

VIỆN NGHIÊN CỨU VÀ ĐÀO TẠO VIỆT - ANH, ĐẠI HỌC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO CUỐI KỲ
DATA ANALYTICS FOR LIFE SCIENCE

ĐỀ TÀI: Ứng dụng deep learning (CNN-LSTM) và explainable ai (XAI) trong dự đoán hiệu quả cắt của hệ thống crispr-cas9

Giáo viên hướng dẫn: ThS. Nguyễn Chí Thiện

TS. Trần Thanh Hòa

Họ và tên sinh viên:

- 1. Trần Xuân Cường.....22040004**
- 2. Nguyễn Văn Nhi.....22040006**

MỤC LỤC

1. PROJECT OVERVIEW	3
1.1. Objectives	3
1.2. Technology used	3
1.3. Project Structure	4
2. ANTSBAMBOOSYSTEM (HỆ THỐNG KIẾN TRÚC)	4
2.1. Overall Architecture	4
2.2. Models Used	5
3. MODEL TRAINING PROCESS	6
3.1. Data Preparation	6
3.2. Data Splitting	6
3.3. Training Setup:	6
3.4. Callbacks Strategy:	7
3.5. Result:	7
4. BUILDING THE STREAMLIT APP	7
4.1. Application Architecture	7
4.2. Main Functions	8
4.2.1. Batch CSV Upload (Chức năng nhập liệu hàng loạt)	8
4.2.2. Single SMILES (Chức năng nhập liệu đơn lẻ - DNA Sequence)	8
4.3. XAI Integration	8
5. PIPELINE VÀ EXPLAINABLE AI	9
5.1. End-to-End Screening Pipeline	9
5.2. Explainable AI (XAI)	9
6. RESULTS AND EVALUATION	10
7. CONCLUDE	10

1. PROJECT OVERVIEW

1.1. Objectives

Dự án này được phát triển với mục tiêu giải quyết bài toán cốt lõi trong công nghệ chỉnh sửa gen CRISPR-Cas9: dự đoán chính xác hiệu suất cắt (cleavage efficiency) của các single-guide RNA (sgRNA). Việc lựa chọn sgRNA tối ưu là yếu tố tiên quyết để đảm bảo thành công của thí nghiệm và hạn chế các tác động ngoài mục tiêu (off-target effects).

Các mục tiêu cụ thể bao gồm:

1. **Xây dựng mô hình Deep Learning (Học sâu):** Phát triển một kiến trúc mạng nơ-ron lai ghép (Hybrid Neural Network) có khả năng học các đặc trưng phức tạp từ trình tự DNA để dự đoán điểm hiệu quả (efficiency score).
2. **Tích hợp Khả giải thích (Explainable AI - XAI):** Không chỉ đưa ra dự đoán, hệ thống phải cung cấp cơ chế giải thích lý do tại sao một trình tự lại có hiệu suất cao hoặc thấp thông qua bản đồ quan trọng (Saliency Maps).
3. **Triển khai Ứng dụng thực tế:** Xây dựng giao diện tương tác người dùng (Web UI) hiện đại, cho phép các nhà nghiên cứu sinh học dễ dàng nhập liệu và nhận kết quả phân tích trực quan mà không cần kiến thức lập trình.

1.2. Technology used

Dự án sử dụng hệ sinh thái Python với các thư viện tiên tiến nhất trong lĩnh vực Khoa học dữ liệu và Trí tuệ nhân tạo:

- **Ngôn ngữ lập trình:** Python 3.x.
- **Deep Learning Framework:** TensorFlow/Keras (xây dựng, huấn luyện và lưu trữ mô hình).
- **Xử lý & Phân tích dữ liệu:**
 - *Pandas & NumPy:* Thao tác trên khung dữ liệu bảng và tính toán ma trận số học.
 - *Scikit-learn:* Chia tập dữ liệu, chuẩn hóa và đánh giá các chỉ số thống kê (Spearman correlation).
 - *SciPy:* Hỗ trợ các tính toán khoa học chuyên sâu.

- **Web Framework:** Streamlit (Xây dựng giao diện web tương tác, Dashboard).
- **Visualization (Trực quan hóa):**
 - *Matplotlib & Seaborn:* Vẽ biểu đồ trong quá trình huấn luyện và đánh giá model.
 - *Altair:* Vẽ biểu đồ tương tác trên ứng dụng web.
- **Explainable AI:** GradientTape (tính toán đạo hàm để xây dựng Saliency Maps).

1.3. Project Structure

Cấu trúc dự án được tổ chức khoa học thành các module riêng biệt để đảm bảo tính duy trì và mở rộng:

1. Data Processing & Training Module (CRISPR-Cas9.ipynb):

- Chịu trách nhiệm tải dữ liệu thô.
- Tiền xử lý: Cắt chuỗi DNA (23bp), One-Hot Encoding.
- Định nghĩa kiến trúc Model.
- Huấn luyện, tinh chỉnh (Fine-tuning) và đánh giá mô hình.

2. Application Logic & Interface (app.py):

- Tải mô hình đã huấn luyện (best_model.keras).
- Xử lý logic quét chuỗi DNA (DNA Scanning Algorithm).
- Tích hợp module XAI để giải thích dự đoán.
- Xây dựng giao diện người dùng (UI/UX) theo phong cách "Dark Biotech".

2. ANTSBAMBOOSYSTEM (HỆ THỐNG KIẾN TRÚC)

2.1. Overall Architecture

Hệ thống được thiết kế theo luồng xử lý tuần tự khép kín (End-to-End Pipeline), đảm bảo dữ liệu đi từ dạng thô đến kết quả phân tích cuối cùng một cách mượt mà:

- **Bước 1 - Input Interface:** Người dùng cung cấp chuỗi gen đích (Target DNA Sequence).

- **Bước 2 - Pattern Matching Engine:** Hệ thống sử dụng thuật toán cửa sổ trượt (sliding window) để tìm kiếm motif PAM ("GG"). Tại mỗi vị trí tìm thấy, hệ thống trích xuất 23 nucleotide liền trước để tạo thành ứng viên sgRNA.
- **Bước 3 - Vectorization:** Các chuỗi ứng viên (A, T, G, C) được chuyển đổi sang dạng ma trận số học thông qua One-Hot Encoding.
- **Bước 4 - Inference Engine:** Ma trận đầu vào được đưa qua mô hình Deep Learning đã huấn luyện để tính toán xác suất hiệu quả (Efficiency Score).
- **Bước 5 - Interpretation Layer (XAI):** Đối với các kết quả được chọn, hệ thống kích hoạt lớp giải thích để tính toán trọng số ảnh hưởng của từng vị trí nucleotide.
- **Bước 6 - Visualization & Reporting:** Kết quả được tổng hợp thành bảng xếp hạng, biểu đồ nhiệt và báo cáo văn bản hiển thị trên Streamlit.

2.2. Models Used

Trái tim của hệ thống là một mô hình **Hybrid CNN-BiLSTM-Attention**, được thiết kế chuyên biệt để xử lý dữ liệu chuỗi sinh học.

- **Input Layer:** Chấp nhận đầu vào có kích thước (23, 4), tương ứng với chiều dài chuỗi 23bp (20bp spacer + 3bp PAM) và 4 kênh one-hot.
- **Convolutional Block (Multi-scale CNN):**
 - Sử dụng 3 nhánh Convolution 1D song song với các kích thước hạt nhân (kernel size) lần lượt là 3, 5, và 7.
 - *Mục đích:* Nhánh kernel nhỏ (3) học các motif cục bộ, trong khi nhánh kernel lớn (7) học các mẫu ngữ cảnh rộng hơn.
 - Mỗi nhánh đi kèm lớp BatchNormalization để ổn định quá trình học và hàm kích hoạt ReLU để tăng tính phi tuyến.
 - Kết quả từ 3 nhánh được gộp lại bằng lớp Concatenate.
- **Sequence Modeling Block (Bi-LSTM):**
 - Sử dụng lớp Bidirectional LSTM với 128 đơn vị ẩn (units).
 - *Mục đích:* Học sự phụ thuộc xa và ngữ cảnh hai chiều của chuỗi DNA (tương tác giữa các nucleotide ở đầu và cuối chuỗi).

- **Attention Mechanism (Simplified):**
 - Sử dụng lớp GlobalAveragePooling1D để nén đặc trưng, tập trung vào các thông tin quan trọng nhất trên toàn bộ chuỗi thời gian.
- **Prediction Head (Fully Connected):**
 - Các lớp Dense (128 units, 64 units) kết hợp với Dropout (0.3) để giảm thiểu hiện tượng Overfitting.
 - Lớp Output sử dụng hàm kích hoạt Sigmoid (hoặc Linear tùy chỉnh trong file huấn luyện) để trả về một giá trị thực (Regression) đại diện cho điểm hiệu quả.

3. MODEL TRAINING PROCESS

3.1. Data Preparation

Quy trình chuẩn bị dữ liệu trong CRISPR-Cas9.ipynb bao gồm các bước nghiêm ngặt:

- **Data Ingestion:** Dữ liệu được đọc từ tệp CSV (FC_plus_RES_withPredictions.csv), chứa các cặp trình tự sgRNA và điểm hiệu quả thực nghiệm.
- **Sequence Processing:** Dữ liệu thô (thường là 30mer) được cắt (slice) để lấy vùng quan trọng nhất (23bp), bao gồm 20bp protospacer và 3bp PAM.
- **One-Hot Encoding:**
 - Chuyển đổi mỗi nucleotide thành vector 4 chiều: A=[1,0,0,0], C=[0,1,0,0], G=[0,0,1,0], T=[0,0,0,1].
 - Kích thước đầu vào của mỗi mẫu dữ liệu là (23, 4).

3.2. Data Splitting

Để đảm bảo tính khách quan trong đánh giá, dữ liệu được chia theo tỷ lệ chuẩn:

- **Training Set (80%):** Dùng để cập nhật trọng số mạng.
- **Validation Set (10%):** Dùng để kiểm tra chéo trong quá trình huấn luyện và điều chỉnh siêu tham số (Hyperparameter Tuning).
- **Test Set (10%):** Dữ liệu hoàn toàn mới, chỉ dùng để đánh giá độ chính xác cuối cùng của mô hình.

3.3. Training Setup:

- **Optimizer:** Sử dụng thuật toán Adam với tốc độ học (learning rate) khởi tạo là 0.0003 - mức tối ưu để mô hình hội tụ ổn định mà không bị dao động quá mạnh.
- **Loss Function:** Sử dụng Mean Squared Error (MSE) vì đây là bài toán hồi quy (dự đoán giá trị liên tục).
- **Evaluation Metric:** Sử dụng Mean Absolute Error (MAE) để theo dõi sai số trung bình thực tế.

3.4. Callbacks Strategy:

Chiến lược Callbacks là điểm sáng giúp mô hình đạt hiệu suất cao:

- **ModelCheckpoint:** Chỉ lưu lại mô hình khi val_loss đạt giá trị thấp nhất. Điều này đảm bảo mô hình cuối cùng (best_model.keras) là phiên bản tốt nhất, không phải phiên bản ở epoch cuối cùng (có thể đã bị overfitting).
- **EarlyStopping:** Tự động dừng huấn luyện nếu val_loss không cải thiện sau một số epoch (patience) nhất định, giúp tiết kiệm tài nguyên tính toán.
- **ReduceLROnPlateau:** Giảm tốc độ học khi loss đi ngang, giúp mô hình hội tụ sâu hơn vào điểm cực tiểu toàn cục.

3.5. Result:

- Mô hình được huấn luyện qua 100 epochs (thực tế dừng sớm hơn nhờ EarlyStopping).
- Biểu đồ Loss cho thấy sự hội tụ tốt giữa tập Train và Validation, không có dấu hiệu phân kỳ (Overfitting).

4. BUILDING THE STREAMLIT APP

4.1. Application Architecture

Ứng dụng được xây dựng trên nền tảng **Streamlit**, đóng vai trò là cầu nối giữa mô hình AI phức tạp và người dùng cuối.

- **Frontend Layer:** Sử dụng Custom CSS để tạo giao diện "Dark Mode" chuyên nghiệp. Các thành phần như nút bấm, bảng hiển thị được thiết kế với hiệu ứng Neon, tạo cảm giác công nghệ cao (Biotech/Cyberpunk style).

- **Backend Layer:** Chạy trên môi trường Python, quản lý việc tải TensorFlow model vào bộ nhớ (Caching) để tăng tốc độ phản hồi cho các lần dự đoán sau.

4.2. Main Functions

4.2.1. Batch CSV Upload (Chức năng nhập liệu hàng loạt)

Trong phiên bản hiện tại, chức năng này được tích hợp thông qua việc quét chuỗi Gen dài (DNA Scanning).

- Hệ thống cho phép người dùng nhập một đoạn DNA dài (Gene Sequence) thay vì từng chuỗi ngắn lẻ tẻ.
- Thuật toán bên dưới sẽ hoạt động như một trình xử lý hàng loạt (Batch Processor): Tự động cắt chuỗi dài thành hàng trăm đoạn con (sgRNA candidates) dựa trên vị trí PAM.
- Toàn bộ danh sách này được đưa vào mô hình xử lý song song, trả về kết quả dưới dạng bảng dữ liệu có thể sắp xếp và lọc.

4.2.2. Single SMILES (Chức năng nhập liệu đơn lẻ - DNA Sequence)

Dựa trên bản chất sinh học của dự án CRISPR, mục này đề cập đến việc xử lý "Single DNA Sequence".

- Người dùng nhập trực tiếp một chuỗi DNA 23bp cụ thể.
- Hệ thống kiểm tra tính hợp lệ của chuỗi (độ dài, ký tự cho phép A/T/G/C).
- Mô hình thực hiện dự đoán tức thì và trả về điểm số cụ thể kèm theo phân loại (High/Medium/Low).
- Tại đây, chức năng XAI được kích hoạt ngay lập tức để giải thích cho kết quả đơn lẻ này.

4.3. XAI Integration

- Ứng dụng tích hợp một module trực quan hóa riêng biệt cho XAI.
- Khi người dùng chọn một kết quả dự đoán, ứng dụng sẽ gọi hàm `get_saliency_map`.

- Kết quả trả về là một mảng giá trị trọng số, được tô màu trực tiếp lên chuỗi DNA hiển thị trên màn hình (HTML/CSS injection).
- Màu càng đậm (đỏ/cam) thể hiện nucleotide đó càng quan trọng đối với quyết định của mô hình.

5. PIPELINE VÀ EXPLAINABLE AI

5.1. End-to-End Screening Pipeline

Quy trình sàng lọc (Screening Pipeline) trong hệ thống được tự động hóa hoàn toàn:

1. **Raw Gene Input:** Nhận chuỗi gen thô (ví dụ: trình tự gen TP53 hoặc MYC).
2. **Preprocessing & Validity Check:** Loại bỏ ký tự xuống dòng, khoảng trắng, chuyển về in hoa. Kiểm tra tính hợp lệ sinh học.
3. **Candidate Discovery:** Quét tìm motif 'GG' (PAM). Lấy lùi về trước 21 ký tự để tạo chuỗi 23bp.
4. **Model Inference:** Chạy mô hình Hybrid CNN-LSTM.
5. **Post-processing:**
 - Chuyển đổi giá trị dự đoán (0-1) thành thang điểm phần trăm.
 - Xếp hạng (Ranking) dựa trên điểm số.
 - Gán nhãn chất lượng (Excellent, Good, Average, Poor).

5.2. Explainable AI (XAI)

Đây là tính năng đột phá giúp hệ thống trở thành một "Oracle" (Nhà tiên tri) thực thụ thay vì chỉ là một cỗ máy tính toán.

- **Cơ chế Gradient-based Saliency:**
 - Hệ thống sử dụng tf.GradientTape để theo dõi quá trình lan truyền ngược (Backpropagation) từ đầu ra (Score) về đầu vào (Input Embedding).
 - Công thức: $S = |\frac{\partial y}{\partial x}|$. Trong đó y là điểm hiệu quả, x là vector one-hot đầu vào.
 - Giá trị tuyệt đối của đạo hàm cho biết sự thay đổi nhỏ tại nucleotide đó sẽ làm thay đổi kết quả dự đoán nhiều như thế nào.
- **Trực quan hóa:**

- Kết quả Saliency được chuẩn hóa về đoạn $[0, 1]$ và vẽ dưới dạng biểu đồ cột màu (Heatmap Bar Chart).
- Các vị trí quan trọng (thường là vùng Seed gần PAM) sẽ có cột cao và màu sắc nổi bật (đỏ/cam), giúp nhà nghiên cứu nhận diện ngay lập tức vùng chức năng của sgRNA.
- **Quy tắc sinh học bổ sung:**
 - Bên cạnh AI, hệ thống tích hợp các quy tắc sinh học cố định (Rule-based):
 - **GC Content:** Tính tỷ lệ G và C, cảnh báo nếu nằm ngoài khoảng tối ưu (40-70%).
 - **Motif Warning:** Tự động quét và cảnh báo các chuỗi nguy hiểm như Poly-T (TTTT) - gây kết thúc sớm quá trình phiên mã, hoặc Poly-G/Poly-C.

6. RESULTS AND EVALUATION

Kết quả đánh giá trên tập dữ liệu kiểm thử (Test Set) cho thấy hiệu năng ấn tượng của mô hình:

- **Hiệu năng mô hình:** Mô hình Hybrid CNN-LSTM đạt độ chính xác cao trong việc dự đoán hiệu quả cắt, vượt trội hơn so với các phương pháp Machine Learning truyền thống nhờ khả năng học đặc trưng tuần tự.
- **Tính minh bạch:** Việc tích hợp XAI giúp xác thực độ tin cậy của mô hình. Các vùng quan trọng mà AI chỉ ra (thông qua Saliency Map) khớp với các kiến thức sinh học thực nghiệm (ví dụ: tầm quan trọng của vùng Seed 10-12bp gần PAM).
- **Trải nghiệm người dùng:** Ứng dụng Streamlit hoạt động mượt mà, giao diện trực quan, cung cấp thông tin toàn diện (điểm số + giải thích + cảnh báo an toàn), hỗ trợ đắc lực cho quy trình thiết kế thí nghiệm.

7. CONCLUDE

Tổng kết các đóng góp chính:

1. Đề xuất thành công kiến trúc mạng **Hybrid CNN-BiLSTM**, chứng minh khả năng học đặc trưng chuỗi gen vượt trội so với các phương pháp truyền thống.
2. Giải quyết vấn đề minh bạch hóa AI thông qua **XAI (Saliency Maps)**, giúp xây dựng niềm tin cho người dùng là các nhà sinh học/người nghiên cứu viên.
3. Ứng dụng **Streamlit** hoạt động ổn định, giao diện thân thiện, tích hợp đầy đủ quy trình từ nhập liệu đến báo cáo, sẵn sàng cho việc triển khai thực tế.

Hướng phát triển trong tương lai:

- Mở rộng tập dữ liệu huấn luyện để bao gồm nhiều loại tế bào và điều kiện thí nghiệm khác nhau nhằm nâng cao độ chính xác tổng quát.
- Tích hợp thêm module dự đoán **Off-target** (cắt nhầm) để cung cấp cái nhìn toàn diện hơn về độ an toàn của sgRNA.
- Phát triển phiên bản API để tích hợp vào các pipeline tin sinh học lớn hơn.