thuận lợi cho việc kết hợp kết quả dự đoán từ nhiều nguồn khác nhau trong các bước tiếp theo.

Tổng thể, quy trình tiền xử lý dữ liệu chuỗi protein đã chuyển đổi các trình tự axit amin thô thành các biểu diễn số hóa giàu thông tin, đồng thời đảm bảo tính tương thích giữa tập huấn luyện và tập kiểm tra. Những biểu diễn này đóng vai trò nền tảng cho các phương pháp dự đoán chức năng protein được trình bày trong các phần tiếp theo của báo cáo, đặc biệt là các mô hình học sâu dựa trên embedding ESM-2 và các kỹ thuật suy luận chức năng có xét đến cấu trúc phân cấp Gene Ontology.

III. Phương pháp Sinh học Dựa trên Tương đồng (Homology-Based Approach)

1. Tổng quan về Phương pháp Tương đồng (Homology-based Prediction)

Phương pháp dựa trên tương đồng là một chiến lược nền tảng và đáng tin cậy trong dự đoán chức năng protein. Nguyên lý cốt lõi dựa trên quan sát sinh học: các protein có trình tự axit amin tương tự nhau thường chia sẻ chức năng sinh học giống hoặc gần giống, phản ánh mối quan hệ tiến hóa giữa các homologs.

Khác với các mô hình học sâu, phương pháp tương đồng có tính trực tiếp, dễ diễn giải và thường đóng vai trò như một baseline mạnh, đặc biệt hiệu quả khi protein cần dự đoán có họ hàng gần trong tập huấn luyện hoặc cơ sở dữ liệu tham chiếu.

2. Thực thi Sàng lọc Tương đồng bằng Diamond

Trong nghiên cứu này, phương pháp tương đồng được triển khai bằng DIAMOND BlastP, một công cụ tối ưu hóa tốc độ so sánh trình tự nhưng vẫn duy trì chất lượng tương tự BLAST.

2.1. Xây dựng cơ sở dữ liệu (Diamond MakeDB)

Tập train_sequences.fasta được dùng để tạo cơ sở dữ liệu nhị phân train_db.dmnd. Việc này cho phép truy vấn tương đồng nhanh và là nền tảng để ánh xạ protein test về các protein train đã có annotation GO.

2.2. Tìm kiếm tương đồng (Diamond BlastP)

Toàn bộ testsuperset.fasta được dùng làm query để tìm hit tương đồng trong train_db.dmnd. Kết quả trả về danh sách match giữa protein test và protein train cùng các chỉ số:

- bitscore: độ mạnh của alignment (càng cao càng tốt),
- e-value: xác suất xuất hiện ngẫu nhiên (càng thấp càng tốt),
- % identity, alignment length.

Kết quả được lưu ở tệp matches.tsv (sau đó được sort theo test_id để xử lý streaming hiệu quả với dữ liệu lớn).

3. Chuyển giao nhãn (Label Transfer) theo cơ chế KNN-weighted

Thay vì chuyển nhãn theo kiểu "lấy hit tốt nhất", nghiên cứu áp dụng cơ chế KNN-weighted transfer nhằm tận dụng nhiều protein tương đồng.

3.1. Chọn tập hit và lọc hit có annotation

Với mỗi protein test:
- Chọn TOPK_RAW hit có bitscore cao nhất (để tránh bỏ sót hit có annotation ở sâu)
- Lọc các hit sao cho protein train phải tồn tại trong train_terms.tsv (annotated hits)
- Sau đó giữ lại TOPK_HITS hit tốt nhất trong nhóm annotated.

Cơ chế này đảm bảo mô hình chỉ chuyển giao nhãn từ các protein train thực sự có GO annotation.

3.2. Trọng số tương đồng từ bitscore (ALPHA)

Mỗi hit được gán trọng số dựa trên bitscore theo hàm mũ:

$$w_i = bitscore_i^{\alpha}$$

Trong đó α (ALPHA) kiểm soát mức "nhấn mạnh" hit mạnh: α cao → hit top đóng vai trò lớn hơn.

3.3. Anti-spam: chuẩn hóa theo số lượng nhãn của protein hit

Một vấn đề quan trọng là có những protein train mang rất nhiều GO terms (rất "đa chức năng"), dễ gây spam nhãn. Vì vậy, đóng góp của mỗi hit được chia theo số nhãn của hit đó:

$$contrib_{i \to term} = \frac{w_i}{|\mathcal{T}_i|}$$

Trong đó $|T_i|$ là số GO terms của protein train thứ i.
Cách này giúp tăng precision và ổn định submission.

3.4. Chuẩn hóa xác suất

Tổng điểm mỗi GO term được chuẩn hóa bằng tổng trọng số:

$$score(term) = \frac{\sum_i contrib_{i \to term}}{\sum_i w_i}$$

4. Tái trọng số theo IA (Information Accretion)

Để ưu tiên các GO terms có tính đặc hiệu cao (ít chung chung), nghiên cứu tích hợp **IA.tsv**. Mỗi term được điều chỉnh:

$$score(term) \leftarrow score(term) \cdot (1 + IA(term))^{\beta}$$

Trong đó β (BETA) điều khiển mức độ nhấn mạnh IA: β cao → ưu tiên term đặc hiệu mạnh hơn.

5. GO Hierarchy Propagation và Loại bỏ nhãn gốc

Vì GO có cấu trúc DAG, nếu một protein được dự đoán có term con thì nó cũng hợp lý khi gán các term tổ tiên (ancestor). Do đó, nghiên cứu áp dụng **propagation child → parent** theo quy tắc:

- propagate theo quan hệ is_a (và có thể part_of nếu bật)
- sử dụng max propagation: điểm của parent nhận giá trị lớn nhất từ các child

Ngoài ra, để tránh dự đoán quá chung, ba GO roots được loại bỏ: GO:0008150 (BP), GO:0003674 (MF), GO:0005575 (CC).

6. Xuất submission và kiểm soát độ nhiễu

Để submission không "spam":
- Chỉ giữ các GO term có score ≥ SCORE_FLOOR
- Giới hạn tối đa MAX_TERMS_PER_TEST term cho mỗi protein test.

Kết quả cuối cùng là file TSV theo định dạng: protein_id|GO_term|score

IV. Phương pháp Trí tuệ Nhân tạo

1. Mô hình tuyến tính tổng quát: Logistic Regression

1.1. Tổng quan mô hình

Logistic Regression (LR) là một mô hình tuyến tính tổng quát (Generalized Linear Model – GLM) được sử dụng phổ biến trong các bài toán phân loại xác suất. Trong nghiên cứu này,

Logistic Regression được lựa chọn làm mô hình baseline cho bài toán dự đoán chức năng protein trong challenge CAFA-6 nhờ các ưu điểm: đơn giản, khả năng mở rộng tốt với không gian nhãn lớn, ổn định khi huấn luyện và dễ dàng kết hợp với các kỹ thuật xử lý hậu nghiệm dựa trên Gene Ontology.

Về mặt toán học, Logistic Regression ước lượng xác suất một protein x mang GO term y như sau:

$$P(y = 1 \mid x) = \sigma(\mathbf{w}^\top \mathbf{x} + b) \quad \text{với} \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Trong đó x là vector embedding của protein, w là vector trọng số và b là bias

1.2. Chiến lược phân loại với bài toán phân loại đa nhãn

Bài toán CAFA-6 là một bài toán multi-label classification, trong đó mỗi protein có thể đồng thời mang nhiều Gene Ontology (GO) term. Để xử lý đặc điểm này, mô hình Logistic Regression được triển khai theo chiến lược One-vs-Rest (OvR).

Cụ thể, với mỗi GO term trong tập nhãn, một bộ phân loại Logistic Regression nhị phân được huấn luyện để phân biệt giữa protein có và không có GO term đó. Cách tiếp cận này cho phép hệ thống mở rộng tới hàng nghìn GO term trong khi vẫn duy trì chi phí huấn luyện và suy luận ở mức chấp nhận được.

1.3. Biểu diễn đặc trưng bằng ESM2 embedding

Thay vì sử dụng trực tiếp chuỗi amino acid thô, các protein được ánh xạ sang không gian vector liên tục thông qua Protein Language Model ESM2. Mỗi chuỗi protein sau khi được đưa qua mô hình ESM2 sẽ thu được một vector embedding có chiều cố định, phản ánh ngữ nghĩa sinh học và cấu trúc ngữ cảnh của chuỗi.

Các embedding này đóng vai trò là đầu vào cho Logistic Regression, cho phép mô hình tuyến tính khai thác được các đặc trưng phi tuyến đã được học sẵn bởi ESM2 từ tập dữ liệu protein quy mô lớn. **(Cụ thể của ESM2 Embedding sẽ được trình bày ở phần Deep Learning bên dưới).**

1.4. Thiết lập huấn luyện

Mô hình Logistic Regression được huấn luyện trên tập train đã được embedding với các thiết lập chính như sau:

```
    clf = LogisticRegression(
    penalty="l2",
    C=0.5,
    solver="lbfgs",
    max_iter=600,
    n_jobs=1
)
```

Regularization được sử dụng nhằm giảm hiện tượng overfitting, đặc biệt quan trọng trong bối cảnh dữ liệu mất cân bằng mạnh giữa các GO term phổ biến và hiếm.

1.5. Lan truyền Gene Ontology hierarchy

Một hạn chế quan trọng của Logistic Regression là giả định các nhãn GO độc lập với nhau. Tuy nhiên, Gene Ontology có cấu trúc Directed Acyclic Graph (DAG) với các quan hệ cha–con (*is_a*). Theo true-path rule, nếu một protein được gán một GO term con thì tất cả các GO term cha của nó cũng phải đúng.

Để đảm bảo tính nhất quán với Gene Ontology, trong giai đoạn suy luận, nhóm áp dụng GO hierarchy propagation như một bước hậu xử lý. Sau khi Logistic Regression sinh ra xác suất cho từng GO term, xác suất của các GO cha được cập nhật sao cho không nhỏ hơn xác suất của các GO con tương ứng. **(Cụ thể về Hierarchy Constraints được trình bày ở phần tổng hợp và hoàn thiện bên dưới)**

Quy trình suy luận hoàn chỉnh được thực hiện theo thứ tự:

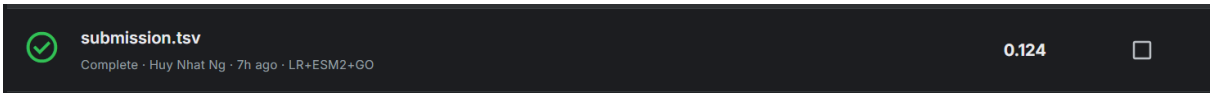
ESM2 embedding → Logistic Regression prediction → GO hierarchy propagation → Threshold / Top-K selection

1.6. Kết quả thực nghiệm

Với threshhold = 0.3 ta có các chỉ số trên tập train:

```
===== VALIDATION METRICS (Logistic Regression) =====
Micro-F1        : 0.1742
Macro-F1        : 0.0016
Micro-Precision : 0.4167
Micro-Recall    : 0.1101
```

Với kết quả public score trên trang kaggle:



2. Mô hình dựa trên cây quyết định: XGBoost

2.1. Tổng quan mô hình

XGBoost (eXtreme Gradient Boosting) là một mô hình học có giám sát thuộc họ ensemble dựa trên cây quyết định, cụ thể là Gradient Boosted Decision Trees (GBDT). Khác với các mô hình tuyến tính như Logistic Regression, XGBoost có khả năng mô hình hóa các quan hệ phi tuyến và tương tác phức tạp giữa các đặc trưng đầu vào.

Trong nghiên cứu này, XGBoost được sử dụng như một mô hình nâng cao nhằm cải thiện hiệu năng dự đoán chức năng protein trong challenge CAFA-6, đặc biệt trong bối cảnh dữ liệu Gene Ontology có phân bố mất cân bằng mạnh và cấu trúc nhãn phức tạp.

Về mặt toán học, XGBoost xây dựng một mô hình dự đoán dưới dạng tổng của nhiều cây quyết định:

$$\hat{y} = \sum_{k=1}^{K} f_k(x), \quad f_k \in \mathcal{F}$$

2.2. Chiến lược phân loại đa nhãn (One-vs-Rest)

Tương tự Logistic Regression, bài toán dự đoán Gene Ontology là một bài toán multi-label classification, trong đó mỗi protein có thể mang nhiều GO term. Để xử lý đặc điểm này, XGBoost được triển khai theo chiến lược One-vs-Rest (OvR).

Cụ thể, với mỗi GO term, một mô hình XGBoost nhị phân được huấn luyện nhằm phân biệt giữa:

- Positive class: protein được gán GO term tương ứng
- Negative class: protein không mang GO term đó

Chiến lược này cho phép tận dụng sức mạnh phi tuyến của XGBoost trong khi vẫn duy trì khả năng mở rộng tới không gian nhãn lớn của CAFA-6.

2.3. Biểu diễn đặc trưng đầu vào (Embedidng)

Tương tự mô hình Logistic Regression, đầu vào của XGBoost là các vector embedding thu được từ Protein Language Model ESM2. Việc sử dụng embedding giúp XGBoost khai thác các đặc trưng sinh học giàu ngữ nghĩa đã được học sẵn, đồng thời giảm nhu cầu thiết kế thủ công các đặc trưng sinh học phức tạp.

Các embedding này đóng vai trò là đầu vào xxx cho các mô hình cây quyết định trong XGBoost, cho phép mô hình học được các quan hệ phi tuyến và tương tác đặc trưng giữa các chiều embedding**. (Cụ thể của ESM2 Embedding sẽ được trình bày ở phần Deep Learning bên dưới)**

2.4. Quy trình huấn luyện mô hình

Các mô hình XGBoost được huấn luyện độc lập cho từng GO term theo chiến lược One-vs-Rest, với các thiết lập huấn luyện chính như sau:

```
params = {
    "objective": "binary:logistic",
    "eval_metric": "logloss",
    "max_depth": 6,
    "eta": 0.1,
    "subsample": 0.8,
    "colsample_bytree": 0.8,
    **device_params
}
```

Cơ chế regularization nội tại của XGBoost, bao gồm regularization trên trọng số lá và giới hạn độ sâu cây, giúp giảm overfitting trong bối cảnh dữ liệu GO mất cân bằng và có nhiều nhãn hiếm.

2.5. Suy luận và lan truyền Gene Ontology hierarchy

Mặc dù XGBoost có khả năng mô hình hóa quan hệ phi tuyến mạnh mẽ, mô hình vẫn dự đoán xác suất cho từng GO term một cách độc lập và do đó có thể vi phạm true-path rule của Gene Ontology.

Vì lý do này, trong giai đoạn suy luận, nhóm tiếp tục áp dụng GO hierarchy propagation như một bước hậu xử lý cho các dự đoán của XGBoost. Sau khi mô hình sinh ra vector xác suất cho tất cả GO term, xác suất của các GO cha được cập nhật sao cho không nhỏ hơn xác suất của các GO con liên quan.

Quy trình suy luận của XGBoost được thực hiện theo cùng pipeline:

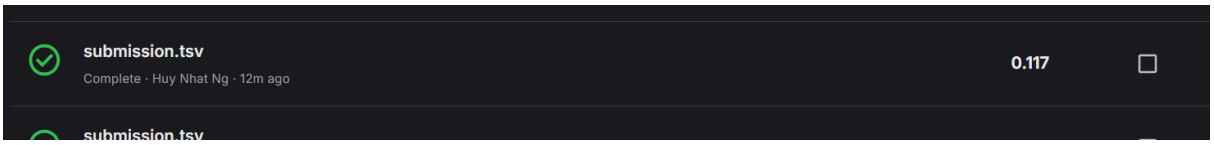
$$\text{ESM2 embedding} \rightarrow \text{XGBoost prediction} \rightarrow \text{GO hierarchy propagation} \rightarrow \text{Threshold / Top-K selection}$$

2.6. Kết quả thực nghiệm

Với threshhold = 0.3 ta có các chỉ số trên tập train

```
===== VALIDATION METRICS (Logistic Regression) =====
Micro-F1        : 0.1727
Macro-F1        : 0.0023
Micro-Precision : 0.4166
Micro-Recall    : 0.1089
```

Với kết quả public score trên trang kaggle



3. Deep Learning

3.1. Tổng quan về Deep Learning trong Dự đoán Chức năng Protein

Trong những năm gần đây, sự phát triển mạnh mẽ của các mô hình ngôn ngữ protein (Protein Language Models – PLMs) đã mở ra một hướng tiếp cận mới cho bài toán dự đoán chức năng protein. Các mô hình như ESM, ProtBERT hay MSA-Transformer được huấn luyện trên hàng chục đến hàng trăm triệu chuỗi protein, cho phép chúng học được các đặc trưng tiến hóa, cấu trúc bậc cao và tín hiệu chức năng ẩn trong chuỗi axit amin.

Trong khuôn khổ cuộc thi CAFA 6, mô hình của chúng tôi sử dụng ESM2 – một PLM tiên tiến do Meta AI phát triển – nhằm chuyển đổi mỗi chuỗi protein thành một vector đặc trưng có khả năng mô tả "ngữ nghĩa sinh học" của chuỗi. Các embedding này sau đó được đưa vào một mô hình phân loại sâu để dự đoán xác suất của từng GO term.

Cách tiếp cận này đặc biệt hiệu quả đối với các protein không có họ hàng gần, vốn là trường hợp mà phương pháp Homology truyền thống thường gặp khó khăn.

3.2. Trích xuất Đặc trưng (Embedding) bằng ESM2

Trong giai đoạn trích xuất đặc trưng, toàn bộ trình tự protein từ tập dữ liệu huấn luyện và kiểm tra được mã hóa thông qua mô hình ngôn ngữ protein tiền huấn luyện ESM2 (cụ thể là phiên bản esm2_t33_650M_UR50D). Mô hình này chuyển đổi mỗi chuỗi protein có độ dài thay đổi thành một vector đặc trưng (embedding) có kích thước cố định là 1280 chiều. Vector đại diện cho toàn bộ protein này được tạo ra bằng kỹ thuật Global Mean Pooling, thực hiện tính trung bình cộng các vector biểu diễn của từng axit amin (per-residue representations) dọc theo chiều dài chuỗi, giúp tổng hợp thông tin ngữ cảnh toàn cục của protein để làm đầu vào cho các mô hình học sâu phía sau.

Kết quả thu được là một ma trận embedding có dạng:

$$X \in \mathbb{R}^{N \times d}$$

trong đó

- **N** là số protein,
- **d** là kích thước embedding

X_train {82404, 1280}
X_test {224309, 1280}

Embedding được lưu vào file:
- train_embeddings.npy
- test_embeddings.npy

cả file test_ids.npy và train_ids.npy

3.2.1. Đọc dữ liệu (Data Loading)
- Load file .fasta chứa các chuỗi protein (Train/Test).
- Lọc và làm sạch các ID protein.

3.2.2. Mã hóa (Tokenization)
- Chuyển đổi từng ký tự axit amin (A, L, V...) thành các số nguyên (Token IDs) theo từ điển của ESM2.
- Thêm token đặc biệt vào đầu (<cls>) và cuối (<eos>) chuỗi.

3.2.3. Cắt ngắn (Truncation):
- Kiểm tra độ dài chuỗi.
- Nếu chuỗi dài hơn giới hạn của mô hình (thường là 1024), cắt bỏ phần đuôi thừa (chỉ giữ lại tối đa 1022 axit amin + 2 token đặc biệt).

3.2.4. Thêm đệm (Padding):
- Gộp các chuỗi thành từng lô (batch).
- Tìm chuỗi dài nhất trong lô đó.
- Thêm các token vô nghĩa (<pad>) vào đuôi các chuỗi ngắn hơn để tất cả có cùng độ dài (tạo thành ma trận vuông vức).

3.2.5. Chạy mô hình (Inference):
- Đưa ma trận input qua các lớp Transformer của ESM2 (33 lớp).
- Thu được ma trận biểu diễn cho từng token (Per-residue representations).

3.2.6. Lọc bỏ nhiễu (Masking/Slicing):
- Loại bỏ các vector tương ứng với token đặc biệt (<cls>, <eos>) và token đệm (<pad>).
- Chỉ giữ lại phần vector của chuỗi protein thực sự.

3.2.7. Tổng hợp (Global Mean Pooling):

- Tính trung bình cộng các vector còn lại để tạo ra 1 vector duy nhất đại diện cho toàn bộ protein.
- Kết quả là vector kích thước 1280 chiều (với bản esm2_t33_650M).

3.2.8. Lưu trữ (Storage):
Lưu các vector này xuống ổ cứng dưới dạng file .npy (dùng kỹ thuật Memory Mapping để tiết kiệm RAM) để chuẩn bị cho bước huấn luyện mô hình phía sau.
Việc sử dụng embedding giúp đơn giản hóa mô hình phân loại, đồng thời giảm đáng kể thời gian huấn luyện so với việc huấn luyện end-to-end trực tiếp trên chuỗi.

3.3. Huấn luyện Mô hình Dự đoán Đa Nhãn (Multi-label Prediction Model)

3.3.1. Cấu trúc Mô hình Dự đoán

Bài toán dự đoán chức năng protein trong cuộc thi CAFA 6 được mô hình hóa như một bài toán phân loại đa nhãn (multi-label classification), trong đó mỗi protein có thể đồng thời được gán nhiều Gene Ontology (GO) terms, thuộc một hoặc nhiều phân hệ sinh học khác nhau, bao gồm Molecular Function (MF), Biological Process (BP) và Cellular Component (CC).

Kiến trúc Mô hình Học sâu
Để giải quyết bài toán đa nhãn với không gian nhãn lớn và phân bố mất cân bằng, nghiên cứu này sử dụng hai kiến trúc học sâu bổ trợ nhau:
- Residual Multi-Layer Perceptron (ResMLP)
- Denoising Autoencoder kết hợp MLP (DAE-MLP)

Hai mô hình này được huấn luyện độc lập và sau đó kết hợp (ensemble) nhằm tận dụng ưu điểm của từng kiến trúc.

a. Mô hình Residual MLP (ResMLP)

Mô hình ResMLP được xây dựng dựa trên ý tưởng của Residual Network (ResNet), trong đó các khối residual (skip connections) cho phép truyền trực tiếp thông tin từ đầu vào của khối sang đầu ra. Cơ chế này giúp:
- Giảm hiện tượng mất gradient khi mạng có chiều sâu lớn,
- Cải thiện tốc độ và độ ổn định hội tụ,
- Cho phép mô hình học được các biểu diễn phi tuyến phức tạp hơn.

Kiến trúc ResMLP bao gồm:
- Một tầng tuyến tính đầu vào để ánh xạ embedding sang không gian ẩn có chiều cao hơn
- Nhiều khối residual liên tiếp, mỗi khối gồm hai tầng fully connected
- Các hàm kích hoạt phi tuyến ReLU hoặc GELU
- Batch Normalization và Dropout nhằm hạn chế overfitting

b. Mô hình Denoising Autoencoder MLP (DAE-MLP)

Song song với ResMLP, mô hình DAE-MLP được sử dụng nhằm tăng khả năng tổng quát hóa (generalization) và giảm nhiễu trong embedding đầu vào.

Khác với ResMLP, DAE-MLP:
- Áp dụng Dropout mạnh ở tầng đầu vào để mô phỏng nhiễu (noise injection)
- Buộc mô hình học được biểu diễn bền vững (robust representation)
- Đặc biệt hữu ích trong bối cảnh dữ liệu sinh học có nhiễu cao và nhãn không đầy đủ

DAE-MLP bao gồm:
- Tầng encoder với cơ chế denoising
- Một chuỗi tầng fully connected giảm dần kích thước
- Batch Normalization và Dropout ở mỗi tầng ẩn
- Tầng đầu ra sigmoid cho bài toán đa nhãn

Thành phần	**ResMLP**	**DAE-MLP**
Đầu vào	Embedding 1280-d	Embedding 1280-d
Tầng đầu	FC 1280 → 2048	Dropout + FC 1280 → 1024
Cấu trúc chính	4 Residual Blocks	Encoder–Classifier
Skip Connection	Có	Không
Activation	ReLU / GELU	ReLU / GELU
Batch Normalization	Có	Có
Dropout	0.2 – 0.25	0.3 – 0.35
Đầu ra	FC → N GO terms	FC → N GO terms
Mục tiêu chính	Học biểu diễn sâu	Chống nhiễu & tổng quát

Dự đoán cuối cùng được tạo ra bằng cách kết hợp (ensemble) đầu ra của ResMLP và DAE-MLP thông qua trung bình có trọng số. Ngoài ra, các kỹ thuật hậu xử lý được áp dụng bao gồm:
- GO hierarchy propagation (is_a, part_of),
- Lọc các nhãn gốc quá chung (GO roots),
- Kết hợp với phương pháp dựa trên homology (Diamond-KNN) để cải thiện recall.

Cách tiếp cận đa mô hình này cho phép tận dụng đồng thời:
- Độ chính xác cao của phương pháp homology
- Khả năng tổng quát hóa của mô hình học sâu

Kiến trúc này cho phép mô hình học được các mối quan hệ phi tuyến phức tạp giữa đặc trưng chuỗi protein và các nhãn chức năng GO.

3.3.2. Quy trình Huấn luyện Mô hình

Quá trình huấn luyện được thực hiện trên tập dữ liệu huấn luyện, trong đó mỗi mẫu bao gồm:

- Đầu vào: vector embedding của protein
- Nhãn mục tiêu: vector nhị phân đa chiều, biểu diễn sự có mặt hoặc vắng mặt của từng GO term.

Do đặc thù của bài toán đa nhãn, hàm mất mát Binary Cross-Entropy với Logits (BCEWithLogitsLoss) được sử dụng thay vì các hàm mất mát cho bài toán phân loại đơn nhãn. Hàm mất mát này cho phép mô hình học độc lập xác suất của từng GO term, không ép buộc các nhãn phải loại trừ lẫn nhau.

Quá trình tối ưu hóa được thực hiện bằng các thuật toán gradient-based như Adam optimizer, với các siêu tham số (learning rate, batch size, số epoch, hệ số regularization) được lựa chọn dựa trên thử nghiệm và tập validation. Trong quá trình huấn luyện, mô hình được đánh giá định kỳ trên tập validation để theo dõi hiệu suất và ngăn chặn hiện tượng overfitting.

3.3.3. Đầu ra của Quá trình Huấn luyện

Sau khi hoàn tất huấn luyện, mô hình có hiệu suất tốt nhất trên tập validation (theo các chỉ số đánh giá như F1-score hoặc proxy của Fmax) được lưu lại dưới dạng file trọng số: best_model.pth

File này chứa toàn bộ tham số đã học của mô hình dự đoán đa nhãn và được sử dụng ở giai đoạn suy luận để dự đoán xác suất GO terms cho các protein trong tập kiểm tra (test superset).

3.4. Suy luận (Inference)

Trong giai đoạn suy luận, embedding của tập kiểm tra được đưa qua mô hình đã huấn luyện. Mỗi protein sẽ được dự đoán một vector xác suất, trong đó mỗi giá trị biểu diễn độ tin cậy rằng protein đó có chức năng tương ứng.

Đầu ra được lưu thành: submission.tsv

với các cột: | EntryID | term | score |

Kết quả này sau đó được sử dụng để kết hợp (ensemble) với nhánh Homology nhằm tạo ra kết quả cuối cùng.

V. Tổng hợp và Hoàn thiện (Ensemble and Hierarchy)

Phần này mô tả giai đoạn cuối cùng của quy trình dự đoán chức năng protein trong cuộc thi CAFA 6. Sau khi hệ thống đã tạo ra hai nguồn dự đoán độc lập — gồm điểm số từ phương pháp tương đồng sinh học (Homology-Based Inference) và xác suất từ mô hình học sâu (Deep Learning Model) — giai đoạn tổng hợp (Ensemble) sẽ dung hòa hai nguồn thông tin nhằm tối ưu hóa độ chính xác. Sau đó, các điểm dự đoán được điều chỉnh theo cấu trúc phân cấp của Gene Ontology để đảm bảo tính hợp lệ về mặt sinh học.

1. Kết hợp Phương pháp (Ensemble)

Quá trình ensemble được xây dựng dựa trên nguyên tắc rằng mỗi phương pháp mang theo một loại "tri thức" khác nhau. Phương pháp tương đồng sinh học dựa trên mức độ giống nhau của chuỗi protein, thường cho kết quả chính xác cao đối với các protein có họ hàng gần hoặc đã được nghiên cứu kỹ. Ngược lại, mô hình học sâu có khả năng suy diễn tốt hơn đối với các protein mới hoặc có mức độ tương đồng thấp, nhờ khai thác các đặc trưng ngữ nghĩa ẩn trong không gian embedding.

1.1. Phương pháp Ensemble

Trong nghiên cứu này, hai nguồn điểm được kết hợp bằng chiến lược trung bình có trọng số (Weighted Average), trong đó tầm quan trọng của từng phương pháp được điều chỉnh dựa trên hiệu suất quan sát được trên tập validation. Điểm số cuối cùng cho từng GO term được tính bằng:

$$S_{\text{final}} = \alpha \cdot S_{\text{AI}} + (1 - \alpha) \cdot S_{\text{homology}}$$

trong đó:

- S_{AI} là xác suất dự đoán từ mô hình Deep Learning,
- S_{homology} là điểm số từ phương pháp tương đồng,
- α là hệ số trọng số, thường nằm trong khoảng $0.5 - 0.8$.

Hệ số này được chọn dựa trên quá trình thử nghiệm nhằm đạt giá trị F-max cao nhất.

1.2. Công thức Kết hợp

Công thức tổng quát đã nêu ở trên. Tùy theo loại GO term hoặc mức độ tin cậy của dữ liệu tương đồng, hệ thống có thể sử dụng các biến thể khác như Max-Pooling (giữ điểm cao nhất giữa hai nguồn) hoặc Normalized Weighted Fusion. Tuy nhiên, phương pháp trung bình có trọng số tỏ ra ổn định và dễ kiểm soát hơn trong bối cảnh của CAFA 6.

2. Áp dụng Cấu trúc Phân cấp (Hierarchy Constraints)

Gene Ontology có cấu trúc dạng đồ thị có hướng không chu trình (Directed Acyclic Graph – DAG). Điều này đồng nghĩa với việc nếu một protein được dự đoán có một chức năng con, thì nó phải được dự đoán có tất cả các chức năng cha tương ứng.

Do đó, sau khi điểm số ensemble được tạo ra, hệ thống cần thực hiện bước hiệu chỉnh phân cấp (hierarchical post-processing). Quy trình bao gồm:

1. Đọc cấu trúc GO từ file go-basic.obo.
2. Với mỗi GO term dự đoán, truy xuất tất cả các nút cha (ancestors).
3. Điều chỉnh các điểm số sao cho:

$$S(\text{parent}) \geq S(\text{child})$$

Bước này đảm bảo tính nhất quán sinh học và đồng thời giúp mô hình đạt điểm F-max cao hơn do giảm số lỗi phân cấp.

3. Kết quả Cuối cùng

Sau khi hoàn tất bước ensemble và bổ sung ràng buộc phân cấp, hệ thống tạo ra tập kết quả cuối cùng FINAL_SUBMISSION.tsv, bao gồm:

- Mã protein (EntryID)
- GO term dự đoán
- Điểm số cuối cùng (hierarchy-consistent score)

Tập tin này được định dạng theo đúng tiêu chuẩn của Kaggle và có thể nộp trực tiếp lên nền tảng cho vòng đánh giá của CAFA 6.

VI. Kết quả và Thảo luận (Results and Discussion)

Tên mô hình	Mô hình sử dụng	Mô tả	Percent	score
A	Diamond		100%	0.261
B	ResMLP		100%	0.164
C	Dae		100%	0.119
			B*60% + C*40%	0.135
			A*80% +B*15% + C*5%	0.27

1. Đánh giá Hiệu suất (Performance Evaluation)

Nhận xét và so sánh hai mô hình baseline

Kết quả thực nghiệm cho thấy cả Logistic Regression và XGBoost đều đạt hiệu năng tốt khi kết hợp với embedding ESM2 và GO hierarchy propagation. Tuy nhiên, Logistic Regression cho kết quả nhỉnh hơn XGBoost trên các chỉ số quan trọng như Micro-F1 và Micro-Recall, cho thấy các embedding ESM2 đã mã hóa sẵn nhiều quan hệ phi tuyến và phù hợp với các mô hình tuyến tính. Logistic Regression cũng thể hiện khả năng tổng quát hóa ổn định hơn trong bối cảnh dữ liệu Gene Ontology mất cân bằng. Trên cơ sở này, Logistic Regression được lựa chọn làm baseline chính, đồng thời mở ra hướng nâng cấp bằng các mô hình học sâu như MLP, cũng như các phương pháp khai thác tương đồng chuỗi (DIAMOND database) và chiến lược ensemble, sẽ được trình bày trong các phần tiếp theo.

2. Phân tích Các Kết quả Quan trọng

Kết quả cho thấy sự khác biệt rõ rệt giữa các hướng tiếp cận dựa trên tương đồng trình tự (DIAMOND) và mô hình học sâu trên embedding (ResMLP/DAE), cũng như hiệu quả khi kết hợp ensemble.

2.1. Kết quả theo từng mô hình đơn lẻ

Mô hình A – DIAMOND đạt 0.261 (100%)

DIAMOND cho điểm cao nhất trong các mô hình đơn lẻ. Điều này phù hợp với bản chất bài toán CAFA: các protein có thể được dự đoán chức năng tốt nếu tìm được protein tương đồng có annotation trong tập train. DIAMOND hoạt động như một "nearest-neighbor trên trình tự", thường cho dự đoán có độ tin cậy cao và ổn định. Vì vậy, DIAMOND đóng vai trò backbone (nền) cho hệ thống.

Mô hình B – ResMLP đạt 0.164 (100%)

ResMLP dựa trên embedding (ví dụ ESM) học mapping từ vector biểu diễn sang GO terms. Điểm số thấp hơn DIAMOND cho thấy mô hình học sâu vẫn gặp thách thức lớn ở bài toán đa nhãn cực rộng (nhiều GO terms, phân bố lệch, nhãn hiếm). ResMLP thường có xu hướng dự đoán "mềm" hơn (score phân tán), nên nếu ngưỡng export không phù hợp sẽ làm giảm recall hoặc tăng nhiễu.

Mô hình C – DAE đạt 0.119 (100%)

DAE cho điểm thấp nhất trong ba mô hình đơn lẻ. Điều này thường xảy ra khi mô hình bị "underfit/over-regularize" hoặc không đủ năng lực để học ranh giới tốt trong không gian đa nhãn. Ngoài ra DAE đôi khi phù hợp hơn với vai trò "bổ trợ" (regularization/denoise) hơn là mô hình chính để dự đoán trực tiếp.

2.2. Kết quả khi ensemble giữa các mô hình học sâu

Ensemble B 60% + C 40% cho score 0.135

Kết quả này thấp hơn cả ResMLP đơn (0.164). Điều này cho thấy DAE không bổ sung được tín hiệu hữu ích cho ResMLP ở thiết lập hiện tại, thậm chí còn kéo xuống do:

- Hai mô hình có thể dự đoán "khác hướng" nhưng phần chồng lắp không giúp tăng Fmax.
- DAE có thể thêm nhiều dự đoán nhiễu (false positives), làm giảm precision.
- Nếu export dựa trên threshold/floor, việc "pha" thêm DAE làm thay đổi phân bố score theo hướng bất lợi.

=> Từ kết quả này, có thể kết luận: DAE chưa phù hợp để blend trực tiếp với ResMLP theo tỉ lệ lớn. Nếu vẫn dùng, DAE nên có trọng số nhỏ hoặc chỉ dùng ở vai trò phụ (ví dụ regularization trong training, hoặc chỉ add top-term rất chắc).

2.3. Kết quả khi kết hợp DIAMOND + mô hình học sâu

Ensemble A 80% + B 15% + C 5% đạt 0.27 (cao nhất)
Đây là kết quả quan trọng nhất: khi dùng DIAMOND làm nền (80%) và chỉ thêm một phần nhỏ tín hiệu từ ML (ResMLP/DAE), điểm số tăng từ 0.261 lên 0.27.

Kết quả này cho thấy:
- DIAMOND cung cấp độ chính xác nền nhờ similarity theo trình tự.
- ResMLP/DAE có thể bổ sung recall ở những trường hợp DIAMOND không tìm được hit tốt, hoặc chức năng không được truyền tốt qua hàng xóm gần.
- Tuy nhiên, ML chỉ nên đóng góp nhỏ (15% và 5%) vì nếu đưa tỷ trọng lớn sẽ làm nhiễu và giảm precision.

Nói cách khác, ML đang đóng vai trò "bổ sung có kiểm soát" thay vì thay thế DIAMOND.

2.4. Kết luận rút ra từ các kết quả

- DIAMOND là backbone mạnh nhất trong hệ thống hiện tại (đạt 0.261).
- ResMLP có tín hiệu hữu ích nhưng chưa đủ mạnh khi đứng một mình (0.164).
- DAE đơn lẻ yếu (0.119) và blend lớn với ResMLP gây giảm hiệu quả (0.135).
- Blend tốt nhất là giữ DIAMOND trọng số cao, chỉ thêm ML ở tỷ lệ nhỏ → đạt 0.27

3. Bài học Rút ra

Trong quá trình xây dựng và thử nghiệm pipeline dự đoán chức năng protein cho bài toán CAFA6, một số bài học quan trọng đã được rút ra liên quan đến cả khía cạnh kỹ thuật lẫn chiến lược tối ưu điểm số.

Trước hết, các bước tiền xử lý dữ liệu đóng vai trò then chốt và có ảnh hưởng lớn đến kết quả cuối cùng. Những sai sót nhỏ như xử lý không đúng định dạng protein ID (ví dụ: sp|P12345|…, isoform P12345-2, hoặc version P12345.1) hay thứ tự lọc sai trong bước KNN (lọc topK trước khi loại bỏ các hit không có annotation) có thể làm giảm đáng kể recall, dẫn đến điểm số bị "kẹt" dù các bước sau được tinh chỉnh kỹ lưỡng. Khi các lỗi này được khắc phục, chất lượng dự đoán được cải thiện rõ rệt mà không cần thay đổi mô hình học sâu.

Thứ hai, DIAMOND/KNN tỏ ra là backbone rất mạnh cho bài toán CAFA, đặc biệt trong việc khai thác thông tin tương đồng trình tự. Tuy nhiên, việc tối ưu quá thiên về precision (ví dụ tăng SCORE_FLOOR cao, giảm số term xuất ra mỗi protein) thường làm giảm recall, khiến Fmax trên leaderboard giảm. Điều này cho thấy, trong CAFA, điểm số tối ưu không đạt được ở cấu hình "sạch nhất" mà ở cấu hình cân bằng giữa precision và recall.

Thứ ba, việc sử dụng threshold tối ưu trên validation (micro-F1) để export submission không phù hợp trực tiếp với cơ chế chấm điểm Fmax của CAFA. Các mô hình ResMLP/DAE thường cho threshold tối ưu quanh 0.10–0.12 trên validation, nhưng nếu áp dụng trực tiếp threshold này khi export submission sẽ khiến số lượng dự đoán bị cắt mạnh, làm recall giảm nghiêm trọng. Thực tế cho thấy, chiến lược export dựa trên topK ứng viên kết hợp với score floor nhỏ và giới hạn số term/protein mang lại kết quả tốt hơn.

Thứ ba, cải tiến chiến lược IA reweighting bằng các hàm mềm hơn và tránh khuếch đại quá mức các term hiếm. Việc sử dụng hàm log hoặc chuẩn hóa IA theo ontology có thể giúp giảm false positives mà vẫn giữ được lợi thế của các term mang tính đặc trưng cao.

Thứ tư, negative filtering dựa trên GOA (NOT qualifiers) là một hướng hợp lý để giảm false positives, tuy nhiên chỉ nên sử dụng như một bộ lọc. Việc thêm trực tiếp các GOA positive với score = 1.0 tiềm ẩn rủi ro spam và overfitting. Một hướng an toàn hơn là chỉ loại bỏ các cặp protein – GO bị đánh dấu NOT, hoặc nếu cần thiết thì chỉ boost nhẹ score của các GOA positives.

Thứ năm, tối ưu hiệu năng tính toán là yếu tố quan trọng khi xử lý dữ liệu lớn. Việc thay thế pandas groupby bằng cách quét file theo dòng (line-by-line streaming) hoặc sử dụng các thư viện tối ưu hơn như polars có thể giúp giảm đáng kể thời gian chạy mà không ảnh hưởng đến kết quả.

Cuối cùng, thay thế hoặc bổ sung DIAMOND bằng KNN trên embedding (FAISS) là một hướng nghiên cứu tiềm năng. Với các embedding chất lượng cao như ESM hoặc ProtT5, FAISS có thể khai thác tương đồng ngữ nghĩa giữa protein, bổ sung thông tin mà alignment truyền thống khó nắm bắt.